Volume - 7 | Issue - 1 | March 2025

e-ISSN: 2582-4252
4 issues per year
https://doi.org/10.36548/jiip
Indexing
Scopus | GoogleScholar | Crossref | MicrosoftAcademic | ScienceGate | J-Gate
Publisher
Inventive Research Organization
Open Access Journal

Cite score, SJR and SNIP
https://www.scopus.com/sourceid/21101274722

Journal of Innovative Image Processing (ISSN: 2582-4252)
www.irojournals.com/iroiip/

Performance of Face Recognition Machine Learning Algorithms in Attendance Recording System with Limited Training Data

Erico Darmawan Handoyo[1], Sulaeman Santoso[2], Rossevine Artha Nathasya[3]

Faculty of Smart Technology and Engineering, Maranatha Christian University, Bandung, Indonesia.

E-mail: [1]erico.dh@it.maranatha.edu, [2]sulaeman.santoso@it.maranatha.edu, [3]rossevine.an@it.maranatha.edu

Abstract

The effectiveness of the face recognition algorithm is crucial to a digital attendance system. Differences in lighting, camera, room size, and the number of students create a challenging environment for face detection, but one problem that is rarely discussed in face recognition is the problem of limited training data. In attendance recording scenarios, attaining a large number of participant image data might not be feasible due to time constraints or other limitations. The research comprises several key stages, including application requirements analysis, development of the Attendance System application, data collection, development of artificial intelligence (AI) components, and application testing. Within this framework, the study compares the performance of three face-recognition machine learning algorithms (SVM, KNN, and Random Forest) under limited training data conditions. It is demonstrated that the best performing algorithm (SVM with hyperparameter tuning) resulted in the highest accuracy of 0.45. When the data quality is increased by removing some anomalies, the algorithm performs at a higher accuracy of 0.61. The effect of limited data on training algorithms is then examined and discussed further.

Keywords: Face Recognition, Attendance Recording, Computer Vision, Education Technology.

1. Introduction

Attendance recording is one of the common administrative tasks in many institutions, especially in educational institutions. The number of attendance or absences for a student is one of the key indicators of that student's success or failure in class. The traditional approach of roll calling has many disadvantages [1]. The first is that it takes a long time [2]. Time in class that could otherwise be used more effectively is spent on attendance recording. This is made worse if the class has a large number of students. Roll call as one of the most popular attendance recording methods can introduce human error in recording. With scenarios such as when a student fails to answer or the teacher skips a name. If the record is not revised in a short amount of time, it is simply impossible to prove attendance based solely on roll call data. Some form of roll call may involve a distributed attendance record in class that the student may sign

DOI: https://doi.org/10.36548/jiip.2025.3.008
Received: 12.07.2025, received in revised form: 10.08.2025, accepted: 24.08.2025, published: 02.09.2025

to indicate attendance. Bogus or false attendance then becomes a problem, as a student may fabricate attendance by having a false signature signed by their friends. Due to the many problems associated with traditional approaches, a digital solution is then proposed.

One of the digital approaches taken to overcome this problem is using facial recognition. Facial recognition can solve the problem of recording attendance, as bogus or fabricated attendance can be prevented because the student photo serves as digital proof of attendance that is harder to fabricate. Recording can also be done in the least amount of time, and the possibility of human error can be reduced. In our previous studies [3], facial recognition was done with the help of manual face tagging after the candidate's face was detected using Haar Cascade. Our previous study also mentioned some problems with lighting, image size, and resolution that are consistent with other similar research [4]. This method of manual face checking can recognize faces with perfect accuracy. However, if the number of people in the class increase, this manual facial recognition will take significantly longer. The face checking can be delegated to the students but doing this will add more complexity and open the possibility of cheating. This is where automatic facial recognition can play a significant role. Automatic facial recognition identifies an individual from digital data in the form of images or videos. Throughout the years, automatic facial recognition has undergone extensive research and has shown major improvements with a variety of algorithms.

Face recognition with deep learning algorithms is very popular and has shown high accuracy. However, to use deep learning algorithms, a large amount of data needs to be acquired. In scenarios where there is not enough data, it is advisable to use different types of algorithms altogether. This research focuses on classroom attendance recording. In real-life scenarios, the number of students in a classroom may vary from a small class to a large class. Facial recognition would require image data from these students to learn. In research settings, the image-gathering process can be adjusted as needed; however, it may not always be feasible in a classroom setting. In some cases, the number of images for learning might not be adequate or suitable. To solve the problem of low image numbers, image augmentation can be done to create more data for training. However, this process typically requires more computation and a better-quality image source, which is not always available.

Much prior research in attendance recording with face recognition is conducted within a controlled environment with sufficient training data and does not account for cases where data is limited. Limited data may introduce overfitting to learning models; it also makes it harder for the algorithm to learn. While face recognition algorithms may yield high accuracy in controlled environments, the effectiveness of these algorithms may differ when provided with limited data. This research will explore the underdeveloped problem of face recognition under limited-data constraints. To explore this problem, this research will test the viability and performance of several machine-learning face recognition algorithms with limited data. These algorithms are chosen because they are commonly used in face recognition scenarios. After identifying the best-performing algorithm, it will be tested again with varying numbers of training images to demonstrate the impact of data quantity on recognition accuracy. The results of this research are expected to show the impact of limited data on various face recognition software. These algorithms have been shown in other research to be high in accuracy in controlled environments. Hence, this research's unique contribution is in showing the limitations of algorithms within limited data constraints. Since the goal of this research is to test recognition with limited data, data augmentation will not be conducted. This will enable results that better mimic real-world scenarios. The novelty of this research lies in highlighting and quantifying the limitations of machine learning face recognition algorithms in limited data scenarios. These scenarios are common but are rarely investigated. In summary, the research can be framed around the following key questions:

RQ 1: Which face recognition algorithm best suits the needs for attendance recording with limited data?

