Automated Registering System through Facial Recognition

David Han1\*, Krish Gupta1\*\*, Jack Xiang1\*\*\*

*1King George V School, Hong Kong*

*\*Contact: dhan4805@gmail.com*

*\*\*Contact: krishgupta2701@gmail.com*

*\*\*\*Contact: jackylx112358@gmail.com*

***Abstract* — In this project, we created, trained, and tested facial recognition software that automatically records attendance for students as they enter a classroom. We were driven to execute this project since we have firsthand experience with the inefficiencies of standard attendance taking methods. This learning summary documents our learning journey in this course and findings from our project.**

1. Introduction

Attendance taking is an essential process in almost any class no matter the education level. Recording attendance is important in order to maintain a full class, maintain students' accountability and to ensure student safety in general. However, it is extremely time consuming. Assuming that manually registering one student takes 3-4 seconds, completing the register for a single class can take 1-2 minutes. This means that around ~3% of class time is wasted on doing the register in classes. Teachers may also occasionally make mistakes by forgetting to call a student’s name or recording attendance incorrectly. Recognising this issue, we decided to create a solution which would streamline the attendance-taking process.

1. Methodology

This project can be broken down into 3 main stages:

1. *Background Research*

The first stage involved investigating the market for existing solutions and determining whether or not there was a need to solve the problem we discovered. The first mentorship session enabled us to communicate our ideas with various industry leaders, principals and professors. By discussing the issue we received assurance that the issue we defined was a real-world problem that needed to be solved. This can save precious class time.

After looking through several academic papers, we identified one potential solution to this challenge: a South African study that attempted to utilise radio frequency signals transmitted by the students' phones. It worked successfully, but required access to the students’ phones. Therefore, this solution would not be suitable for younger students, which is a large portion of our target user base.

1. *Devising a solution*

The next stage was brainstorming other possible solutions and then selecting the optimal idea.

One initial idea we had was to track the classroom’s audio and record attendance by determining which students are speaking. However, there seemed to be too many uncertainties with this solution. For example, it may be difficult to track multiple students speaking at once and some students might not speak at all. Another idea was to use thumb prints since it is a common biometric authentication method. However, this type of equipment may not be feasible in a school setting and would likely take just as long as traditional attendance taking methods.

We finally decided on using facial recognition by placing a camera at the door of every classroom to automatically recognise and register students. This solves our initial problem by reducing attendance taking time. Furthermore, the technology requirement is only a camera and we make good use of the AI knowledge we learnt in this course.

1. *Developing the program*

Once we decided on the optimal solution, we moved onto the third and final stage of our project: developing the program. The steps below describe how our program works. When the program is run, the camera is activated and shows a live feed of what is in front of it. It stores an image of each frame of video captured. Using the face-recognition library in Python, the program first searches for any faces on the video capture, by searching for changes in brightness of the image. Then, it encodes the details of 68 landmarks on each face detected using a pre-existing model on the internet to generate 128 numerical features.

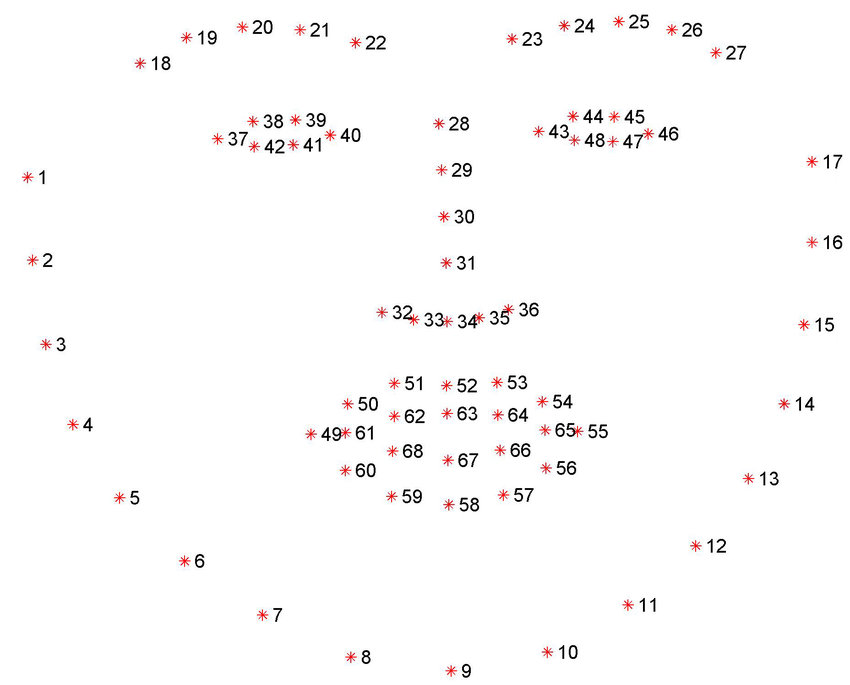


Fig. 1 A model of the 68 landmarks identified by the facial recognition software [1]

The encodings are then compared to the encodings generated from an existing database of faces. If the encodings are found to be sufficiently similar, a match is discovered, and the student's face is recognised. This can then be stored in the register list.

