The Attendance Marking System based on Eigenface Recognition using OpenCV and Python

Khem Puthea¹*, Rudy Hartanto², Risanuri Hidayat³

¹,²,³ Department of Electrical Engineering and Information Technology, Faculty of Engineering, Universitas Gadjah Mada

*Corresponding e-mail: khem.puthea@mail.ugm.ac.id

Abstract-- Attendance of students in a large classroom is hard to be handled by the traditional system, as it is time-consuming and has a high probability of error during the process of inputting data into the computer. This paper proposed automated attendance marking system using face recognition technique. The system deployed Haar cascade to find the positive and negative of the face and eigenface algorithm for face recognition by using python programming and OpenCV library. The proposed method using PCA to resolved the problems such as lightning of the images, noise from the camera, and the direction of the student faces. The attendance of the student was updated to the Excel sheet after student's face has been recognized.

1. Introduction

There is a growing interest in face recognition recently. Unlike a human who can recognize an object instantly with a high accuracy, the automated recognition system requires a longer time to recognize an object, yet with a potential to identify the object with 100% accuracy that is extremely difficult to be accomplished by a human. Many applications have implemented automatic face recognition recently [1]. Facebook is one of the giant companies that has been implemented auto-tagging [2] by using the face recognition techniques since 2008. Another top company that employs automatic face recognition technology is Apple, which recently released the new iPhone 8 in September 2017 and it will present the iPhone X in November 2017. Both smartphones are equipped with FaceID instead of TouchID. Face recognition has gained many interests nowadays, and it is expected to replace some biometric applications.

Face recognition is widely implemented in education system such as the attendance management system[3][4][5][6] . There are a vast number of new student enrolled in modern universities, which require a system to check and verify student attendance in a large classroom in a short time. The existing manual attendance marking system is time-consuming, and it interrupts the lecture; therefore, automated attendance marking system has been developed to resolve this issue. This paper presented automated attendance management system based on face recognition. There are many challenges in the
previous face-recognition based attendance management system that disturb the system performance, such as the quality of the training, the size of the image, the light intensity of the image, and inability of the system to recognize one face but with different face expression, hairstyle change, eyeglasses, or facial changes due to aging.

An efficient face recognition system requires many complex works, including the dimensionality of the images. The principal component analysis and eigenface have been conducted in recent studies [7], [8] to address this dimensionality issue.

2. Literature Review

An easy way to comply with the paper formatting requirements is to use this document as a template and simply type your text into it. Implementing automated attendance management system will help the learning process in a school or a university. Many researchers have explored automated attendance management system using the fundamentals of face recognition. The idea is to match the acquired images with the images in the database of attendance management system to find the student and mark his attendance. Nowadays, many algorithms have been proposed in the field of face recognition system [1], [9].

In [10] proposes an automated attendance management system using face recognition techniques. This system uses two cameras installed on the classroom wall. The first camera is the sensor camera used for finding the place where the student is sitting and the second camera is used for capturing the student’s images. The way to determine the seat of a student is commonly known as active student detecting (ASD) [11]. ASD used to control the second camera to capture the right student seat. The study proposes a continuous observation and recording to improve the performance and video streaming service.

In paper [3] implement the attendance management system using computer vision algorithm on existing learning management system (LMS). The camera is fixed on the front wall to capture the entire classroom and to snap faces of all student in the class, which then verified with the existing face inside the database. The experiment was conducted in a classroom with 147 students, and the proposed system was able to detect approximately 70% of the student’s faces and provide 56% correct face identification. However, the authors mention their concern regarding the privacy of the images in the databases that were stored in a server.

In [12] the author proposes the automated attendance management system based on eigenface recognition using Matlab GUI programming. The objective of the study was to solve the time-consuming problem of a traditional attendance system. The recognition process to find the matching face uses PCA and Matlab programming. The study result shows that the system can recognize the front view of a face in the complex background and various views of a face from different directions inside a classroom.

Another author in [13] proposes a system to resolve the issue of intensity and light problem in previous studies. In other methods mentioned earlier in this review, the camera is fixed on one location to capture the images of the student in the entire classroom. The captured images need to be enhanced and then face detection process distinguish the student’s face from the background. After that, the face recognition process is performed to match each captured image with the appropriate student. The study
implements skin classification and noise removal to improve the face detection and recognition processes. Our study proposed a system using PCA and eigenface algorithms to improve the accuracy of face recognition process.