RQ 2: How does limited training data affect the face recognition algorithm learning process?

2. Related Work

Digital attendance has been researched using various technologies to replace traditional attendance. QR codes [5], RFID cards [6], NFC on mobile phones [7], and more are used to create digital attendance with varying results. These methods are beneficial in some cases, but each has its own limitations. Some approaches require additional items that are not readily available, such as RFID or NFC cards. The problem with an additional item like a card is that it could go missing or be damaged. Other approaches use QR codes; where students scan QR codes to indicate attendance. While this may reduce the time needed for digital attendance, it is very easy to fabricate by distributing the QR code. To avoid additional devices and related problems, many researchers turn to using what is already present on the human body, such as biometrics facial recognition. Among these methods, facial recognition tends to be preferable because it doesn't need additional items from the participant, and the method is non-invasive.

There are a number of works that implement face recognition for digital attendance using varying algorithms. In 2018, Lukas et al [8] used Discrete Wavelet Transform in combination with Discrete Cosine Transform to extract features from the face for recognition. The proposed system achieved 82% accuracy in recognizing faces (121 out of 148 images of 16 students).

In 2020, Serign Modou Bah et.al [4] used an improved Local Binary Pattern algorithm with advanced image processing techniques for face recognition in attendance management systems. The proposed system was able to detect faces with 95% accuracy. This approach uses 181x181pixel images, but did not mention the number of students recognized by the algorithm. Another attempt at face detection in 2020 came from Samridhi et al [1] that are who used Haar Cascade in combination with Gabor Filter and tested face recognition with SVM, CNN, and KNN. This approach achieved 99.27% accuracy with KNN for 70 unique individuals and concluded that SVM was the least accurate algorithm.

In 2021, Nurkhamid et al. used Deep CNN for face recognition. The algorithm achieved 87.7% accuracy for 16 students. Another researcher who applied CNN for face recognition is Bhattacharya [2] who used the Viola-Jones algorithm as a face detection method and CNN for face recognition.

Phul Babu ja [9], Nguyen-tat [10] in 2024, and Bairagi in 2021 [11] used the Haar Cascade / Viola Jones algorithm in combination with the LBPH algorithm for face recognition. These approaches use Haar cascade that is readily available in OpenCV. Phul Babu Ja did not mention the number of people being recognize, Nguyen tat mentioned the number of images use for training and testing but did not mention the number of unique persons being recognize. Bairagi uses 10 unique people for testing during the recognition phase.

In 2023, Andre Budiman et al. [12] compared the Local Binary Pattern Histogram (LBPH) algorithm and Convolutional Neural Network (CNN) for face recognition, finding that CNN was more effective and stable. However, he noted that CNN requires larger datasets, which makes other types of algorithms viable in different cases.

SVM as a face recognition algorithm is also quite popular. Shieh et al. [13] conducted research in 2014 using SVM with PCA, resulting in an accuracy of 91%. Eman Gheni added wavelet transform to PCA and SVM, achieving 92% accuracy [14]. In 2023, Samridhi compared the accuracy of three algorithms: KNN, CNN, and SVM, with K-NN being the most accurate among the three.

Several studies on PCA as a face recognition algorithm have been conducted. Rahul in 2016 [15] used the PCA algorithm in combination with SVD for face detection. This research concluded that the addition of SVD contributes to better accuracy in face detection. In 2015, Priyanka Wagh [16] compared the use of only the PCA algorithm with several other algorithms, showing that the combination of the PCA algorithm with other algorithms can result in higher accuracy.In summary, the reviewed literature demonstrates a variety of algorithmic combinations used for face detection and recognition, with varying levels of accuracy. Nevertheless, many high-accuracy results were obtained using datasets with a small number of individuals and a higher number of training data.

While increasing the number of images for training data is necessary to help training, the small number of unique targets for recognition may inflate accuracy metrics. This may lead to inaccuracies in the practical applications of digital attendance systems, where the number of students can vary from small to large.

Attendance data for this recording system is taken from the teacher's mobile phone. This image will not only contain faces. Thus, before conducting face recognition on the attendance data, the image should be processed with a face detection algorithm. A face detection algorithm works by locating face regions within a given image. This process narrows down the area for the face recognition algorithm and removes non-face regions. There are several algorithms that can be used for face detection, including traditional methods and deep learning [17].

In traditional methods, commonly used algorithms include the Haar Cascade Classifier [18], SVM (Support Vector Machine) [19], Template Matching [20], and more. These methods can be used when the available data is relatively small. When the data is big however and the computing power is sufficient, a deep learning algorithm can be employed.

One of the architectures in deep learning is CNN (Convolutional Neural Network) [21]. Deep learning methods can achieve higher accuracy than traditional methods [17]. This is because deep learning can handle a wide variety of cases and is able to detect faces even under poor lighting conditions. This research will use an existing CNN model as a face detection algorithm to avoid the cost of learning. Figure 1 shows a simple CNN network, in which input is processed through several convolution layers and other processes to create the intended output. This type of CNN is used in the face detection processes of this research. By utilizing the existing CNN model, this research can focus more on the face recognition component.

