Angkasa Jurnal Ilmiah Bidang Teknologi
ISSN : 2085-9503 (Print)
ISSN : 2581-1355 (On Line)
Accredited Third Grade by Kemenristekdikti, Decree No: 10/C/C3/DT.05.00/2025
DOI : 10.28989/angkasa.v17i2.3103

Face Recognition Application for Lecture Attendance Using FaceNet

Uke Kurniawan Usman[1*], Ricky Marcelino Rafieldo[2], Heru Syah Putra[3]
[1,2]Department of Telecommunication Engineering, Telkom University, Indonesia
[3]Department of Electrical Engineering, Telkom University, Indonesia

Article Info

Article history:

Received October 2, 2020
Accepted October 10, 2020
Published November 1, 2020

Keywords:

Attendance System
Face Recognition
Google ML Kit
Facenet
Accuracy

ABSTRACT

Student Attendance Systems that still rely on manual or semi-manual methods are often prone to recording errors and misuse, which can disrupt the academic evaluation process. Facial recognition technology can offer a solution by enabling the unique identification of individuals based on facial features and allowing automatic, real-time attendance recording. This study aims to develop a facial recognition attendance system using Google ML Kit and FaceNet in a mobile application. Testing was conducted under various conditions, including different distances, lighting, and the use of accessories, to evaluate the system's reliability in real-world scenarios. The results show 100% accuracy at distances of 50 cm, 100 cm, and 150 cm, although recognition time slowed from 1.328 seconds at 50 cm to 1.963 seconds at 150 cm. Accuracy decreased in low-light conditions, and the simultaneous use of accessories such as hats and glasses reduced accuracy to 78.75%. Additionally, the system exhibited a False Acceptance Rate (FAR) of 10% when tested with faces outside the database. Overall, the developed facial recognition system demonstrates high accuracy under ideal conditions but still requires optimization for varying environmental conditions.

Corresponding Author:

Uke Kurniawan Usman
Department of Telecommunication Engineering,
Telkom University,
Jl. Telekomunikasi No.1, Bandung, Jawa Barat 40257, Indonesia.
Email: *ukeusman@telkomuniversity.ac.id

1. INTRODUCTION

Attendance systems implemented in many universities still often rely on manual or semi-manual methods. These methods have several weaknesses, one of which is the potential for errors in recording by lecturers. In addition, manual methods are frequently misused by students who ask their friends to sign in on their behalf, leading to inaccurate attendance data [1]. This is certainly detrimental, especially in academic evaluation and data management processes that require accuracy. The time-consuming nature of manual attendance taking can also reduce the efficiency of lecture time [2], [3].

Student attendance in lectures has a very important role in the learning process, because it reflects the commitment and sincerity of students to attend lectures [4]. Research also shows that good attendance can have a positive impact on learning, increase interaction between students and lecturers, and contribute to improving student academic performance [5], [6], [7]. Therefore, a more effective attendance system is needed to support the success of the educational process in higher education.

Several recent studies have developed attendance systems using biometric technologies, such as facial recognition, to improve accuracy and efficiency. For example, research by [8] utilized the FaceNet algorithm which successfully achieved facial recognition accuracy of up to 99.63% under controlled conditions. However, challenges such as changes in lighting, distance, and the use of accessories are still obstacles in implementation in real environments. In addition, the integration between facial recognition-based attendance systems with mobile platforms and academic databases still needs further development to ensure compatibility and reliability.

Based on these challenges, this study aims to develop a face recognition-based attendance system that can function in real-time, overcome problems such as lighting changes and the use of accessories, and integrate with existing academic systems. In addition, this study will also evaluate the level of accuracy, processing speed, and False Acceptance Rate (FAR) of the developed system.

2. RESEARCH METHODS

2.1. Face Recognition

Face recognition is an evolution of face detection technology, enabling computers to recognize or identify a person's identity by combining images captured from a camera with facial data already registered in the system [9]. Generally, there are five main stages in the face recognition process: detection, position identification, normalization, encoding, and comparison [10]. This process allows the system to distinguish faces from the existing database.

2.2. FaceNet

FaceNet is a system that maps facial images into Euclidean space, where the distance between points directly reflects the degree of facial similarity. FaceNet uses a Deep Convolutional Neural Network (DCNN) architecture called Inception-ResNet [11]. This architecture combines inception modules and residual connections, enabling the model to capture complex facial features more efficiently. These features are then mapped into a low-dimensional embedding space, where similar faces have embeddings that are close to each other within that space [8].

