Portable smart attendance system on Jetson Nano

Edward Yose[1], Victor[1], Nico Surantha[1,2]

[1]Department of Computer Science, BINUS Graduate Program-Master of Computer Science, Bina Nusantara University, Jakarta, Indonesia

[2]Department of Electrical, Electronic, and Communication Engineering, Faculty of Engineering, Tokyo City University, Tokyo, Japan

Article Info

Article history:

Received Feb 20, 2023
Revised Aug 8, 2023
Accepted Oct 13, 2023

Keywords:

Attendance system
Graphical user interface
Jetson Nano
Masked face recognition
Neural network

ABSTRACT

The masked face recognition-based attendance management system is an important biometric-based attendance tracking solution, especially in light of the COVID-19 pandemic. Despite the use of various methods and techniques for face detection and recognition, there currently needs to be a system that can accurately recognize individuals while they are wearing a mask. This system has been designed to overcome the challenges of widespread mask use, impacting the effectiveness of traditional face recognition-based attendance systems. The proposed system uses an innovative method that recognizes individuals even while wearing a mask without the need for removal. With its high compatibility and real-time operation, it can be easily integrated into schools and workplaces through an embedded system like the Jetson Nano or conventional computers executing attendance applications. This innovative approach and its compatibility make it a desirable solution for organizations looking to improve their attendance-tracking process. The Experimental results indicates using maximum resources possible the execution time needed on Jetson Nano is 15 to 22 seconds and 14 to 18 seconds respectively and the average frame capture if there are at least one face detected on Jetson Nano is 3-4 frames.

Corresponding Author:

Nico Surantha
Department of Computer Science, BINUS Graduate Program-Master of Computer Science
Bina Nusantara University
Jakarta, Indonesia
Email: nico.surantha@binus.ac.id

1. INTRODUCTION

The management attendance system is crucial and widely used in many organizations, such as schools to keep track of their student's attendance [1] and companies to track down the attendance of their employees [2]. In general, there are two types of attendance systems: traditional attendance system and automatic attendance system [1], [3]. Traditional attendance systems are inefficient as they take more time and have a higher likelihood of errors caused by proxies and impersonation [4], like manually filling out their attendance sheet. However, automatic attendance systems are proven to be more efficient and effortless. Moreover, these systems provide a quick and accurate way of tracking attendance using various identification techniques such as biometric and radio-frequency identification [5].

Nowadays, the process of tracking attendance has been revolutionized by using biometrics. According to Jain *et al*. [6], biometrics is a pattern recognition system programmed to acquire individual biometric data, extracting a feature based on captured data and comparing this feature against the template data in the database [7]. The use of biometrics has been widely used recently, like face recognition, iris recognition [8], fingerprint [9], and speech recognition [10].

While among all methods of biometrics, one of the particular methods that do not require physical interaction is face recognition. Biometric-based face recognition attendance systems are widely used in educational institutions and organizations to improve the accuracy and efficiency of attendance tracking. This proves that face recognition has become a popular biometric application, as demonstrated by Siswanto *et al.* [11], despite limitations such as its relative lack of uniqueness and reduced performance compared to other biometrics. The critical advantage of face recognition is that it enables identification without requiring physical contact. Face recognition is done in two ways: first, to detect faces, and second, to match with the database. It involves capturing an image from an attached camera where the face must be detected by a bounded box, crop, and match with the existing identities based on available databases to identify the person.

However, due to the COVID pandemic since 2019, individuals are required to wear masks. Which poses a challenge for standard face recognition-based attendance systems as the face is partially obscured and making traditional face recognition harder to detect faces. In addition to using masks, other factors affect face recognition detection ability and accuracy such as low resolution, noise, illumination, pose variation, expressions, aging, and plastic surgery [12].

This paper contains face detection to confirm the presence of a human face. The system also implements mask and non-mask classifiers. The system then performs face alignment to accurately estimate the face's position, even if the face was masked. Subsequently, the face recognition system will scan and capture the face and compare it with the database. Finally, the system will record the individual's attendance if a match is found.

Intending to improve the performance, accuracy, and efficiency of attendance tracking, this paper aims to develop an attendance management system. The system aims to recognize individuals even when they are wearing a mask without requiring them to remove it. In addition, students or employees can go in without additional involvement except in the room, so the biometrics system is fully autonomous with minimum physical interaction.

This paper condenses the work focuses on an IoT platform to monitor student attendance by implementing an embedded system using Jetson Nano that can be activated automatically when the system goes online. To test if the system could be implemented in Jetson Nano with a minimum acceptable capture rate and accuracy since Jetson Nano has limited capabilities and resources. The system worked when someone appeared in front of the camera system, and the system captured every frame of the student's facial image to validate whether the frame had a face or not, even if it was masked. If there is a face, the system looks for the region of interest (ROI), crops the image face, and analyses it by matching it with the database. If the cropped face matches and gets a face extraction feature to compare with databases, if it matches it automatically will be marked by the system that matches with the database and uploads the mark attendance into the integrated database.