Some of the work of face detection using deep learning includes the work of Majeed et al, who developed a face detection model using Light CNN. The architecture of this Light CNN consists of 15 layers, including Conv1D, MaxPooling1D, Dense, and Flatten layers. The model achieved an accuracy of 0.998 using a data split ratio of 40% for testing and 60% for training [22]. Desai et al. conducted a study using CNN, PCA, LBPH, and KNN for face detection. The results showed that CNN achieved the highest accuracy compared to the other methods, with an accuracy difference ranging from approximately 15% to 21% [23]. An example of a simple CNN architecture is shown in Figure 1. In building a CNN model, there are several essential layers, including the convolutional layer used to extract image features, the pooling layer to reduce feature dimensions, the activation layer to introduce non-linearity, and the fully connected layer for classification. In designing a CNN model, there is no strict limit on the number of layers; however, the general structure remains the same. One example of a CNN implementation is the face detection library available on PyPI that can be used for facial detection tasks [24]. This implementation of CNN is used in this research for face detection. Once the face region is segmented, the region can then be processed using a face recognition algorithm.

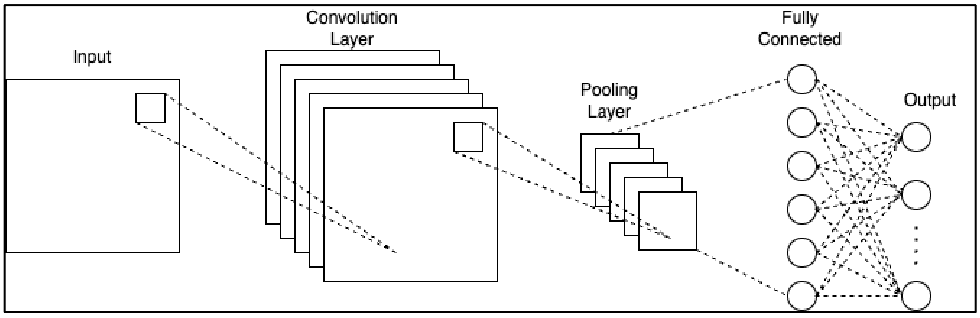

Figure 1. Simple CNN Architecture

Face recognition is the process of recognizing a specific face within a given image. This process requires a target to be defined and then searches for the target or clarifies the existence of the target within the image. Common challenges in face recognition include scaling, pose, illumination, variations, rotations, and occlusions [1]. Face Recognition algorithms for digital attendance need to address these challenges within a reasonable timeframe and with accuracy.

In general, there are two approaches to face recognition: feature-based and image-based [8]. Feature-based methods work by extracting key features from an image, such as the eyes or nose. This includes techniques like Local Binary Pattern Histogram and Scale-Invariant Feature Transform. Image-based methods utilize the entire face region, employing techniques such as eigenfaces (PCA), fisherfaces (LDA), and Support Vector Machines. This research will compare three learning algorithms: Random Forest, SVM, and KNN. SVM is a learning algorithm that generates a hyperplane to separate different classes and is well-suited for high-dimensional data [25]. In face recognition, these hyperplanes will separate each individual. Some comparisons have been made for the performance of SVM and other learning algorithms. For example, Raksit et al. conducted a study on developing a face recognition model using SVM and KNN methods. In the KNN approach, the values of K used were 3 and 5. The results showed that the SVM method achieved better accuracy, outperforming KNN by approximately 14% to 20%. However, in other research KNN or other learning algorithms might yield better results [1].

SVM algorithm formula is as follows:

$$f(x) = \sum_{i=1}^{n} \alpha_i y_i K(x_i, x) + b \tag{1}$$

α_i: Lagrange multipliers (obtained during training)

x_i: class label of the i-th training data

y_i: i-th training data

x : input data to be classified

b : bias

K(x_i ,x) : kernel function, replacing the dot product

The other learning algorithm that is being compared is K-Nearest Neighbors (K-NN). KNN is a classification algorithm based on supervised learning that determines the label of a test data point by measuring its distance to labeled training data. The algorithm classifies the new data by identifying its K nearest neighbors and assigning the label that appears most frequently among those neighbors.

$$d(x, y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{2}$$

$d(x, y)$: The Euclidean distance between two data points x and y

x : The feature vector of the first data point

y : The feature vector of the second data point

n : The total number of features

The third algorithm is Random Forest. Random Forest is a classification algorithm based on ensemble learning that constructs multiple decision trees in parallel. Each tree independently predicts the class of a test data point, and the final classification result is determined through majority voting among all trees. The class that receives the most votes is assigned as the final predicted class for the given input.

After the comparison was completed, the PCA algorithm was added to enhance face recognition for the best performing algorithm. PCA, an algorithm created by Turk et .al, is one of the popular methods for feature selections and dimension reduction. PCA aimed at simplifying data complexity while preserving as much variability as possible from the original dataset PCA calculates the covariance between variables to capture their linear relationships and decomposes them into uncorrelated principal components. One way to determine the optimal number of dimensions is by examining the explained variance. If the cumulative explained variance exceeds 90%, the selected number of components is generally considered sufficient to represent the original data effectively.

PCA in face recognition, often called eigenface, reduces the original data space of the image into a simpler, less dimensional data space. Using a simpler data space reduces the complexity of other parts of the algorithm and lessens the memory needed to process them.