On the display, a rectangle is drawn around the face(s) detected, and the names which are matched to the face(s) are displayed below. As the student rotates and shifts the position of their face, there may be a few frames where the face is recognised as many different people; thus, each face must be present in the video frame for a certain number of frames before they are added to the register.

Once class has started, and the camera recording is finished, a list of students present is outputted.

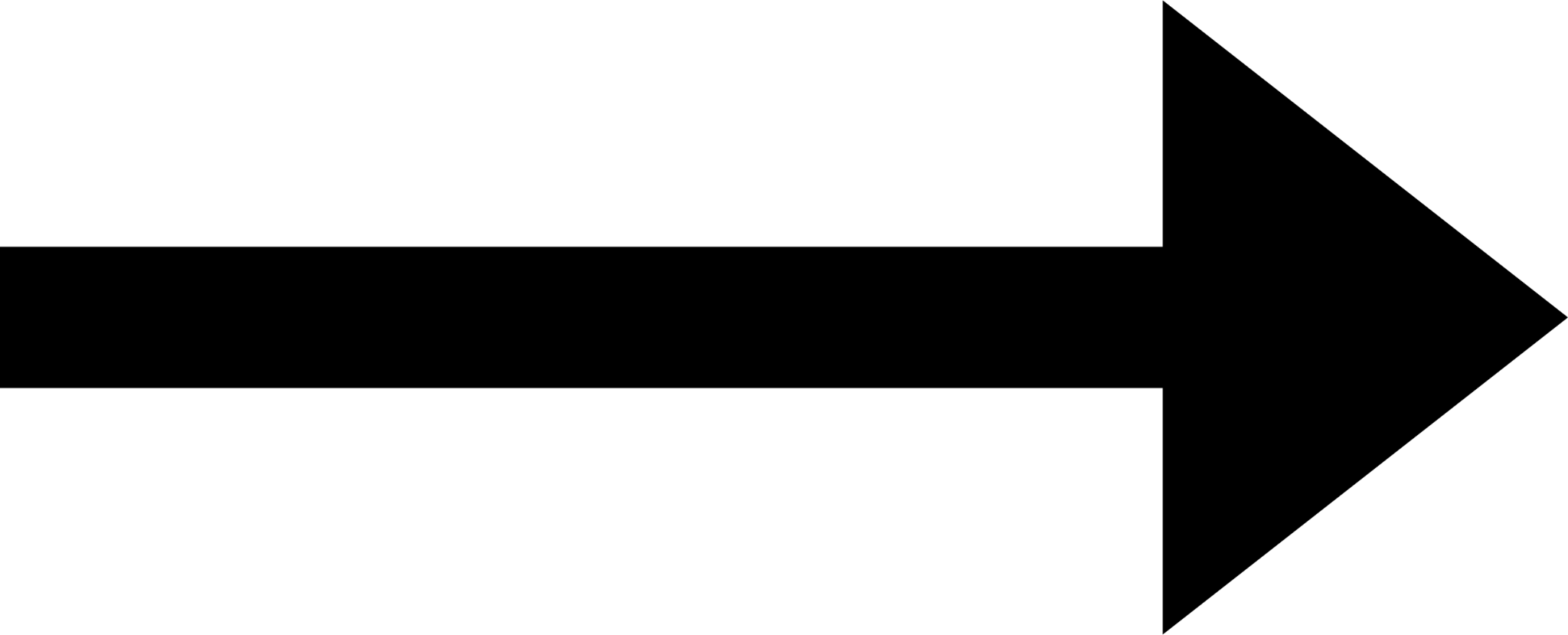
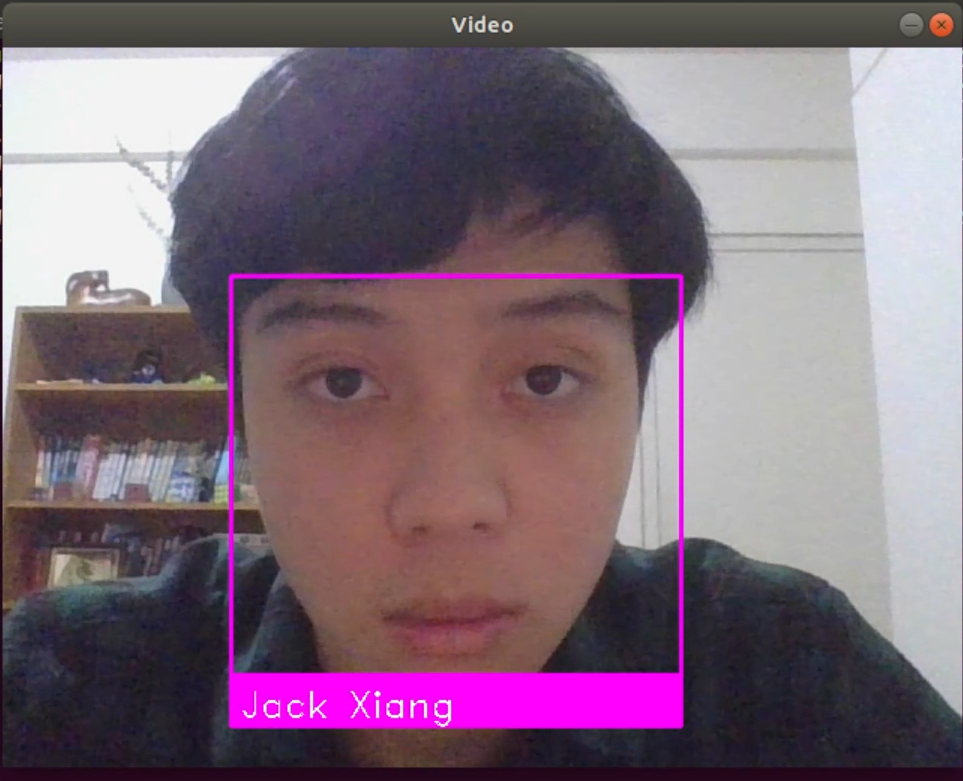