FaceNet model consists of 447 neural network layers and takes input images of size 160x160 pixels with three color channels (RGB) [12]. Additionally, this model is a pre-trained model with a total of 23 million parameters and generates facial embedding vectors with a dimension of 512 [13]. Figure 1 summarizes the architecture of the pre-trained model, including information on various layers, output sizes, number of parameters, and connections between layers.

Layer (type)	Output Shape	Param #	Connected to
input_layer(InputLayer)	(None, 160, 160, 3)	0	-
Conv2d_1a_3x3 (Conv2D)	(None, 79, 79, 32)	864	input_layer[0][0]
Conv2d_1a_3x3_BatchNorm(BatchN	(None, 79, 79, 32)	96	Conv2d_1a_3x3[0][0]
...			
Dropout(Dropout)	(None, 1792)	0	AvgPool[0][0]
Bottleneck (Dense)	(None, 512)	917,504	Dropout[0][0]
Bottleneck_BatchNorm (BatchN	(None, 512)	1,536	Bottleneck[0][0]

Total params: 23,497,424
Trainable params: 23,467,824
Non-trainable params: 29,600

Figure 1. FaceNet Architecture

In general, FaceNet performs feature extraction as seen in Figure 2 by feeding a number of facial images into a deep learning architecture consisting of multiple layers and L2 normalization, which then produces facial embedding. In addition, FaceNet also uses a triplet loss function to train the model to produce more effective embedding. Triplet loss is designed to ensure that the same face has embeddings that are close to each other, while different faces have embeddings that are further apart.

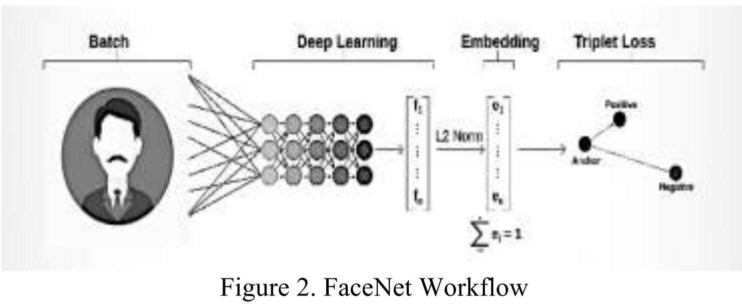

Figure 2. FaceNet Workflow

Once the model is trained, FaceNet can be used to match faces and verify identities. Face matching is done by comparing new face embeddings with those in the database using the Euclidean distance calculation, as described in equation 1 [13]. A threshold is set to determine whether two faces are considered the same or different. If the distance between two embeddings is smaller than the threshold, then the two faces are considered identical. Conversely, if the distance is greater than the threshold, then the faces are considered different

$$d(p,q) = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2} \tag{1}$$

with: p_i = the i-th element of the detected embedding
q_i = the i-th element of the face embedding that is registered
n = the number of elements in the embedding.

2.3. Tensorflow

TensorFlow is a comprehensive open-source platform for machine learning. It offers a broad and flexible ecosystem of tools, libraries, and community resources, enabling researchers to drive progress in ML and making it easy for developers to build and deploy machine learning (ML)-based applications [14]. TensorFlow is one of the most popular deep learning libraries. In the field of deep learning, neural networks have achieved significant success and gained widespread recognition across sectors [15]. TensorFlow also provides additional features such as TensorFlow Lite (for mobile devices) and TensorFlow.js (for web-based applications).

3. RESULTS AND DISCUSSION

3.1. Research Results

In this section, Figure 3 shows the design of a mobile application-based attendance system involving several main components. Students register by entering personal information and facial photos, which are then processed and stored in a database. When attendance begins, students take attendance using facial recognition, and the system matches the detected face with the data in the database. If it matches, attendance is recorded and stored, accessible by lecturers and students to check attendance history.