3. Proposed Work

The research methodology applied in this study is illustrated in Figure 2. It consists of several structured stages: (1) Application requirements analysis, where this work builds upon our previous image-based attendance system [3] by incorporating face recognition and reviewing relevant literature; (2) Development of the Attendance System (Mobile Application), which involved updating the existing app to enable data collection for face recognition; (3) Data Collection, carried out by recording students' face images and IDs during regular class sessions over one semester; (4) AI Component Development, in which the dataset was divided into training (75%) and testing (25%) subsets, and three algorithms (SVM, KNN, and Random Forest) were trained, evaluated using standard metrics, and optimized through hyperparameter tuning; and (5) Application Testing, where the system's recognition results were validated against actual student identities.

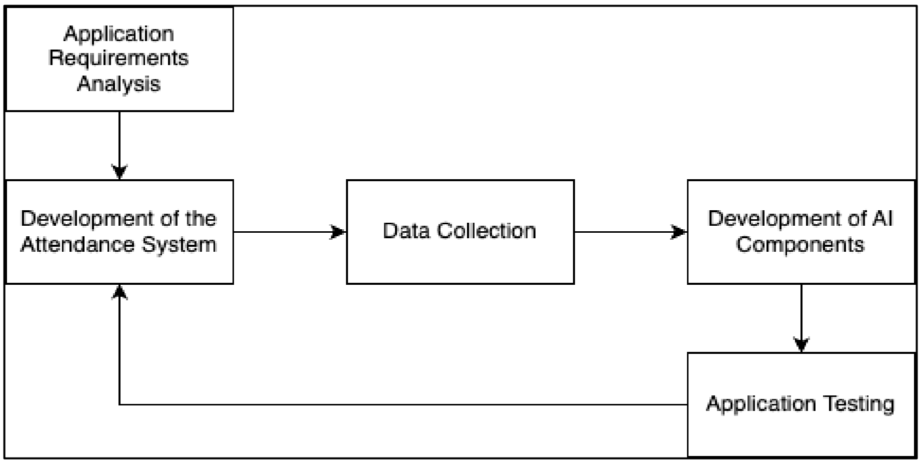

Figure 2. Research Methodology

Attendance recording is performed via a mobile app for teachers and students, running on devices with at least Android 5.0 Lollipop (API 21), 2 GB RAM, basic GPUs, 8 GB storage, and 8 MP cameras. Figure 3 presents the overall system architecture and workflow. Teacher application supplies classroom photos to the server where training and testing are conducted. Initial classroom photos are given to supply algorithms with training data. This data is used to create the machine learning model that is then used by the student on the student side of the application.

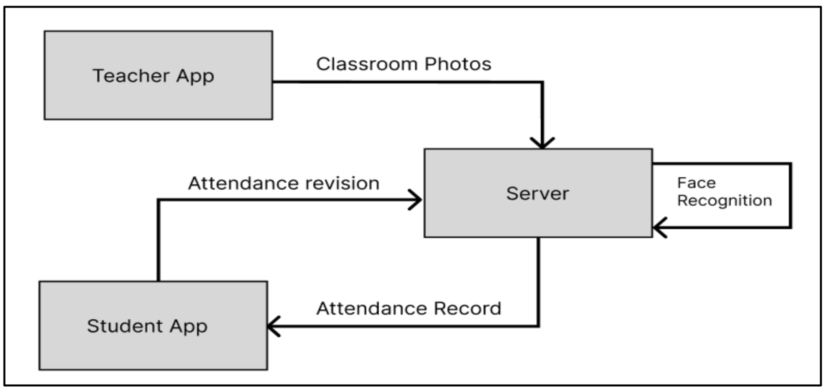

Figure 3. Attendance System Architecture

The teacher would take pictures of the classroom, which includes the whole classroom. That picture is then uploaded to the server to undergo the face recognition process. Results of the process are then shown to the students to double-check their attendance. This is done to minimize the effect of errors in automatic face recognition. The feedback can also be used for further learning algorithms. Figure *4* shows the attendance recording application for both teacher and student. This interface shows the interface in which data are collected. As shown in the image, the teacher side app requires the teacher to select a teaching schedule and upload several images of their classroom. The student side app shows the result after the teacher uploaded the image. The name shown in the app is the result of face recognition software, and the student can then approve the recognition or deny it.

(a) (b)

Figure 4. Attendance Recording (a) Teacher Side (b) Student Side

Data collection is conducted over the course of 4 weeks across 11 classrooms. Each class has a total of at least 14 sessions and contains 15 to 40 students (a total of 110 students) with different room sizes. Images for attendance are taken using mobile phones that generate images of varying resolutions; however, since a feature reduction algorithm is employed, this difference in resolution does not affect the outcome of the algorithm. Most face recognition algorithms encounter problems with lighting and angle. Most of the data in this research is well-lit, with faces facing forward at a slight angle. This focuses the research on other factors of face recognition, such as the distance of the camera from the subject. A total of 623 face images were collected, and this data is divided into several subsets for the learning algorithm to process. Seventy-five percent of the data is used for training, and 25% is used for testing. Figure 5 shows a sample image taken by the teacher and processed by the face detection algorithm; the other image shows the result of cropping performed on the initial input. It is important to note that the classroom images are all well-lit and facing the camera. This condition allows the research to omit the challenges of angle and lighting in face recognition. However, some obstructions may still occur within the data.

Figure 5. Sample Image Data and Cropped Image

Before the images were processed, they went through a preprocessing phase. The images were cropped using face detection. The resulting images are areas of the face for each student in the image. Aside from the cropped images, JSON files are also created for each face detected. These JSON files are used in the next phase to aid the face recognition algorithm. Figure 6 shows an example of the JSON file created from an image. Within this JSON file, information on the location of faces is stored along with the name and id of the student.