2. LITERATURE REVIEW

In our study of attendance management systems, we have reviewed several papers that propose different methods for tracking attendance using biometrics, specifically face recognition. One of the papers by Naen *et al.* [9] presents a study on the development of a biometric-based fingerprint attendance management system that utilizes artificial intelligence optimization in the cloud. Naen *et al.* [9] proposed a system to improve the efficiency and accuracy of attendance tracking in educational institutions and reduce errors. The study's results suggest that the proposed system positively impacts the users and is effective in managing attendance.

Another research conducted by Yang and Han [13] describes the design and implementation of a face recognition attendance system based on real-time video processing. The system aims to improve the accuracy and efficiency of attendance management in schools and reduce the truancy rate among students. The system uses a combination of programming languages, including Python, Java, and C++, and uses the OpenCV library for face recognition. In this paper, the system achieved an accuracy rate of around 82% when tested in two colleges. Both papers aim to improve the accuracy and efficiency of attendance management in educational institutions.

The third paper conducted by Anzar *et al.* [14] proposes a new system for managing attendance in virtual classrooms using a combination of deep learning and artificial intelligence. The system, called random interval attendance management system (RIAMS), incorporates a customized face recognition module and ancillary submodules for checking students' responses to CAPTCHAs and unique identification queries. The system operates at random intervals to ensure that students cannot predict when attendance will be taken. Additionally, the system employs an intelligible and adaptive weighting strategy to make attendance decisions more efficient and reduce discrepancies. The authors note that this is the first system of its kind, and further testing with larger virtual classrooms is needed to evaluate its effectiveness further.

The fourth paper conducted by Chen and Li [15] presents the design and development of an attendance management system for small and medium-sized enterprises that combines radio frequency identification devices (RFID) and face recognition technologies. The system is intended to improve the accuracy and efficiency of attendance management by using a mixed authentication method. The system is reported to have a high accuracy rate of face recognition (over 99.8%), fast processing time (recognition time for a single face image is less than 1 second), and strong adaptability to different fields with only minor modifications. Additionally, the system has high reliability, can handle concurrent processing without stalling, and is convenient to operate. The system is designed to provide essential core functions of face recognition and attendance without the high cost or complex configuration. It can be used in small and medium-sized enterprises to efficiently manage attendance information and promote the statistical efficiency of attendance data.

The fifth paper conducted by Kariapper *et al.* [16] published in systematic reviews in Pharmacy, a two-factor verification method is proposed to improve the attendance system in organizations. The first verification is done using RFID and IoT technology, where students must place their RFID tag in front of the reader to confirm their presence. The second verification is done using a machine learning algorithm, where a camera takes a picture of the student and verifies their face with their RFID tag. The system also includes a GSM module that sends notifications to the parents of absent students. A prototype of the system was developed, and the results showed that it effectively and efficiently addressed the attendance issues and had a high level of security, reliability, faster performance, and cost-effectiveness. The study also highlighted the use of multiple previous studies using RFID and machine learning techniques to improve attendance systems.

The sixth paper titled conducted by Suhaimin *et al.* [17] was developed in response to the current COVID-19 pandemic, where the widespread use of face masks has become the norm. It aims to develop a framework for a system that can detect masks in real-time and recognize faces for attendance tracking. The paper also evaluates the performance of two different face recognition techniques, eigenfaces and local binary pattern histograms (LBPH), to determine which method best suits the proposed system.

The seventh paper conducted by Cárdenas *et al.* [18] explored the use of deep learning in a proposed model to improve face recognition rates in low-resolution scenarios. This research was carried out using two datasets, the Caviar and UCSP databases, to test the proposed model's performance compared to existing methods. The study showed a significant increase in accuracy and a low rate of false positives in low-resolution videos.

According to the study by Sutabri *et al.* [19] propose an automatic attendance system for university students using face recognition based on deep learning. This paper utilized convolutional neural networks (CNN) to detect faces, Dlib's CNN or deep metric learning for facial embedding, and K-NN to classify faces. The system demonstrated the ability to accurately recognize the faces of attending students, and the attendance data was automatically saved.

From this literature review, there are various methods for implementing facial recognition-based attendance systems. The summary of literature review mentioned in Table 1 with every device used on each publication. However, these methods have yet to be implemented on Jeston Nano embedded system. Therefore, the design aims to implement a smart attendance system with face recognition by using a Jetson Nano with mask and not mask classification to recognize faces with and without a mask.