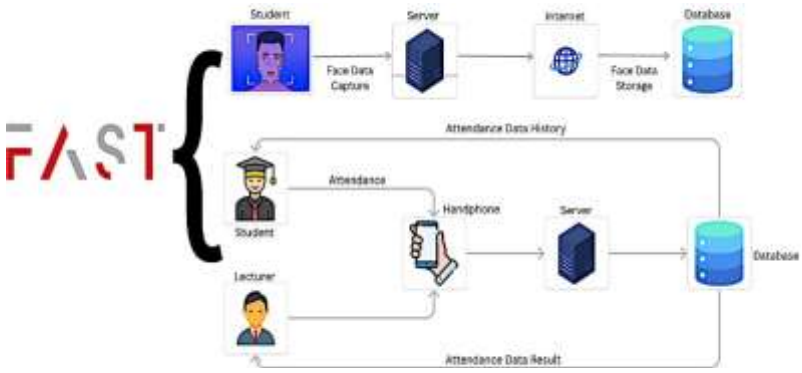

Figure 3. System Design Overview

3.1.1. System Flow Diagram

Figure 4 shows a flowchart of the facial registration process, starting with students entering personal data such as name, student ID number, and study program. The system then activates the camera to take a photo of the student's face. After the photo is taken, preprocessing is performed to improve the image quality, before the facial feature extraction stage. The extracted features are then stored along with the student's identity information in a database, completing the registration process. One of the aspects that reviewers assess is the quality of the research results in this article. To demonstrate that the research results are of high quality and contribute to science, this section should provide a comparison between the results of this study and those of other studies, particularly those mentioned in the Introduction section. This section must cite other research findings.

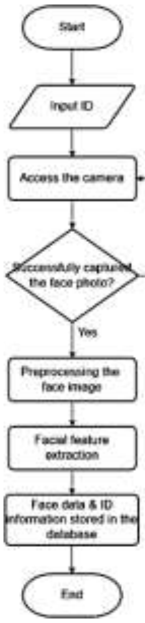

Figure 4. Registration process flowchart

Figure 5 shows a flowchart of the lecture attendance process. The process begins by activating the camera to initiate facial recognition. The camera then detects the facial image and proceeds to the recognition and verification stage. The system will compare the detected face with the data in the database. If the face matches the stored data, the system will record the student's attendance and the attendance process is complete. However, if the face is not recognized, the system will display a message that the face is not detected and the student is asked to repeat the facial recognition process.

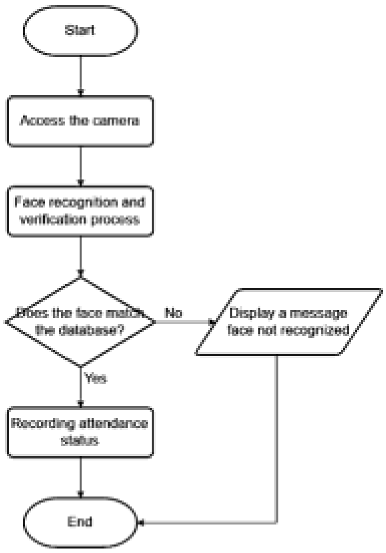

Figure 5. Attendance process flowchart

3.1.2. Testing and Analysis of Facial Recognition Accuracy

Discussion of the results of the system testing scheme is designed to evaluate the performance of the face recognition application in the context of lecture attendance. The main objective of this test is to assess the extent to which the system is able to accurately detect, recognize, and record student attendance, taking into account various real conditions that may occur in a lecture environment. The testing involved 40 registered students who served as the primary sample. Throughout the entire process, 1,458 test data points were obtained, covering a wide range of conditions. The variations in conditions applied included bright and dim lighting, image capture distances at three points (50 cm, 100 cm, and 150 cm), and the use of accessories such as glasses and hats, which often act as distracting factors in the facial recognition process. With this combination, the system was thoroughly tested in scenarios that closely resembled real-world conditions often encountered in classrooms and in the field.