These JSON files enable the algorithm to assign name and student id when the face recognition phase is over.


```json
"data": [
    {
        "color": "#00ff00",
        "top": 442,
        "bottom": 478,
        "left": 77,
        "right": 113,
        "NRP": "",
        "Name": "",
        "Id_Photo": "40",
        "path": "../storage/app/public/photo/46/16/output/KooIu02voCwNvTAsj
    },
```


Figure 6. Sample JSON

The application was tested in 11 classrooms with a total of 110 unique students. These students comprise 20 female students and 90 male students, all between 17 and 23 years old. Each student has previously agreed to have their photo taken for this research via a consent form that was distributed physically. Each class session will have an average of 2 to 3 images taken. Images of the classroom sessions are captured from the teacher's mobile app without any standardization. A total of four different devices were used during data acquisition. In most images, the students are facing the camera with slight variations in head position, obstruction, and lighting. As a result, the number of images per student ranged from 2 to 12. Figure 7 shows the distribution of images collected per student. The data is unbalanced due to conditions during real-life data collection. Some students have only a small number of images because those students did not attend many of their earlier classes. The earlier classes are those in which the initial data collection was conducted. Thus, some students have an inadequate amount of training images. This, along with some other conditions (such as the teacher neglecting to collect data in class), is a common occurrence in many educational settings that interrupts the training data. It is also evident that half of the students have fewer than 5 images (mean = 5.16, median = 5).

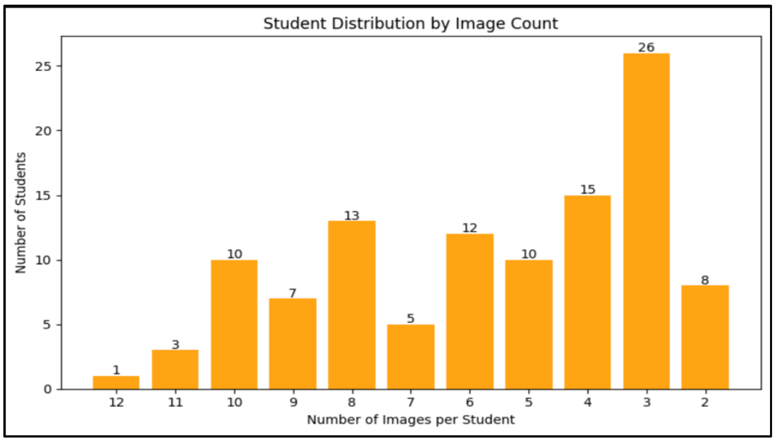

Figure 7. Distribution of Students by Number of Images Collected Per Student

Figure 8 illustrates the detailed stages of developing AI components. During the preprocessing stage, all face images are resized to have the same dimensions of 128×128 pixels. This ensures uniformity across the dataset and produces a consistent array size for subsequent processing. Feature extraction is then performed on the image to convert it into a one-dimensional array of size 49152. Flattening process is necessary to match the one-dimensional input format of the classification methods. The processed dataset is then partitioned into the training and testing subset.

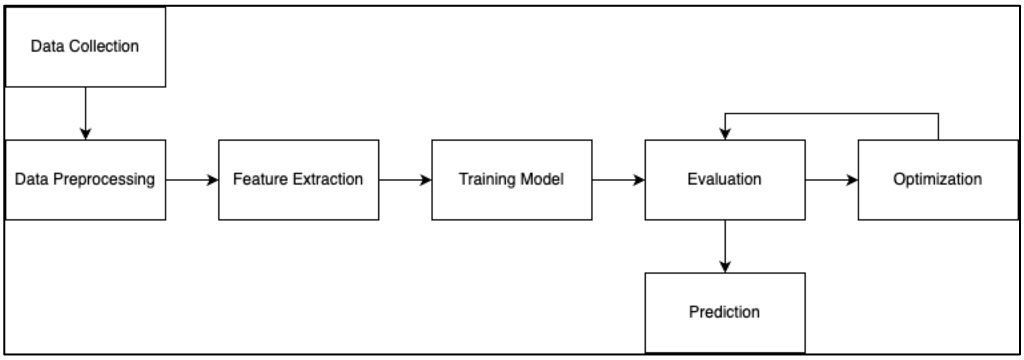

Figure 8. Development Stages of AI components

The data is split randomly; 75% for training and 25% for testing. Several methods were used for model training, including ANN, CNN, SVM, KNN, and Random Forest. Each model was trained without additional parameter adjustments. The training results were then evaluated to identify the method with the best performance. The evaluation metrics considered were precision, recall, accuracy, and the average F1-score. Once the model with the best evaluation metrics was obtained, optimization was performed through hyperparameter tuning using GridSearch to achieve improved performance.

In the recognition phase, a new face image is processed through the same preprocessing steps (resizing and flattening) to ensure compatibility with the trained model. The trained classifier then predicts the identity of the face by matching the input against the learned features from the training dataset. The recognition output is the predicted class label along with the associated confidence score.

4. Results and Discussion

Preliminary experiment results for multiple algorithms are shown in Table 1 and Table 2. To compare the algorithms, several performance metrics are used, including Recall, Precision, F1-Score, and Accuracy, with the formulas as follows:

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

TP : True Positive
FN : False Negative

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

FP = False Positive

$$\text{F1-Score} = 2\, x\, \frac{precision\ x\ recall}{precision + recall} \tag{5}$$

FP = False Positive

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}\ x\ 100\% \tag{6}$$

TN = True Negative

In the first experiment, deep learning models ANN, CNN, and SVM were built. The results can be seen in Table 1. The ANN model has very poor accuracy with an accuracy of..., and CNN also has an accuracy that is not much different from SVM. However, based on the training time results obtained, CNN requires a fairly long training time of around... so it was decided to continue the experiment with traditional algorithms.