Table 1. Literature review

Reference	Biometric method	Hardware devices
[9]	Fingerprint	Fingerprint scanner
[13]	Face recognition	Camera and computer device
[14]	Face recognition	Student's video camera used to monitor student attendance
[15]	Face recognition (ArcFace 2.0)	Computer with a camera, RFID electronic tag, and RFID reader
[16]	Face recognition	Arduino microcontroller is used to control and monitor the system
[17]	Face recognition (Eigenface + LBDH)	A camera of full HD with a 3.6 mm lens and LED ring lights are for light enhancement
[18]	Face recognition	Dahua surveillance video camera with an HD resolution at a rate of 30 fps
[19]	Face recognition	Raspberry Pi NoIR v2 camera used to connect to Raspberry Pi
This paper	Masked face recognition	Jetson Nano development Kit P3450, ubuntu 18.04 LTS

3. PROPOSED SYSTEM

This paper aims to propose an easy and useful smart system of attendance system based on face recognition with mask occlusion to help administration mark attendance intelligently based on real-time video processing using a camera and embedded system attached in front of the room. The system implements a graphical user interface (GUI) for visual application with the support of PyQt5. GUI for this system is shown in Figure 1. which is compatible with most operating systems. This prototype will be in an embedded system with as low-cost production as possible and designed to be implemented in various operating systems. This paper also compares the capability and performance between embedded systems and default operational systems like windows and Linux in personal computers.

Figure 1. System GUI

3.1. General system design

The system created is a GUI-based application system. The first to be developed in designing the system is to plan the architecture system. The complete proposed architecture system is shown in Figure 2. For starters, the system turns on the power supply and automatically launches the script to make the smart attendance application execute. Next, the administration could edit additional information like room numbers in the settings option and register faces to store in the database. Finally, the new faces are applied after the system is rebooted.

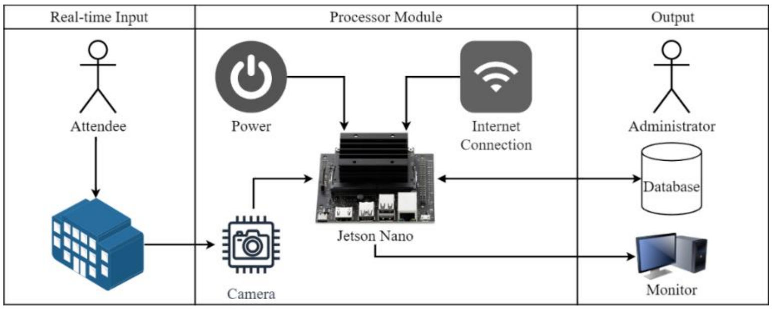


Figure 2. Proposed hardware system architecture

After the system booted, the camera records and automatically scans faces if available and checks if the scanned faces are valid according to their databases with or without masks. Every scanned face will be cropped and stored locally into a temporary image, and aligned image will be based on the face alignment algorithm. Face alignment is a function of finding the best localization, normalization, and distances of the face and estimating the position of the face and face extraction based on a captured image. All extraction is stored in face embeddings. Face embeddings are a 128-dimensional vector stored as a numeric vector shaped like a trained face. If all embeddings are matched with the dataset, therefore the faces are marked as valid [20], [21]. If the face were validated, the system would send a mark attendance to the attendance database to mark students or employees at the exact time. The administrator can see the attendance report in the section report. The complete system workflow is shown in Figure 3.

3.2. Hardware design

This section outlines the hardware design with a selection of electronic equipment and all integrated components needed. All presentations of hardware specifications are shown in Table 2. Meanwhile, the hardware design of the attendance system is shown in Figure 4. For the processing module, we use the Jetson

Nano model P3450. This processor uses an Ethernet LAN cable for internet access, a camera USB Xiaovv 1080p usable for Jetson Nano to capture the facial image, a power supply of either use micro-USB or DC adapter with a minimum of 5 V/2 A, and last one is SD micro card that is used as a storage.

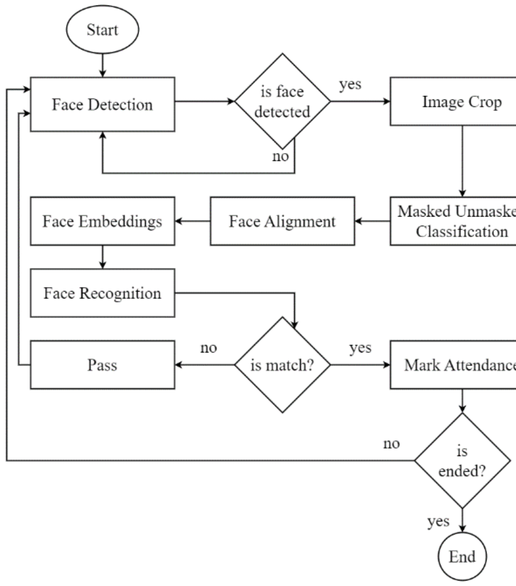


Figure 3. System workflow

Table 2. Jetson Nano hardware devices

Device name	Specifications
Jetson Nano	Model P3450 with NVIDIA Maxwell GPU, quad-core ARM Cortex A-57 Processor, 4 GB LPDDR4 memory, Micro-SD card storage
Camera	Camera USB Xiaovv 1080p
Power supply	Micro-USB or DC Adapter 5 V/2,5 A
SD card	128 GB storage (recommended minimal 32 GB storage)