3. Proposed Methodology

3.1. Architecture

The architecture of the proposed attendance marking system is displayed in Fig. 2. The automated attendance system will perform appropriately if the database is in good quality. Figure 1 shows the process of preparing the dataset for face recognition and training the data for the next step of attendance marking system.

![Fig.1 The process of training data](image)

The first step was to get all the images of each student, then these images were resized to size, and a face vector was created. In this process, the name of each student was also obtained, and it was given ID when stored in the vector. This vector was used for processing in the next step of eigenface. The other task was training the data for eigenface and saving it in the XML file.

Steps of implementing attendance management system can be classified as:

- Camera
- Image Acquisition
- Image enhancement
- Face detection
- Face Recognition
- Mark Attendance
- Camera: The camera could be installed on the classroom wall, and we must make sure that the camera could capture the image of the entire classroom.

- Image Acquisition: The captured images from the camera were saved into the database for the next process.

- Image enhancement: This step is required to improve face recognition system. Acquired images were resized and cropped to 110 x 110 size, and then converted to grayscale to improve the result by removing the brightness. Image enhancement was applied on the grayscale images.

- Face Detection: This step distinguished the face of a student from the captured images from the camera. Face detection detected the front view of a face from the acquired images. The images from the camera were used for recognition process in the next step.

![Fig. 2 Attendance Marking System Architecture](image)
• Face Recognition: The cropped images were verified with the images inside the system database. The databases contain all student images with corresponding name and id. In face recognition step, we identified and validated the region of interest in the images captured by the camera and in the database. Eigenface method was applied in this step.

3.2. PCA Algorithm in Face Recognition

The Principal Component Analysis is conducted at the following steps:

• Suppose the training images consists of M total images and the same size \( N \times N \) or \( N^2 \) dimension. Let \( I_1, I_2, I_3, \ldots, I_M \) as asset of images \( \Phi_i \).

![Fig. 3: the training images with \( N \times N \) or \( 1 \times N^2 \) dimension](image)

• All training face images are converted to the face vectors \( \Gamma_1, \Gamma_2, \Gamma_3, \ldots, \Gamma_m \) as illustrated in Fig. 4.

![Fig. 4: Vector Image](image)
• Calculate $\varphi = \gamma - \Phi$ then create a new matrix $A = [\Phi_1, \Phi_2, \Phi_3, \ldots \Phi_m]$ (1)

with $\Phi$ is the mean of face

$$\Phi = \frac{1}{M} \sum_{i=1}^{M} \Gamma_i = \frac{\Gamma_1 + \Gamma_2 + \ldots + \Gamma_m}{M} \quad (2)$$

• Calculate the covariance matrix $C = A^T A$, and the dimensionality is reduced because the covariance has the dimension $N \times N$.

• Select the best eigenvalue $K$ which, $K < M$ represents the training data

• Calculate eigenvalue and eigenvector

• Reduce the eigenface by computing the matrix $A$ with the set of eigenvectors

• Then calculate the eigenface of images

• Calculate the distance between eigenface and the images using Euclidian distance

$$d(A, B) = \sqrt{\sum_{i=1}^{D} (a_i - b_i)^2} = \|A - B\| \quad (3)$$

• If the distance between the testing images with the training database face, it means the test images belong to the person in the training face [14], [15].

• The final step is updating the recognized face into the excel sheet.

3.3. Experiments and Result

This part discusses and shows the analysis process in the attendance marking system. The experiment of this paper was conducted using OpenCV3 and Python3 to implement eigenface recognition algorithm. The step from the starting point to the end of the implementation process is explained as follows.

A. Create Dataset

The system requires the database to verify the captured images to implement face recognition. The step to create a dataset for automated attendance marking system using face recognition is explained as follows.

• Import opencv library : import opencv library for images manipulation

  // import cv2

• Load Haar Cascades : import the haar cascade for face and eye detection in the images.

  // face = cv2.CascadeClassifier('haarcascade_frontal.xml')

  // eye = cv2.CascadeClassifier('haarcascade_eye.xml')

• Create camera object: OpenCV provides a campatability function to use camera for capturing the images in the videos.

  // video = cv2.VideoCapture(0)
• convert images to grayscale: get all the images from the camera object and then convert them to grayscale
  
  // gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

• Face and eye detection: Using Haar cascade to detect face and eyes from the images.
  
  // faces = face.detectMultiScale(gray, 1.3, 5)

  with 1.3 is the scale factor of the images and 5 is the number of minimum Neighbors.