Table 1. Experiment Result 1

	Precision	**Recall**	**Accuracy**	**F1-Score**	**Training Time**	**Confidence Level per fold**	**Standard Deviation**
SVM	0.44	0.46	0.456	0.41	1.42	[0.47, 0.47, 0.43, 0.47, 0.42]	0.021
ANN	0.0132	0.0385	0.0385	0.0147	5.16	[0.0480, 0.0400, 0.0160, 0.0081, 0.0806]	0.0257
CNN	0.3771	0.4206	0.4206	0.3774	24.75	[0.4800, 0.3680, 0.4000, 0.4597, 0.3952]	0.0422

SVM is shown to have the highest precision, recall, accuracy, and f1-score values at 0.456 accuracy. Although SVM are shown to have better accuracy, 0.456 accuracy is too small compared to other similar research. From the confidence interval and standard deviation analysis, Random Forest shows the highest stability (std = 0.007) followed by KNN (std = 0.014) and SVM (std = 0.021). This indicates that while SVM produces the best average accuracy, Random Forest offers more consistent performance across different data folds.

Table 2. Experiment Result 2

	Precision	**Recall**	**Accuracy**	**F1-Score**	**Training Time**	**Confidence Level per fold**	**Standard Deviation**
SVM	0.44	0.46	0.456	0.41	1.42	[0.47, 0.47, 0.43, 0.47, 0.42]	0.021
KKN	0.39	0.42	0.424	0.37	0.35	[0.41, 0.44, 0.41, 0.44, 0.42]	0.014
Random Forest	0.38	0.41	0.408	0.36	26.68	[0.4, 0.4, 0.39, 0.38, 0.38]	0.007

To increase accuracy, hyperparameter tuning was performed for the SVM model. Four parameters were used for tuning, 'C', 'kernel', 'gamma', and 'degree'. The details of the parameter grid are as follows:

- C: [0.1, 1, 10], which controls the regularization strength of the model.
- kernel: ['linear', 'rbf', 'poly', 'sigmoid'], determining the kernel function used for mapping the data into higher dimensions.
- gamma: ['scale', 'auto', 0.01, 0.001], influencing the impact of individual training samples.
- degree: [2, 3, 4], applicable only when the 'kernel' is set to 'poly', controlling the degree of the polynomial kernel function.

After obtaining the results from the hyperparameter-optimized model, further testing was conducted using PCA. The hyperparameter tuning process resulted in the following best configuration: C set to 10, degree set to 3, gamma set to 'scale', and the RBF kernel. In the PCA experiment, the number of components was tested from 1 to 200. The number of components that produced the highest accuracy was 23. With this number of components, the explained variance reached 95%.

The evaluation results for the model with the best parameters and the model with PCA are presented in Table 3. The results showed an increase in accuracy when using the model with the best parameters (SVM BP), as well as when using the best-parameter model trained with PCA-processed data (SVM BP + PCA).

Table 3. Experiment Result Model with Best Parameter

	Precision	**Recall**	**Accuracy**	**F1-Score**	**Training Time**	**Confidence level per fold**	**Standard Deviation**
SVM BP	0.47	0.47	0.472	0.45	0.32	[0.49, 0.5, 0.42, 0.45, 0.45]	0.028
SVM BP + PCA	0.49	0.54	0.544	0.49	0.44	[0.55, 0.44, 0.47, 0.52, 0.53]	0.039

There is a large discrepancy between this result and other similar research regarding precision and the F1-Score. This discrepancy might be attributed to the lack of training data and large number of recognition targets. To demonstrate this, another experiment is conducted by splitting the data into 2 groups, data with more than 5 images per person and data with less than 5 images (mean of the data is 5). It is apparent from the result that a larger amount of data leads to higher precision. Table 4. shows the accuracy when the data is split between less than 5 and above 5.

Table 4. Experiment Result Model with Split Image At 5

	Precision	**Recall**	**accuracy**	**F1-Score**	**Training Time**
SVM BP + PCA with Up6	0.61	0.55	0.61	0.53	0.40
SVM BP + PCA with Under5	0.33	0.43	0.48	0.36	0.22

A Mann Whitney U test is conducted on the recognition data to show if there is a statistical significance between data from groups with under 5 images and above. Table 5. shows a statistically significant difference in precision between the two groups (p-value < 0.05 for precision). The results show that the accuracy of the face recognition algorithm is directly impacted by the amount of data available.

Table 5. Mann-whitney u Test Result

Metric	**U Statistic**	**p-value**
Precision	1881.5	0.017
Recall	1608.0	0.510
F1 Score	1683.0	0.267

Experiments show the precision of SVM with BP and PCA reaches 0.49. This indicates that the algorithm can only identify faces half of the time. This contrasts with some other research that uses face recognition with SVM, which shows a higher percentage of above 80% [13]. Further examination of the research methodology reveals that most of this research uses small recognition sets and 10 or more images per person. The experiment demonstrates that the number of data points causes differences between algorithms. Therefore, prior algorithms tested in a controlled environment do not reflect realistic application scenarios.