The testing focused on three main parameters. First, facial recognition accuracy, which was measured by calculating the percentage of success of the system in recognizing students' faces according to the data stored in the database. Second, system response speed, which is the duration of time required for the system to detect faces, process data, and provide recognition results. This parameter is important for assessing the efficiency of the system in situations where many students are present at the same time. Third, the system failure rate, which records the number of cases where the system fails to recognize faces or provides incorrect results.

$$Accuracy = \frac{\text{Number of Successful Trials}}{\text{Total Number of Trials}} \times 100\% \quad (2)$$

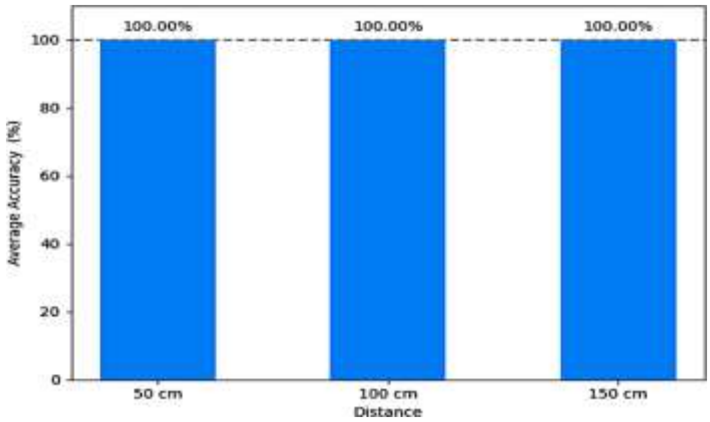

Figure 6. Average accuracy based on distance

Figure 6 shows the average accuracy graph of all trials based on the distance tested. At a distance of 50 cm, the system recognized the participant perfectly in every trial. Testing at 100 cm also yielded satisfactory results, indicating that the application can identify faces well at medium distances. Testing at a distance of 150 cm maintained 100% accuracy, without any failures, indicating that the application can maintain the quality of facial recognition even at longer distances. Overall, the 100% accuracy results recorded across all trials confirm that the application is very effective in detecting and recognizing students without errors at the various distances tested.

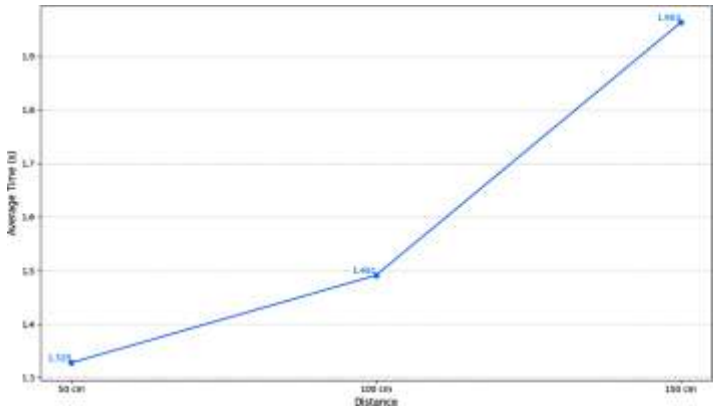

Figure 7. Average recognition speed based on distance

Based on the data in Figure 7, the time required by the system for face recognition varies at different distances. At a distance of 50 cm, the average recognition time was recorded at 1.328 seconds, indicating high efficiency due to the clear facial image quality and optimal lighting. However, as the distance increases, the efficiency decreases. At a distance of 100 cm, the recognition time increases to 1.491 seconds, and at 150 cm to 1.963 seconds, due to the deterioration of the image quality, reduced facial sharpness, and less than optimal lighting. Although the system performs well at close range, its efficiency decreases at further distances, so it is recommended that users remain at close range to ensure fast and efficient face recognition.

3.1.3. System Failure Rate and Analysis

The failure rate of the system is an important aspect to test the reliability of the facial recognition application in various real conditions. This test aims to identify factors that can cause failure, both from external factors and system limitations. Testing is carried out by considering lighting, the use of additional objects such as glasses and hats, and testing faces that are not registered in the database. In lighting, it is tested in bright and dark conditions, because poor lighting can interfere with facial recognition. Additional object testing aims to see how the system overcomes the challenge of covered faces, while testing with faces outside the database aims to assess the system's ability to distinguish between registered and unregistered faces without making recognition errors.