Fig. 2 A demonstration of the software recognising a student and marking them as present on the register system

1. Results / Application

We decided to test the application in our computer science class after successfully trialing it in our group. This would not only allow us to put the product to the test in the real world, but also provide us with valuable feedback. After obtaining their permission to test our program, we collected one image from each of them. We then labeled and uploaded them into our database. During this beta test we ran the program using the laptop camera during this test because it was more convenient.



Fig. 3 The results of our test in a real classroom.

The classroom test scores were positive. Our algorithm correctly identified 13/14 of the students evaluated, with a 92.9% accuracy. This is especially impressive given that each student's photograph was only found in the database once. Furthermore, we were able to get direct input from our target market (students and teachers). The general consensus was that this program might significantly enhance the school’s attendance taking ability.

1. Discussion / Limitation

Despite having a relatively high accuracy of 92.9% and recognizing students entering the classroom with relative ease, our algorithm has a few limitations. The issues and possible solutions we have identified are listed below.

* 1. *Privacy*

The biggest concern raised by our solution is privacy. For our system to work, we would need to store the faces of all students attending a class. This requires their consent (or their parents if they are aged under 18). As we discovered from our testing, many students in our class refused to share multiple photos of themselves for the training data; this may impair the accuracy of our program. Furthermore, unlike many other forms of data, faces cannot be encrypted [2]. To ensure students' pictures are kept secure at all times we may store their faces on a local server and ensure the servers are protected.

* 1. *Effectiveness*

Although the program was relatively effective, there are certain factors that may impede the program's accuracy.

First is the student's position relative to the camera. When students stand far away from the camera, they are less likely to be detected or detected less accurately. Moreover, if they aren’t directly facing the camera they may be harder to recognise. To solve this issue, we could potentially use a high resolution camera. Additionally, in our training data we only had one picture of each student. By increasing this number and including student’s faces with different angles, the issue regarding the student's position affecting the program's accuracy can be resolved.

Our software also had minor difficulties in recognising multiple people at once. We can solve this by using semantic segmentation, which groups pixels of the same object by analyzing colour gradients. However, this is likely not a big issue since students walk through the door one by one.

* 1. *Face Spoofing*

Another possible issue arises when students attempt to deceive the system by presenting an image of another person’s face to the camera. This is called face spoofing; the act of using a photo or video of someone else’s face in order to pass as a different identity. This is a significant concern as students could bring photos of absent classmates in order to take attendance for them. A solution for this is to use cameras to also detect blinking. By implementing an algorithm to detect eye blinks as a pattern of EAR values in a short temporal window and comparing it to the Modified EAR threshold value, we could detect whether a person is actually present [3].

1. Conclusions

We learned a lot about machine learning, programming, debugging, and so much more through this project! Overall, we had a terrific time working together to solve real world issues since it was both challenging yet rewarding at the same time. Our efforts were rewarded when the model gave accurate findings! Looking forward, we hope that schools will start adapting facial recognition to identify students. This process won’t just make classes efficient, but also remove the possibility of human errors.

Acknowledgment

We would like to thank Dr. Ray Cheung, John Wong and all the TAs for their constant support and advice. We also thank the Hong Kong Education Bureau’s Gifted Education Fund and the City University of Hong Kong for providing us with this unique opportunity.

References

1. Rosebrock, A. (2017, April 3). Facial landmarks with dlib, OpenCV, and Python. PyImageSearch. Retrieved February 15, 2023, from https://pyimagesearch.com/2017/04/03/facial-landmarks-dlib-opencv-python/
2. Lively, T. K. (2021, December 1). Facial Recognition in the US: Privacy Concerns and Legal Developments. ASIS International. Retrieved February 15, 2023, from https://www.asisonline.org/security-management-magazine/monthly-issues/security-technology/archive/2021/december/facial-recognition-in-the-us-privacy-concerns-and-legal-developments/
3. Dewi, C. (2022, April 18). Adjusting eye aspect ratio for strong eye blink detection based on facial landmarks. NCBI. Retrieved February 15, 2023, from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9044337/