http://ejournal.seaninstitute.or.id/index.php/InfoSains
Jurnal Info Sains : Informatika dan Sains, Volume 11, No 01, 2021
E-ISSN.2797-7889, P-ISSN.2089-3329

Facial Recognition Using The Haar Cascade Classifier Method For Smart Absence

Reza Alamsyah[1], Indra Sidabutar[2]
[1]STMIK Methodist, Informatics Engineering, Binjai, Indonesia, [2*]STMIK Methodist, Information Systems, Binjai, Indonesia
Email: indra3753@gmail.com

Keywords

Smart Attendance, Facial Recognition, Haar Cascade

Abstract. The Processing student attendance data at STMIK Methodist Binjai plays an important role in implementing teaching and learning activities. The STMIK Methodist Binjai attendance system still uses stationery which is less efficient, thus affecting learning productivity. Therefore, a solution is needed to help with attendance problems so that attendance can be run efficiently and with fast computing. Namely by using Face Recognition technology for Smart Attendance using the haar cascade method in OpenCV. The results of the developed application can recognize faces with an accuracy rate of 81%. And users also manage data in the system and recording attendance data is stored in excel files

1. INTRODUCTION

Processing student attendance data at STMIK Methodist Binjai plays an important role in implementing teaching and learning activities. The STMIK Methodist Binjai attendance system still uses stationery which is less efficient, thus affecting learning productivity. In determining attendance, the time required to call each student can disrupt the concentration of the learning process [1], [2].

Recording attendance data in the era of globalization can be done digitally. As a result of the data recording system, digital images can have optical form such as photos, analog form such as signals on video tape as seen on a TV screen, or in digital form that can be directly stored on storage media [3]. The human face plays an important role as a distinguishing factor for personal identification or as an individual identity, based on uniqueness and special characteristics. One of the challenges related to faces is facial recognition [4], [8].

At STMIK Methodist, there are several obstacles in the attendance system, which include problems such as being able to manipulate attendance data, losing attendance book records, and difficulties in summarizing attendance data. Therefore, a solution is needed to help with the problem of attendance, one of the attendance systems is to use Smart Attendance via facial recognition [5], [7]

A system capable of counting the number of faces detected in images taken from a webcam [9]. The method used is called Haar Cascade. This method uses Haar-like features, and to apply this method, initial training is required to create a decision tree called a cascade classifier. This cascade classifier functions to determine whether an object is present or not in each frame processed. Haar features are defined by comparing the mean of pixels in dark areas with the mean of pixels in light areas [6], [10].

The Haar style feature calculation process involves subtracting the black box pixel values from the white box pixel values. Before subtraction is carried out, first the black box pixel values in each section have been divided into three numbers, and then these numbers are combined with the white box pixel values. After all these calculations have been carried out, the value of the black box and white box is then subtracted [8]. The problem raised in this research is, designing a facial recognition system using the Haar Cascade Classifier algorithm for smart student attendance so that the attendance process runs efficiently and has speed and accuracy in recognizing faces so that it can help the campus to record and monitor attendance data in real time and minimize acts of cheating during absenteeism [12].

2. METHOD

The method used is called Haar Cascade. This method uses Haar-like features, and to apply this method, initial training is required to create a decision tree called a cascade classifier. This cascade classifier functions to determine whether an object is present or not in each frame processed. Haar features are defined by comparing the mean of pixels in dark areas with the mean of pixels in light areas[11]

The first step taken by the Haar method to identify facial characteristics in an image is to convert the image into a grayscale image. [13],[14],[15]

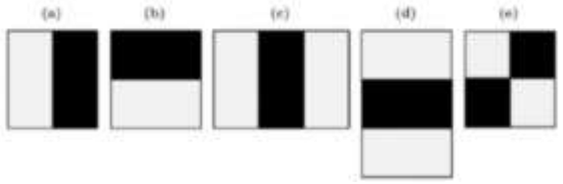


Figure 1. Harlike Feature

The selection of eye, nose and mouth features can be seen in Figure 2.