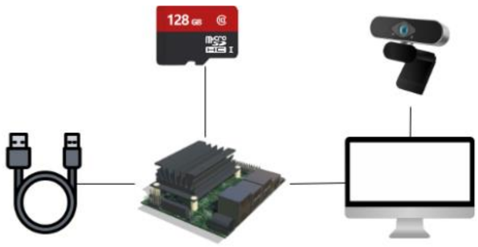


Figure 4. Hardware architecture

3.3. Software design

After hardware design planning, we design the software. Firstly, we create a use case diagram as shown in Figure 5. The administrator started the system in Jetson Nano, which required some time to fully operational. After the system is fully operational, the system is ready to capture using the real-time camera to capture each frame to detect any faces available and recognize them using face recognition.

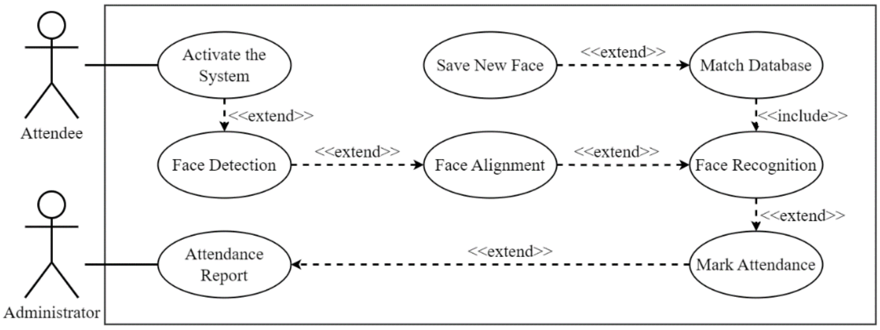


Figure 5. Use case diagram

Face detection is an algorithm to detect whether a human face is essential to many face applications and acts as a necessary preprocessing procedure to face recognition [22]. If there are some human faces, the system will create a box bounding and calculate proportionally if the face is occluded by a mask [23] to make a masked face bound exactly like a non-mask face. After that, proceed into face alignment. The face alignment is purposed to find the best localization, normalization, and distances of the face and estimate the position of the face and face embeddings with the Inception-ResNet-v1 model. All extracted features will be contained in face embeddings with a 128-dimensional numerical vector.

If the face matches and is signed into the database, the system will record attendance based on the name, the recorded date, time, and the designed room ID. All records are stored in the local database, and the administration can see them anytime. The system can record new faces to be marked in the database.

3.4. System setup

The overall hardware system costs roughly US$ 180 including camera compared to personal computers which had cost US$ 800.

- Jetson Nano: the operating system used for Jetson Nano is embedded Linux ARM64 for Jetson Nano development Kit. This operating system can be downloaded online on the NVIDIA Jetson Nano development Kit official website and installation guide. This operating system uses a 128 GB MicroSD Card, but minimal storage can be operated on a 32 GB MicroSD card. After fully installing the operating system, the application used Python 3.7 with a virtual environment setup containing Tensorflow2, OpenCV-Python, and PyQt5. All dependency requirements are used in this research.
- Camera: the webcam camera function is used with the support of OpenCV Python to operate the camera and capture every frame. This paper uses a Resolution of 1280×720 pixels and frame capture rate parameters based on the system resources' maximum capacity available.
- Power supply: to operate Jetson Nano, Jetson Nano provides two power inputs, whether from USB-micro or DC adaptive power. The minimum power to operate is minimum 5 V/2.5 A.

3.5. Implementation of OpenCV-python with PyQt5 GUI

The smart attendance system has been designed with MVC with PyQt5 dependency. PyQt5 is a python binding of the Qt binding. Qt is an open-source free toolkit for creating cross-platform GUI applications [24]. GUI application consists of OpenCV-python video and another layout contains buttons and labels as text and text field.

Implementation of Qt into software applications has been used worldwide. Qt provides tools to build graphical user interfaces to create front-end applications with ease. Implementation Qt with OpenCV is compatible, easy to operate, easy visualization, supports various operating systems, and strong real-time [25]. Also taking advantage of using threading for sending every frame to Qt label to view images in GUI makes utility resources more efficient [26]. The complete cross-functional flowchart for interaction GUI is shown in Figure 6.