• Dataset: Save images from the previous step to the specific folder
  
  // cv2.imwrite(`dataset/User.`+str(count)+``.jpg`, frame)

  Fifty images of the student are arranged as can be seen in:

  ![Image of the student](image.png)

  **Fig. 5: Image of the student**

• Destroy the camera object: the object of the camera is destroyed after finishing the process of capturing each student image to prevent an error when the camera captures the next object.
  
  // cv2.release()

### B. Training Data

After the data is prepared, next step is the training step:

• Import library: import opencv and python libraries
  
  // import os is for os path

  // import cv2 is for opencv library

  // from PIL import Image is for image library

• Create Eigenface Recognizer
  
  // eigenface = cv2.face.createEigenfaceRecognizer(15)

• Get the images: get all images from the dataset folder for each student and correspond them to each ID and student name

• Get the path to each images
  
  // imagePath = [os.path.join(path, f) for f in os.listdir(path)]

• Create to list to store each facelist and ID
C. Eigenface Recognition

The steps of implementing eigenface with OpenCV and Python are as follows:

- Import OpenCV library
  ```python
  import cv2
  ```
- Load Haar Cascades: import the haar cascade for face and eye detection in the images.
  ```python
  faceC = cv2.CascadeClassifier('haarcascade_frontal.xml')
  eyeC = cv2.CascadeClassifier('haarcascade_eye.xml')
  ```
- Create Eigenface recognizer with threshold 4000
  ```python
  recognize = cv2.face.createEigenfaceFaceRecognizer(15, 4000)
  ```
- Create camera object to capture images
  ```python
  camera = cv2.VideoCapture(1)
  ```
- Read camera object and convert gray images
  ```python
  image = camera.read()
  grayFace = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
  ```
- Detect face and eye, and confirm eye inside the face
  ```python
  eye = eyeC.detectMultiScale(grayFace)
  ```
• Get the label and ID from the opencv prediction function
  
  //result = cv2.face.MinDistancePredictionCollector()
  //recognize.predict(grayFace, result, 0)
  //ID = result.getLabel();
  //Conf= result.getDist();

• Then ID and confidence are passed to compare with the name and id inside the databases.

We draw the rectangle on each face in real time with name and confidence from the prediction by eigenface to show the confidence of the system. The results of the students verifying process by the system are displayed in Fig. 6, Fig. 7, Fig. 8.

Fig. 6 : Result of face recognition in frontal direction

Fig. 7 : Result of face recognition in right direction
Face recognition shows that the result in the system has more accuracy than the previous work and related work, due to each direction of the face could be detected in the system, which is proposed in the paper.

D. Update Attendance

- The final step of attendance marking system is verification of the attendance sheet. After the face recognition step, then the next task is updating the ID from the recognized face to the CSV sheet for attendance marking. Updating the attendance into the CSV file is follows the step:

|   | A      | B     | C   |
|---|--------|-------|-----|
| 1 | Name   | ID    | status |
| 2 | Puthea | 406229| Yes  |
| 3 | Nugroho| 406228| No   |
| 4 |        |       |      |
| 5 |        |       |      |
| 6 |        |       |      |
| 7 |        |       |      |
| 8 |        |       |      |
| 9 |        |       |      |
|10 |        |       |      |

- Check the confidence again to the threshold
- Get the ID and label which belong to the student
- Check whether the ID and Label already exist
- Save the ID and label to csv
- The status filed will be updated to “Yes” if the system could recognize the face of a student.

**Table 1**: Summary OpenCV function using in Attendance Marking System

| Task                                      | Expect result                                      | Result                      | Status |
|-------------------------------------------|----------------------------------------------------|-----------------------------|--------|
| Cv2.VideoCapture(0)                      | Connected with camera                              | Camera connected            | Success|
| Load Hair Cascade                         | Load frontal face and eye cascade file              | Ready to extraction frontal face and eye | Success|
| Extract frontal face and eyes             | Extracted face and eye with eigenface               | Face and eye extracted      | Success|
| Recognize.predict()                       | Using PCA and Eigenface to train data              | Training data and save to XML file | Success|
| Recognition                               | Verify acquired images with database images        | Verify and get the result for updating to attendance sheet | Success|

In Fig. 9 is a graph, which shows the accuracy of face ID against to the number of 200 component. In the graph show that the face is good against to 200 component due to the line straight from 0 to 200. However, the previous work did not show the straight line look like Fig. 9. The blue line shows straight which means the face of the student is correct compared to 200 component.