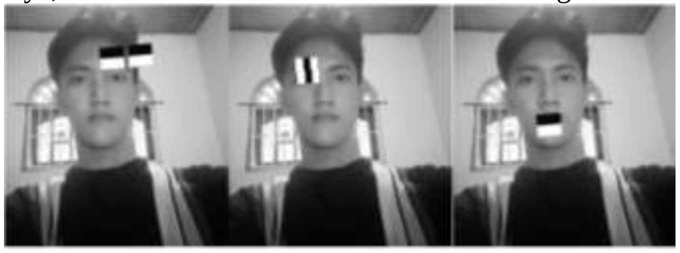


Figure 2. Eye, Nose and Mouth Features

The Haar style feature calculation process involves subtracting the black box pixel values from the white box pixel values. Before subtraction is carried out, first the black box pixel values in each section have been divided into three numbers, and then these numbers are combined with the white box pixel values. After all these calculations have been carried out, the value of the black box and white box is then subtracted.

The facial data that has been taken from 80 facial samples from 5 people will be stored in the facial data folder. The following is a facial data table.

Table 1. Facial Data

No	Student Name	Training Data	Student Code
1	Aldi Pradana		2019101008_Ti
2	Zahrul Kamal		201910100_Ti
3	Sinta Safiro		2019101004_Ti
4	André		201910120_Ti
5	Boy		202120203_Ti

Meanwhile, the integral image calculation is by using the following formula:

$$ii(x,y) = \sum_{x^{i} \leq yi \leq yi} (xi\,, yi)$$

ii (x, y) = Integral image at location x, y
i (x',y') = value of a pixel in the original image

3. RESULTS AND DISCUSSION

Results

The following are the results of the discussion applied by the author as follows.

Table 2. Final results of facial recognition testing

NO	Face Description	Result Status
1	Face in a forward facing position	Succeed
2	More than one person's face	Succeed
3	Face wearing glasses	Fail
4	Face wearing a mask	Fail
5	Face by looking up, right, left	Fail
6	Face wearing a hat	Succeed
7	Face with one eye closed	Succeed
8	The face has a mustache at the mouth	Succeed
9	Face with both eyes closed	Succeed
10	Face with smiling and angry expressions	Succeed
11	Face in dark areas	Succeed

From the table that has been presented, we can carry out calculations to determine the level of accuracy that has been tested in direct face detection and recognition. The level of accuracy in face detection is calculated using the following calculation method:

Result = (Number of faces successfully detected)/(Total detection tests) x 100%
Results = 9/11 x 100%
Yield = 81%

From the calculations carried out above, the results show that the level of accuracy of facial recognition based on the tests carried out reaches 81%.

Evaluation

Designing the user interface for a student attendance system, visual elements such as page layout, buttons, and others must be considered carefully. The organization of interface elements is important to ensure a good user experience. Interface design must pay attention to usability principles such as text readability, hierarchical organization of information, and the use of negative space so that users can easily interact with the attendance system.