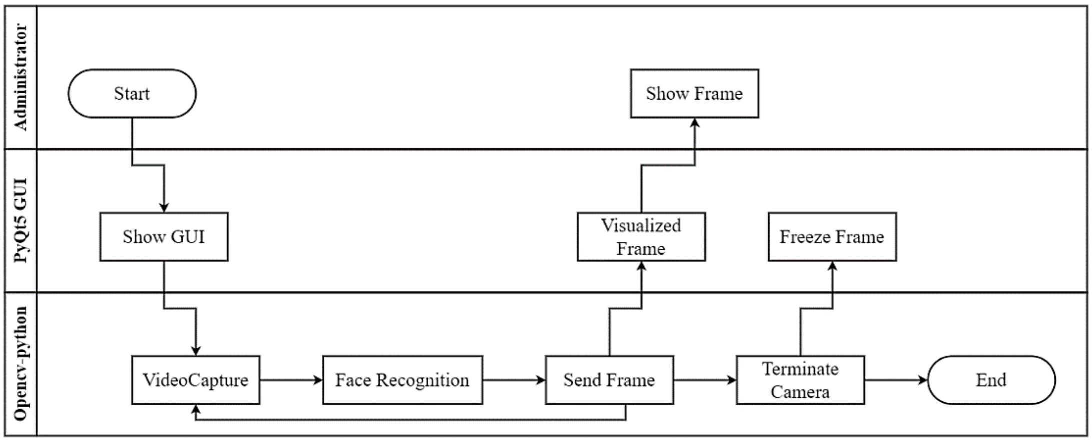


Figure 6. Cross-functional flowchart

OpenCV-python is designed to send every frame captured into the Qt thread label to be viewed on. From Qt's perspective, they will accept every frame received until the system is told to stop receiving them by the toggle button "end system". The application started after the system booted up, using a script to execute on Jetson Nano. The program runs OpenCV-python for camera video and PyQt5 for GUI. After GUI reinitialized OpenCV-python load face detection models and face extraction from the database based on the interception-resnet v1 module into the system and started capturing each frame with masked face recognition. If there are any faces detected, therefore the system makes a box boundary with some additional parameters such as name, mask, and confidence.

3.6. Attendance system

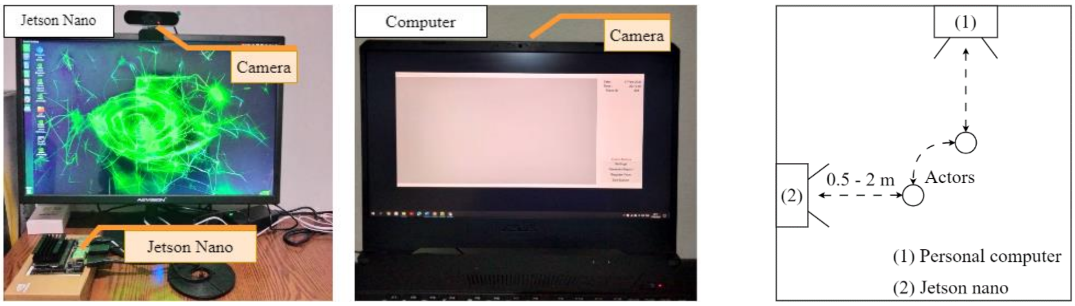
The smart attendance system saves each detected image and stores it in local databases. Attendance parameters consist of attendee's name, room ID, mask/not masked, and time in. The attendee can be informed based on the GUI application with a green boundary box if the attendee uses a mask. The non-mask will be informed with a red boundary. The GUI shows in Figure 1.

4. RESULTS AND DISCUSSION

This section represents the experimental research and implementation of the proposed system. The system implemented using several operating systems is shown in Table 3. For the first time, implement the system for each operating system and implemented the script to execute the system. The system will take real-time video capture and store each frame to evaluate face detection. If there is a face, the system automatically recognizes if the face is using a mask and recognizes face based on database images. Overall, Figure 7(a) shows the prototype of the system hardware and Figure 7(b) shows its implementation in a testing environment using actors and each of hardware systems placement.

Table 3. Operating system resources information and dependency

Information	Laptop ASUS TUF FX505GT	Jetson Nano development kit model P3450
OS	Windows 10 ome SL	Ubuntu 18.04 LTS AMD64
Power supply	20 V/7,5 A	5 V/2,5 A
CPU	Intel i5-9300 CPU Quad-core @2.40 GHz	ARMv8 Quad-core ARM A57 @1.43 GHz
GPU	NVIDIA GTX1650Ti	GPU 128-core Maxwell™ GPU
Memory	8 GB 64-bit DDR5 | 2667 MHz	2 GB 64-bit LPDDR4 | 25.6 GB/s
Storage	512 GB	128 GB
OpenCV	1280×720 Pixel	1280×720 Pixel
TensorFlow	Version 2.11.0	Version 2.11.0
Utility costs	US$ 180	US$ 800



(a) (b)

Figure 7. Test environment; (a) test environment system and (b) test environment area

4.1. Test scenario

The testing scenario involves testing with two parameters to be tested, which are execution time and processing time. The process will involve testing datasets of 30 attendee images captured by the camera from actual faces and video playback. The use of the images is different from images used for train-test models. The scenario was conducted by placing the sample test images in front of a camera with various distances and sample input displayed on another computer's monitor with clear images.