![ID against Number of Components](image1)

**Fig 10**: ID against number of components

![Confidence against Number of Components](image2)

**Fig 11**: ID against number of components
In Figure 10 is a graph shows the confidence against with the number of component in eigenface. The graph shows that the confidence of images against with 50 component still less accuracy, however, the accuracy is increased between 50 to 200 component.

Table 2: Summary OpenCV function using in Attendance Marking System

| Method         | Result |
|----------------|--------|
|                | Left direction | Frontal | Right direction |
| Related work   | 53%    | 78%    | 63%    |
| This paper     | 65%    | 86%    | 69%    |

In Table 2 shows the accuracy and efficiency of the images in different views on the related work and the implementation of this paper.

4. Conclusion

The automated attendance marking system can prevent fake attendance and improve the time-consuming issue. There are many automated attendance management systems implemented with biometric to manage students attendance in a large classroom. However, the face recognition system has better performance than the biometric system, including in the aspects of processing time. The proposed method in this study improved the brightness of the face images and the direction of the eyes and nose.

Attendance Marking System has been tested and the computational time is 1:12mn to recognizing the image against with 200 component of eigenface. Comparing attendance marking with the related work in section II shows that proposed system is very high accuracy and less computational time.

On the other hand, the proposed system can be worked with video-streaming service, which provides a profound application on attendance management system. However, the streaming service also is given a chance to make a continuous recognition, which provides the lowest error on the attendance sheet. However, the proposed system still need the improvement on accuracy. In future work, Artificial Neural network will be conducted to achieve a higher accuracy.

References

[1] M. Sajid, R. Hussain, and M. Usman, 2014, “A conceptual model for automated attendance marking system using facial recognition,” in Ninth International Conference on Digital Information Management (ICDIM 2014), pp. 7–10.
[2] Z. Stone, T. Zickler, and T. Darrell, 2008, “Autotagging Facebook: Social network context improves photo annotation,” in 2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, pp. 1–8.

[3] V. Shehu and A. Dika, 2010, “Using real time computer vision algorithms in automatic attendance management systems,” Inf. Technol. Interfaces (ITI), 2010 32nd Int. Conf., pp. 397–402.

[4] A. S. K. Bijlani, S. Suresh, and P. P, 2016, “Attendance Monitoring in Classroom Using Smartphone and Wi-Fi Fingerprinting,” in 2016 IEEE Eighth International Conference on Technology for Education (T4E), pp. 62–67.

[5] N. K. Jayant and S. Borra, 2016 “Attendance management system using hybrid face recognition techniques,” in 2016 Conference on Advances in Signal Processing (CASP), vol. 2, no. 11, pp. 412–417.

[6] J. Joseph and K. P. Zacharia, 2013, “Automatic Attendance Management System Using Face Recognition,” Int. J. Sci. Res., vol. 2, no. 11, pp. 327–330.

[7] M. Turk and A. Pentland, 1991, “Eigenfaces for Recognition,” J. Cogn. Neurosci.

[8] Neerja and E. Walia, 2008, “Face Recognition Using Improved Fast PCA Algorithm,” in 2008 Congress on Image and Signal Processing, vol. 1, pp. 554–558.

[9] W. Zhao, R. Chellappa, and P. Phillips, 2003, “Face recognition: A literature survey,” ACM Comput. Surv.

[10] Y. Kawaguchi, 2005, “Face Recognition-based Lecture Attendance System,” 3rd AEARU.

[11] S. Nishiguchi, K. Higashi, Y. Kameda, and M. Minoh, 2003, “A sensor-fusion method for detecting a speaking student,” in Proceedings - IEEE International Conference on Multimedia and Expo, vol. 1, pp. I129–I132.

[12] P. Mehta, 2016. “An Efficient Attendance Management System based on Face Recognition using Matlab and Raspberry Pi 2,” Int. J. Eng. Technol. Sci. Res. IJETSR, vol. 3, no. 5, pp. 71–78.

[13] P. Wagh, R. Thakare, J. Chaudhari, and S. Patil, 2015. “Attendance system based on face recognition using eigen face and PCA algorithms,” in 2015 International Conference on Green Computing and Internet of Things (ICGCIoT), pp. 303–308.

[14] V. Perlibakas, 2004, “Distance measures for PCA-based face recognition,” Pattern Recognit. Lett., vol. 25, no. 6, pp. 711–724.

[15] L. Zhao and X. Xu, 2011, “Distance function for face recognition based on 2D PCA,” 2011 Int. Conf. Multimedia Technol., pp. 5814–5817, 2011.