This research attempts to collect data in realistic application scenarios. This approach resulted in non-balanced training data that reduces algorithm accuracy. Sufficient training data is mandatory for any learning algorithm. However, there are cases where training data cannot be acquired due to time or budget constraints. In educational settings for attendance recording, applications should recognize students' faces as early and accurately as possible. To better understand the relationship between the number of training images and recognition accuracy, this study has also compared face recognition performance under different training data conditions. The algorithm is tested with fewer than 8, fewer than 5, more than 8, and 5 images per student. This comparison is conducted to illustrate the performance differences in face recognition accuracy. The results of the experiment showed significant differences between fewer images and more images. The increase in performance shows the effect of sufficient training data in face recognition algorithms. These findings reinforce the importance of training data collection strategies in educational settings to ensure sufficient training data is provided. This research also did not employ any data augmentation to preserve baseline performance and data authenticity, making the comparison to other research without augmentation fairer.

Other qualities of the data in attendance recording might also affect the training result. Some images taken from the classroom are blurred. This blurriness might be caused by the student moving while the photo is taken, the distance of the student from the teacher, and the quality of the camera used to take attendance. Figure 9 shows an example of blurred training data.

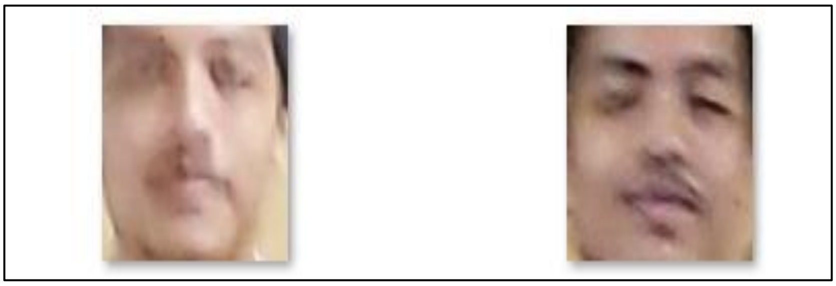

Figure 9. Blurred Training Data

Expression and obstruction potentially affect training results. In attendance recording, students might choose to show different facial expressions during the process. Glasses and other types of facial obstructions might also occur during attendance recording, which might affect training results. Figures 10 and 11 shows training data with obstructions and expressions. In both figures, glasses obstruct the face and might affect the accuracy of the algorithm. However, these cases are very scarce in the training data that they do not significantly lower the accuracy.

Figure 10. Training Data with Obstructions

Figure 11. Training Data with Expressions

A case of accidental data multiplication also occurs in this research. This is caused by the teacher inputting the same attendance record twice or more into the database. This resulted in the duplication of faces that might affect face recognition algorithms. Although there are some cases where the data quality is low, the number of data points seems to be the deciding factor for the algorithm's performance.

5. Conclusion

A mobile application with face recognition has been developed for attendance recording. Users need to upload a photo containing a group of students in the class, and the data will then be processed through face detection, followed by face recognition. Data collection was carried out over a period of four weeks using the developed application. A total of 110 students from 11 different classes were involved. This data collection was conducted in real-life conditions with no tampering or modification. For face detection, the "face recognition" library from PyPI was used. The resulting image is then used for learning and recognition. During the face recognition phase, experiments were conducted using several algorithms, including SVM, KNN, and Random Forest. Initially, each model was tested using a base version of each method, and the model using the SVM method achieved the highest accuracy at 45%. Since the result of the algorithm is significantly smaller than similar research, further analysis was conducted to analyze the cause of the dissimilarity. After examination, some factors that affect training data were discovered. One of the most obvious differences is caused by the amount and quality of training data. The amount of data per student significantly affects the algorithm's performance. Unbalanced data per student biases the model and results in low accuracy. Many similar studies that achieve higher accuracy use more images per student, and the number of students the algorithm recognizes is small. By increasing the image threshold to above 5, the face recognition algorithm's accuracy increased significantly. By increasing the threshold to 8, the face recognition algorithm's accuracy increased further. This experiment shows that face recognition algorithms using machine learning face a problem in the implementation of digital attendance in real-life scenarios. However, an optimal number of faces for learning in this scenario has not been acquired; this is due to the scope of this research and available data. This problem can be addressed in future work. Furthermore, within the limitations of this research, it is shown that SVM with PCA has better potential for face recognition compared to KNN and Random Forest. Additionally, SVM BP + PCA with BI reached the best accuracy within the given data. Since the focus of this experiment is on the problem of limited data, other factors that usually affect face recognition algorithms are ignored. This does not change the result of the experiment, as the experiment was conducted in well-lit and uniform lighting, with the students facing toward the camera. Obstructions were also minimal, since the teacher chose to take the clearest picture of the classroom. This research also opted not to use any method that alters the amount of data, such as augmentation, to show the raw performance of the algorithms in this situation. Such methods and other approaches to increase the performance of algorithms in limited data would be addressed in future work. The contribution of this research provides new insight into the robustness of existing face recognition algorithms and offers a basis for future research on face recognition in a limited data environment.

Acknowledgement

The authors wish to thank LPPM Maranatha Christian University for providing the resources necessary to conduct this research. The authors also wish to thank all the students involved in the research.

References

[1] Dev, Samridhi, and Tushar Patnaik. "Student attendance system using face recognition." In 2020 international conference on smart electronics and communication (ICOSEC), IEEE, 2020, 90-96.

[2] Bhattacharya, Shubhobrata, Gowtham Sandeep Nainala, Prosenjit Das, and Aurobinda 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), IEEE, 2018, 358-360.

[3] Budi, Setia, Oscar Karnalim, Erico D. Handoyo, Sulaeman Santoso, Hapnes Toba, Huyen Nguyen, and Vishv Malhotra. "IBAtS-Image based attendance system: A low cost solution to record student attendance in a classroom." In 2018 IEEE International Symposium on Multimedia (ISM), IEEE, 2018, 259-266.