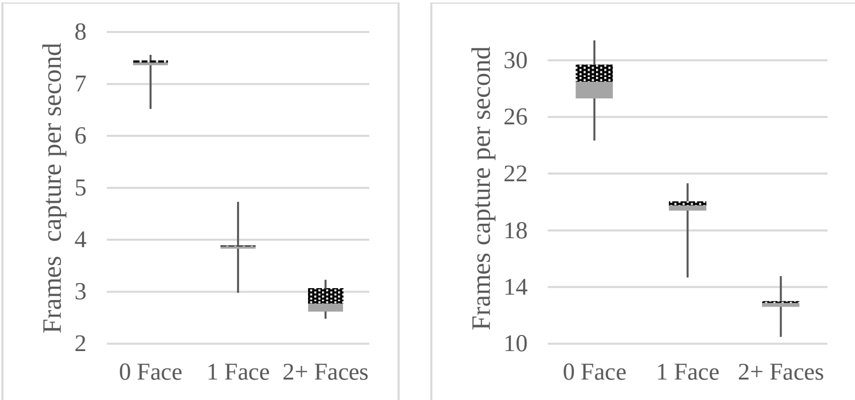
Another parameter for frame capture per second is conducted to know the performance capture rate with the maximum possible system resources. The third parameter for conducting test scenarios is the execution time, which is how much time it takes to execute all functions of building program GUI, threading, Open-CV, and load all embedded databases that are stored for identifying faces even if the face is masked. All scenarios involved using a personal computer and embedded system Jetson Nano as a hardware comparison based on performance, frame capture per second, and execution load time.

4.2. Evaluation method

Evaluation is the process of determining how well the designed system performed. The evaluation will be done based on generated data that is implemented in the system. The report consists of frame capture per second and execution time.

4.2.1. Frame capture per second

In this section, the frame capture per second is reported each second and visualized in a GUI application. The frame capture is determined for comparison based on each operating system resource. The scenario is conducted after the application is online and generates reports under exact resolution (e.g., 1280×720), the same GUI, and stores 30 attendee faces. Figure 8(a) shows the result frame capture per second based on system performance resources used on Jetson Nano and Figure 8(b) shows the result frame capture per second based on system performance resources used on personal computer. The windows OS will use full performance and the Jetson Nano using maximum capacity (MAXN). The result shows for the first 600 seconds with three main circumstances. The first one takes the first three minutes, but no one still faces the camera. After that, for the next three minutes, the system takes the one face that appears in front of a camera and detects precisely one. And for the last four minutes, the system takes more than one face that appears in front of a camera and detects the same number the faces that must appear on a camera. The details of the result are also shown in Table 4.



(a) (b)

Figure 8. Frame capture per second comparison; (a) Jetson Nano and (b) computer

Table 4. Average frame capture per second comparison

Timeline	Jetson Nano (frame/seconds)	Computer (frame/seconds)
0-180	7.39	28.46
181-360	3.87	17.97
361-600	2.96	12.69

The evaluation data shows that if at least one face is detected, the frame rates drop significantly due to the computational need for resources to extract face features such as eyes, nose, and mask if available. Then extracted face needs to be validated for each face by comparing the features of the detected face to the face embeddings in the database, which can be computationally intensive, especially for large databases. This evaluation shows the impact reduction in captured frames per second by 90.9% on Jetson Nano and 58.4% on computer. And it is also applied in many faces detected where at the exact same frame to extract many face features from each detected face and the resources used for computing recognition distributed on each face. Since the system always uses the maximum resources possible, therefore the only way to mitigate exceeded resource usage is to reduce frame capture per second.

4.2.2. Execution load time

In this section, determining how long applications can be fully executed and ready to use are described with graphs and event processes. The scenario divided by four per operating system consists of different saved images in the database, such as 5, 25, 50, and 100 images in the database. The purpose of creating a different variation of the number of attendee images in the database is to determine how the images will be processed and embedded, impacting execution load time. The result of execution time with different scenarios based on the total faces needed to be embedded in the database shows in Table 5.

Table 5. Required time execution program comparison

Stage	Jetson Nano (seconds)	Personal computer (seconds)
OpenCV and other dependencies initiated	0.47	6.24
Load face detection and recognition model	5.27	4.40
Feature extractions and embeddings (5-100 images)	9.27–16.25	3.69–5.69
Total time execution	15.01–21.99	14.33–16.33

The execution time was conducted to test how long the application can be fully functional and how much impact the execution process is if there are many face databases to embeddings. The results are shown in Table 5 that starting all dependencies, including OpenCV and other necessary dependencies in Jetson Nano takes far less time than from a computer. The impact of embedding faces on the database is impactful in Jetson Nano, where from 5 faces overall extractions and embeddings need 9.27 seconds up to 100 faces were conducted in experimental testing with 16.25 seconds. Meanwhile, the computer from 5 faces overall extractions and embeddings require 3.69 seconds, and up to 100 faces need 5.69 seconds. The variation time for embedding face in the database is because every executing program needs to re-embed face from the database that contains cropped images, and each image needs to embed into a 128-dimensional vector stored as a numeric vector shaped like a trained face.