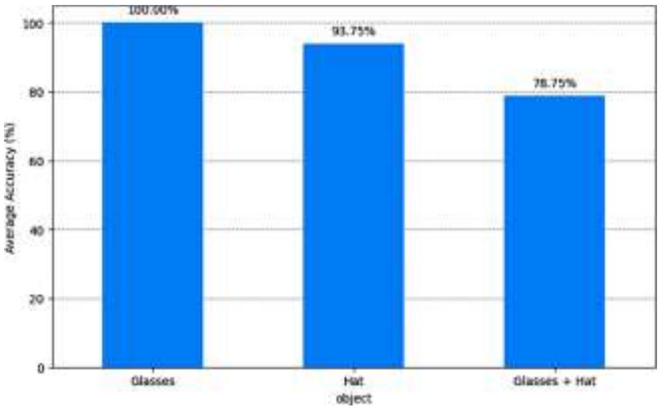

Figure 8. Average accuracy by object type

Figure 8 shows the average accuracy graph based on the type of object used. When wearing glasses, the system achieves 100% accuracy, indicating that the glasses do not block important facial features. However, wearing a hat decreases the accuracy to 93.75%, because the hat covers part of the face. The combination of glasses and a hat caused the accuracy to decrease to 78.75%, because they cover more areas of the face, such as the eyes and forehead, which reduces the information for recognition. This highlights the difficulty of the system in recognizing faces when many objects cover the face simultaneously.

Table 1. Test results with lighting variations

Participants	Light Conditions	Time (s)	Status	Accuracy (%)
1	Bright	1,16	Recognized	100
2	Bright	1,22	Recognized	100
3	Bright	1,15	Recognized	100
4	Bright	1.52	Recognized	100
1	Dark	-	*Unknown/ Not Recognized*	0
2	Dark	3,12	Recognized	100
3	Dark	3,68	Recognized	100
4	Dark	-	*Unknown/ Not Recognized*	0

Based on the data in Table 1, it can be seen that the facial recognition system shows optimal performance in bright lighting conditions. All participants were successfully recognized with 100% accuracy and relatively fast recognition times, ranging from 1.15 to 1.52 seconds. This proves that good lighting allows the system to capture facial features clearly so that the identification process runs consistently and efficiently. Conversely, under low-light conditions, the system's performance declined significantly. Two participants were not recognized at all (0% accuracy), while the other two were recognized but with much longer processing times of 3.12 seconds and 3.68 seconds. These results indicate that low lighting degrades the quality of the facial images captured by the camera, making it difficult for the system to accurately extract facial features. Overall, this comparison confirms that bright lighting directly contributes to the system's speed and accuracy, while low lighting is the primary factor hindering facial recognition performance.

Table 2. Test results with facial objects outside the database

Participants	Recognized	Unknown/ Not Recognized	Accuracy (%)
1	-	Yes	100
2	-	Yes	100
3	-	Yes	100
4	-	Yes	100
5	-	Yes	100
6	-	Yes	100
7	-	Yes	100
8	Yes	-	*0 (FAR case)*
9	-	Yes	100
10	-	Yes	100

$$FAR = \frac{False\ Positives}{False\ Positives + True\ Negatives} \quad (3)$$

$$FAR = \frac{1}{1+9} = \frac{1}{10} = 0.1\ atau\ 10\%\ (100\%)$$

Based on the test results in Table 2, the facial recognition system still shows limitations, especially in terms of consistency in recognizing unregistered faces. A False Acceptance Rate (FAR) value of 10% indicates the possibility that faces that should be rejected are instead accepted as valid identities. Although the system works well under ideal conditions, optimization is still needed to make it more reliable in various real-world situations. As a recommendation, the system should be equipped with preprocessing techniques to improve image quality (e.g., in low-light conditions) and trained using more diverse facial data to adapt to environmental variations. From a technical perspective, several improvements can be considered. First, low-light issues can be addressed using image enhancement methods or additional sensors such as infrared. Second, accuracy degradation caused by accessories like hats and glasses can be minimized by applying more adaptive deep learning models to occlusion. Third, to reduce the FAR value by 10%, optimization of the threshold, implementation of liveness detection, or even combination with other biometrics such as voice is required. Finally, to make the recognition process more efficient on mobile devices, the system can be optimized through model quantization techniques or the use of a lighter CNN architecture.