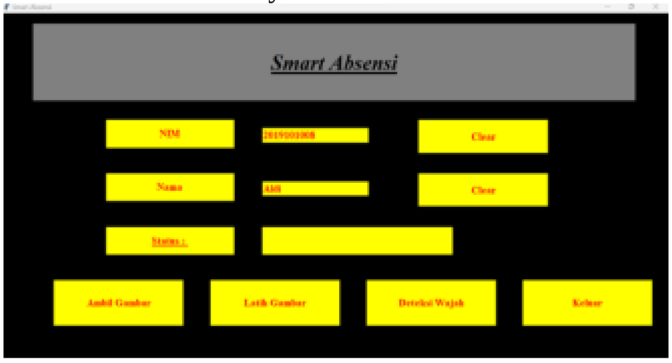
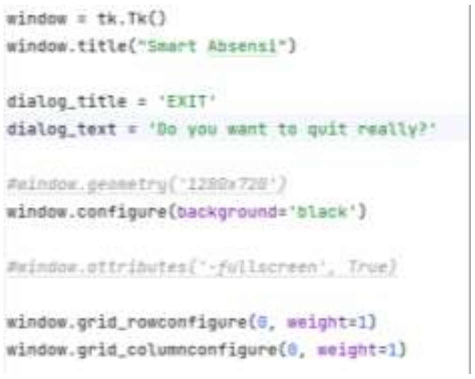
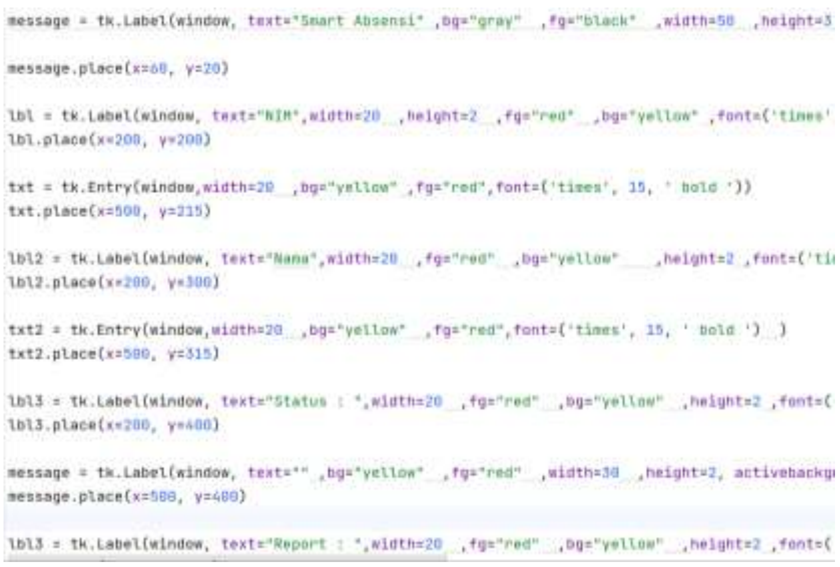


Figure 1. Smart Attendance Application Display



```
window = tk.Tk()
window.title("Smart Absensi")

dialog_title = 'EXIT'
dialog_text = 'Do you want to quit really?'

#window.geometry('1280x720')
window.configure(background='black')

#window.attributes('-fullscreen', True)

window.grid_rowconfigure(0, weight=1)
window.grid_columnconfigure(0, weight=1)
```

The code above is the initial display design process for the smart attendance application that uses Tkinter. The following is the text design for the smart attendance application



```
message = tk.Label(window, text="Smart Absensi" ,bg="gray"  ,fg="black"  ,width=50  ,height=3

message.place(x=60, y=20)

lbl = tk.Label(window, text="NIM",width=20  ,height=2  ,fg="red"  ,bg="yellow" ,font=('times'
lbl.place(x=200, y=200)

txt = tk.Entry(window,width=20  ,bg="yellow" ,fg="red",font=('times', 15, ' bold '))
txt.place(x=500, y=215)

lbl2 = tk.Label(window, text="Nama",width=20  ,fg="red"  ,bg="yellow"    ,height=2 ,font=('ti
lbl2.place(x=200, y=300)

txt2 = tk.Entry(window,width=20  ,bg="yellow"  ,fg="red",font=('times', 15, ' bold ')  )
txt2.place(x=500, y=315)

lbl3 = tk.Label(window, text="Status : ",width=20  ,fg="red"  ,bg="yellow"  ,height=2 ,font=(
lbl3.place(x=200, y=400)

message = tk.Label(window, text="" ,bg="yellow"  ,fg="red"  ,width=30  ,height=2, activebackg
message.place(x=500, y=400)

lbl3 = tk.Label(window, text="Report : ",width=20  ,fg="red"  ,bg="yellow"  ,height=2 ,font=(
```

Before taking attendance, make sure you have filled in student data so that attendance data can be seen

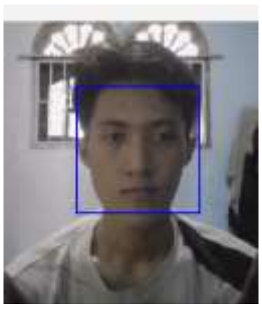


Figure 2. Face Detection Process

The image above is the process of capturing facial data and then carrying out the facial detection process so that the results of the facial data recording are used to train the system to be able to distinguish one face from another.