5. CONCLUSION

This paper has proposed a smart attendance system with a comparison using an embedding system and a conventional operating system. with dependency support of TensorFlow, OpenCV-python, and PyQt5. With testing, the execution time using Jetson Nano is 15 to 22 seconds, and windows take 14 to 18 seconds. It depends on the number of images in the database, and the average frame capture rate per second if there is at least one face on the camera is 3-4 frames per second. The impact of performance and capture rate if there are available faces to extract and compare significantly impactful on Jetson Nano. Even with the system using lower cost utilities such as Jetson Nano with utility costs of US$180, the system is still usable for smart attendance system for masked face recognition at minimum acceptance. The prototype of this system is designed for compatibility with various operating systems as long the dependency supports all requirements. Future work for this system is by implementing liveness analysis and looking for more optimizing systems for sustained execution. Therefore, neither student nor employee manipulate the attendance system by using printed faces and other spoofed face material.

ACKNOWLEDGEMENTS

This publication is fully supported by Bina Nusantara University.

REFERENCES

[1] S. Lukas, A. R. Mitra, R. I. Desanti, and D. Krisnadi, “Student attendance system in classroom using face recognition technique,” in *2016 International Conference on Information and Communication Technology Convergence (ICTC)*, Oct. 2016, pp. 1032–1035, doi: 10.1109/ICTC.2016.7763360.

[2] A. B. H. Yanto, A. Fauzi, and N. Indriyani, “Attendance Mobile Application With Face Recognition and Detect Location,” *Jurnal Teknologi dan Open Source*, vol. 5, no. 1, pp. 51–63, Jun. 2022, doi: 10.36378/jtos.v5i1.2187.

[3] O. Shoewu and O. A. Idowu, “Development of attendance management system using biometrics,” *The Pacific Journal of Science and Technology*, vol. 13, no. 1, pp. 300–307, 2012.

[4] S. Bhattacharya, G. S. Nainala, P. Das, and A. Routray, “Smart Attendance Monitoring System (SAMS): A Face Recognition Based Attendance System for Classroom Environment,” in *2018 IEEE 18th International Conference on Advanced Learning Technologies (ICALT)*, Jul. 2018, pp. 358–360, doi: 10.1109/ICALT.2018.00090.

[5] H. Rathod, Y. Ware, S. Sane, S. Raulo, V. Pakhare, and I. A. Rizvi, “Automated attendance system using machine learning approach,” in *2017 International Conference on Nascent Technologies in Engineering (ICNTE)*, Jan. 2017, pp. 1–5, doi: 10.1109/ICNTE.2017.7947889.

[6] A. K. Jain, A. Ross, and S. Prabhakar, “An Introduction to Biometric Recognition,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 1, pp. 4–20, Jan. 2004, doi: 10.1109/TCSVT.2003.818349.

[7] D. Bhattacharyya, R. Ranjan, F. Alisherov, and M. Choi, “Biometric authentication: a review,” *International Journal of u-and e-Service, Science and Technology*, vol. 2, no. 3, pp. 13–28, 2009.

[8] P. S. Bhagat and S. Y. Chincholikar, “Biometric attendance system using iris recognition,” *International Journal for Innovative Research in Multidisciplinary Field*, 2016.

[9] M. F. Naen, M. H. M. Adnan, N. A. Yazi, and C. K. Nee, “Development of Attendance Monitoring System with Artificial Intelligence Optimization in Cloud,” *International Journal of Artificial Intelligence*, vol. 8, no. 2, pp. 88–98, Dec. 2021, doi: 10.36079/lamintang.ijai-0802.315.

[10] M. J. Shah, M. V. Salunkhe, D. J. Saturwar, and M. O. Parab, “Voice Input based Attendance System,” *International Journal of Recent Technology and Engineering (IJRTE)*, vol. 9, no. 1, pp. 1085–1087, May 2020, doi: 10.35940/ijrte.A2147.059120.

[11] A. R. S. Siswanto, A. S. Nugroho, and M. Galinium, “Implementation of face recognition algorithm for biometrics based time attendance system,” *2014 International Conference on ICT For Smart Society (ICISS)*, 2014, pp. 149–154, doi: 10.1109/ICTSS.2014.7013165.

[12] S. Anwarul and S. Dahiya, “A Comprehensive Review on Face Recognition Methods and Factors Affecting Facial Recognition Accuracy,” in *Proceedings of ICRIC 2019: Recent Innovations in Computing*, 2020, pp. 495–514.