4. CONCLUSION

Based on the test results with various parameters, the facial recognition system showed optimal performance in several aspects. The facial recognition accuracy reached 100% at distances of 50 cm, 100 cm, and 150 cm, although there was a slight decrease in the quality of the facial image at a greater distance. The facial recognition time increased with increasing distance, with an average of 1.328 seconds at a distance of 50 cm, 1.491 seconds at 100 cm, and 1.963 seconds at 150 cm, but remained within acceptable limits. Testing in bright lighting conditions showed faster recognition times, while in dark lighting, accuracy decreased. Wearing a hat reduced accuracy to 93.75%, and the combination of a hat and glasses reduced accuracy to 78.75%. In addition, the system only managed to recognize one participant who was not legitimate as a person registered in the database, with a False Acceptance Rate (FAR) of 10%. Although the facial recognition accuracy under ideal conditions is very good, there are challenges in recognizing faces with certain attributes or conditions, as well as recognition errors on unregistered faces.

REFERENCES

[1] W. F. Wan Abdul Rahman and N. A. S. Roslan, "The development of a face recognition-based mobile application for student attendance recording," *Journal of ICT in Education*, vol. 10, no. 1, pp. 39–55, Jun. 2023, doi: 10.37134/jictie.vol10.1.4.2023.

[2] Prof. Anand Bali, Hafsa Shaikh, Prachi Zodage, Hussain Harianawala, and Shabbir Kagalwala, "Face Recognition Attendance System," *International Journal of Advanced Research in Science, Communication and Technology*, pp. 479–483, Apr. 2023, doi: 10.48175/IJARSCT-9241.

[3] P. S, H. M, D. V, G. R, and A. R, "An Effective Implementation of Autonomous Attendance System using Convolution Neural Networks," *International Journal of Innovative Technology and Exploring Engineering*, vol. 11, no. 7, pp. 1–6, Jun. 2022, doi: 10.35940/ijitee.G9953.0611722.

[4] P. N *et al.*, "Fast and Reliable Group Attendance Marking System Using Face Recognition In Classrooms," in *2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT)*, 2019, pp. 986–990. doi: 10.1109/ICICICT46008.2019.8993323.

[5] E. O. Akay, K. O. Canbek, and Y. Oniz, "Automated Student Attendance System Using Face Recognition," in *2020 4th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, 2020, pp. 1–5. doi: 10.1109/ISMSIT50672.2020.9255052.

[6] T. Fadelelmoula, "The impact of class attendance on student performance," *International Research Journal of Medicine and Medical Sciences*, vol. 6, no. 2, pp. 47–49, 2018, doi: 10.30918/IRJMMS.62.18.021.

[7] R. Samet and M. Tanriverdi, "Face Recognition-Based Mobile Automatic Classroom Attendance Management System," in *2017 International Conference on Cyberworlds (CW)*, IEEE, Sep. 2017, pp. 253–256. doi: 10.1109/CW.2017.34.

[8] 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)*, IEEE, Jun. 2015, pp. 815–823. doi: 10.1109/CVPR.2015.7298682.

[9] N. Rao K Mahalakshmi, “A Novel Face Detection and Recognition System Using Machine Learning Approaches,” *International Journal of Science and Research (IJSR)*, vol. 12, no. 6, pp. 2730–2738, Jun. 2023, doi: 10.21275/SR23626104114.
[10] L. W. Alexander and S. Sentinuwo, “Implementasi Algoritma Pengenalan Wajah Untuk Mendeteksi Visual Hacking,” *Jurnal Teknik Informatika*, vol. 11, no. 1, 2017, doi: 10.35793/jti.v11i1.16969.
[11] D. Sandberg, “Facenet: Face recognition using Tensorflow.” Accessed: Feb. 07, 2025. [Online]. Available: https://github.com/davidsandberg/facenet
[12] S. Serengil and A. Özpınar, “A Benchmark of Facial Recognition Pipelines and Co-Usability Performances of Modules,” *Bilişim Teknolojileri Dergisi*, vol. 17, no. 2, pp. 95–107, Apr. 2024, doi: 10.17671/gazibtd.1399077.
[13] S. I. Serengil, “Deepface: A lightweight face recognition and facial attribute analysis (age, gender, emotion and race) library for python.” Accessed: Feb. 07, 2025. [Online]. Available: https://github.com/serengil/deepface
[14] TensorFlow Developers, “TensorFlow,” Oct. 25, 2024, *Zenodo*. doi: 10.5281/zenodo.13989084.
[15] B. Pang, E. Nijkamp, and Y. N. Wu, “Deep Learning With TensorFlow: A Review,” *Journal of Educational and Behavioral Statistics*, vol. 45, no. 2, pp. 227–248, Apr. 2020, doi: 10.3102/1076998619872761.