[4] Bah, Serign Modou, and Fang Ming. "An improved face recognition algorithm and its application in attendance management system." Array 5 (2020): 100014.

[5] Rahni, AA Abd, N. Zainal, M. Z. Adna, N. E. Othman, and M. F. Bukhori. "Development of the online student attendance monitoring system (SAMSTM) based on QR-codes and mobile devices." J. Eng. Sci. Technol 10 (2015): 28-40.

[6] Rjeib, Hasanein D., Nabeel Salih Ali, Ali Al Farawn, Basheer Al-Sadawi, and Haider Alsharqi. "Attendance and information system using RFID and web-based application for academic sector." International Journal of Advanced Computer Science and Applications 9, no. 1 (2018).

[7] Mohandes, Mohamed A. "Class attendance management system using NFC mobile devices." Intelligent Automation & Soft Computing 23, no. 2 (2017): 251-259.

[8] Lukas, Samuel, Aditya Rama Mitra, Ririn Ikana Desanti, and Dion Krisnadi. "Student attendance system in classroom using face recognition technique." In 2016 International Conference on Information and Communication Technology Convergence (ICTC), IEEE, 2016, 1032-1035.

[9] Jha, Phul Babu, Arjun Basnet, Biraj Pokhrel, Bishnu Pokhrel, Gopal Kumar Thakur, and Surya Chhetri. "An automated attendance system using facial detection and recognition technology." Apex Journal of Business and Management 1, no. 1 (2023): 103-120.

[10] Nguyen-Tat, Bao-Thien, Minh-Quoc Bui, and Vuong M. Ngo. "Automating attendance management in human resources: A design science approach using computer vision and facial recognition." International Journal of Information Management Data Insights 4, no. 2 (2024): 100253.

[11] Bairagi, Rupak, Remon Ahmed, Sadia Afrin Tisha, Md Sumon Sarder, Md Sabiqul Islam, and Md Ashiqul Islam. "A real-time face recognition smart attendance system with haar cascade classifiers." In 2021 third international conference on inventive research in computing applications (ICIRCA), IEEE, 2021, 1417-1425.

[12] Budiman, Andre, Ricky Aryatama Yaputera, Said Achmad, and Aditya Kurniawan. "Student attendance with face recognition (LBPH or CNN): Systematic literature review." Procedia Computer Science 216 (2023): 31-38.

[13] Shieh, Ming-Yuan, Juing-Shian Chiou, Yu-Chia Hu, and Kuo-Yang Wang. "Applications of PCA and SVM-PSO Based Real-Time Face Recognition System." Mathematical Problems in Engineering 2014, no. 1 (2014): 530251.

[14] Gheni, Eman A., and Zahraa M. Algelal. "Human face recognition methods based on principle component analysis (PCA), wavelet and support vector machine (SVM): a comparative study." Indones. J. Electr. Eng. Comput. Sci 20, no. 2 (2020): 991-999.

[15] S. K. Rahul Jain*, "High Accuracy Face Reorganization By Pca - Svd," Nov. 2016, doi: 10.5281/ZENODO.164897.

[16] Wagh, Priyanka, Roshani Thakare, Jagruti Chaudhari, and Shweta Patil. "Attendance system based on face recognition using eigen face and PCA algorithms." In 2015 International Conference on Green Computing and Internet of Things (ICGCIoT), IEEE, 2015, 303-308.

[17] [O'Mahony, Niall, Sean Campbell, Anderson Carvalho, Suman Harapanahalli, Gustavo Velasco Hernandez, Lenka Krpalkova, Daniel Riordan, and Joseph Walsh. "Deep learning vs. traditional computer vision." In Science and information conference, Cham: Springer International Publishing, 2019, 128-144.

[18] Farell, Muhammad. "Implementation of mask use detection with svm and haar cascade in opencv." Jurnal Nasional Teknik Elektro dan Teknologi Informasi 13, no. 1 (2024): 31-37.

[19] Phillips, P. "Support vector machines applied to face recognition." Advances in neural information processing systems 11 (1998).

[20] Bong, Chin Wei, Pung Yu Xian, and Joshua Thomas. "Face recognition and detection using haars features with template matching algorithm." In International Conference on Intelligent Computing & Optimization, pp. 457-468. Cham: Springer International Publishing, 2019.

[21] Li, Haoxiang, Zhe Lin, Xiaohui Shen, Jonathan Brandt, and Gang Hua. "A convolutional neural network cascade for face detection." In Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, 5325-5334.

[22] Majeed, Aseel Wadood, Shaimaa Hameed Shaker, and Ali Adel Saeid. "A Real Time Face Recognition and Tracking Framework Using Lightweight Convolutional Neural Network." In BIO Web of Conferences, vol. 97, 00029. EDP Sciences, 2024.

[23] Kamencay, Patrik, Miroslav Benco, Tomas Mizdos, and Roman Radil. "A new method for face recognition using convolutional neural network." Advances in Electrical and Electronic Engineering 15, no. 4 (2017): 663.

[24] face-recognition: Recognize faces from Python or from the command line. Python. Accessed: Apr. 30, 2025. [Online]. Available: https://github.com/ageitgey/face_recognition

[25] Rakshit, Pranati, Rajit Basu, Sayan Paul, Sonali Bhattacharyya, Jhumpa Mistri, and Ira Nath. "Face Detection using Support Vector Mechine with PCA." In 2nd International Conference on Non-Conventional Energy: Nanotechnology & Nanomaterials for Energy & Environment (ICNNEE). 2019.