[13] H. Yang and X. Han, “Face Recognition Attendance System Based on Real-Time Video Processing,” *IEEE Access*, vol. 8, pp. 159143–159150, 2020, doi: 10.1109/ACCESS.2020.3007205.

[14] S. M. Anzar, N. P. Subheesh, A. Panthakkan, S. Malayil, and H. Al Ahmad, “Random Interval Attendance Management System (RIAMS): A Novel Multimodal Approach for Post-COVID Virtual Learning,” *IEEE Access*, vol. 9, pp. 91001–91016, 2021, doi: 10.1109/ACCESS.2021.3092260.

[15] Y. Chen and X. Li, “Research and Development of Attendance Management System Based on Face Recognition and RFID

Technology,” in *2021 IEEE International Conference on Information Communication and Software Engineering (ICICSE)*, Mar. 2021, pp. 112–116, doi: 10.1109/ICICSE52190.2021.9404086.
[16] R. Kariapper, “Attendance System Using RFID, IoT and Machine Learning: A Two-Factor Verification Approach,” *Journal of Advanced Research in Dynamical and Control Systems*, vol. 12, no. 6, pp. 3285–3297, 2021, doi: 10.5373/JARDCS/V12I6/20202653.
[17] M. S. Md Suhaimin, M. H. A. Hijazi, C. S. Kheau, and C. K. On, “Real-time mask detection and face recognition using eigenfaces and local binary pattern histogram for attendance system,” *Bulletin of Electrical Engineering and Informatics*, vol. 10, no. 2, pp. 1105–1113, Apr. 2021, doi: 10.11591/eei.v10i2.2859.
[18] R. J. Cárdenas, C. A. Beltrán, and J. C. Gutiérrez, “Small Face Detection Using Deep Learning on Surveillance Videos,” *International Journal of Machine Learning and Computing*, vol. 9, no. 2, pp. 189–194, 2019, doi: 10.18178/ijmlc.2019.9.2.785.
[19] T. Sutabri, P. Pamungkur, A. Kurniawan, and R. E. Saragih, “Automatic Attendance System for University Student Using Face Recognition Based on Deep Learning,” *International Journal of Machine Learning and Computing*, vol. 9, no. 5, pp. 668–674, Oct. 2019, doi: 10.18178/ijmlc.2019.9.5.856.
[20] A. Jha, “Classroom attendance system using facial recognition system,” *The International journal of Mathematics, science, technology and Management*, vol. 3, no. 3, pp. 4–7, 2007.
[21] F. Schroff, D. Kalenichenko, and J. Philbin, “FaceNet: A unified embedding for face recognition and clustering,” in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2015, pp. 815–823, doi: 10.1109/CVPR.2015.7298682.
[22] Z.-Q. Zhao, P. Zheng, S.-T. Xu, and X. Wu, “Object Detection With Deep Learning: A Review,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 11, pp. 3212–3232, Nov. 2019, doi: 10.1109/TNNLS.2018.2876865.
[23] S. Yang, P. Luo, C. C. Loy, and X. Tang, “Wider face: A face detection benchmark,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 5525–5533.
[24] M. Fitzpatrick, *Create GUI Applications with Python & Qt5 (PyQt5 Edition): The Hands-on Guide to Making Apps with Python*. 2020.
[25] L. Sun, C. Ge, and Y. Zhong, “Design and Implementation of Face Emotion Recognition System Based on CNN Mini_Xception Frameworks,” *Journal of Physics: Conference Series*, vol. 2010, no. 1, p. 012123, Sep. 2021, doi: 10.1088/1742-6596/2010/1/012123.
[26] J. Willman, *Modern PyQt*. Berkeley, CA: Apress, 2021, pp. 163–208.

BIOGRAPHIES OF AUTHORS

Edward Yose is currently pursuing the bachelor’s and master’s degrees in computer science with BINUS University. His research interests on computer vision, artificial intelligence, and machine learning. He can be contacted at email: edward.yose@binus.ac.id.

Victor is currently pursuing the bachelor’s and master’s degrees in computer science with BINUS University. His research interests on computer vision and machine learning. He can be contacted at email: victor006@binus.ac.id.

Nico Surantha received the B.Eng. and M. Eng. degrees from the Institut Teknologi Bandung, Indonesia, in 2007 and 2009, respectively, and the Ph.D. degree from the Kyushu Institute of Technology, Japan, in 2013. He currently works as a full-time Lecturer with the Department of Electrical, Electronic, and Communication Engineering, Tokyo City University, Japan. He is also a Lecturer with the Department of Computer Science, BINUS Graduate Program, Bina Nusantara University, Indonesia. His research interests include ubiquitous computing, intelligent systems, the internet of things, health monitoring, and system on chip design. He can be contacted at email: nico.surantha@binus.ac.id.