The first thing to do is write the import cv2 code, import numpy as np which means that here we use the OpenCV and NumPy libraries. The coding above uses source taken from the OpenCV module and uses the Python 3.1 programming language. The source in the OpenCV module is an external file that contains a series of algorithms for finding faces in images or videos. This external file is based on XML, this file is called HaarCascade. Then we enter it into the program, to import it we use the CascadeClassifier method.

However, first you have to change the image that has been captured into a black and white image (grayscale), here to make the image or video easier to process using an algorithm. To change the image to black and white, use the code gray=cv2.cvtColor(img, cv2.COLOR_BGR2GRAY). This code will change the image to black and white which is located in the gray variable.

```
cv2.rectangle(img, (x, y), (x + w, y + h), (255, 0, 0), 2)

sampleNum = sampleNum + 1

cv2.imwrite("TrainingImage\ " + name + "." + Id + '.' + str(sampleNum) +
            ".jpg", gray[y:y + h, x:x + w])

cv2.imshow( winname: 'frame', img)

if cv2.waitKey(100) & 0xFF == ord('q'):
    break

elif sampleNum > 80:
    break
    cam.release()
    cv2.destroyAllWindows()
```

The first thing to do is write the import cv2 code, import numpy as np which means that here we use the OpenCV and NumPy libraries. The coding above uses source taken from the OpenCV module and uses the Python 3.1 programming language. The source in the OpenCV module is an external file that contains a series of algorithms for finding faces in images or videos. This external file is based on XML, this file is called HaarCascade. Then we enter it into the program, to import it we use the CascadeClassifier method.

However, first you have to change the image that has been captured into a black and white image (grayscale), here to make the image or video easier to process using an algorithm. To change the image to black and white, use the code gray=cv2.cvtColor(img, cv2.COLOR_BGR2GRAY). This code will change the image to black and white which is located in the gray variable, then enter the code for face detection which was explained previously.

The next step is to apply the following code:

```
faces = detector.detectMultiScale(gray, 1.3, 5)
```

To draw a box on the detected face, a For loop is used for each face variable

```
for (x, y, w, h) in faces:
    cv2.rectangle(img, (x, y), (x + w, y + h), (255, 0, 0), 2)
sampleNum = sampleNum + 1
```

For each face variable that is taken there will be a box image in that face area. In this coding, the rectangle method is used to draw a box on the detected facial area. In the code above, 5 parameters are declared. The first parameter is the location where you will draw, namely on the image, not gray because gray is only used to detect faces, but here you draw on a color image or original image. Then the second parameter is the tuple (x,y) which is the top left point of a square. The third parameter, namely the tuple (x+w, y+h), is useful for defining the bottom right point of the square, so that a square with 2 points will be formed. The fourth parameter is a tuple for OpenCV colors using BGR colors, namely Blue, Green, Red and the opposite of RGB. The strongest color value is 255, so in the code above we use a square box outline using red. And the last parameter is number 2 which is the thickness of the square line. After the face is detected, it will be captured 80 times which will then be stored in the dataset folder.

```
cv2.imwrite("TrainingImage\ "+name +"."+Id +'.'+ str(sampleNum) +
            ".jpg", gray[y:y+h,x:x+w])
```

Then the cv2.waitkey function maintains the window so that it continues to display the image. The cam.release() function is to provide the exit code to the camera. Meanwhile, the cv2.destroyAllWindow() function is to close other windows that are currently open.

Figure 3. Student Data Storage

After going through the face detection process and the image is successfully captured and converted to grayscale, the image training process is then carried out using the code above. The image will be trained and then labeled with a name, and ID or NIM.

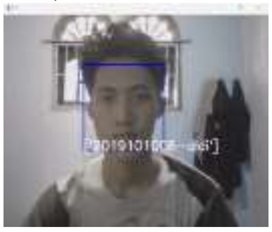
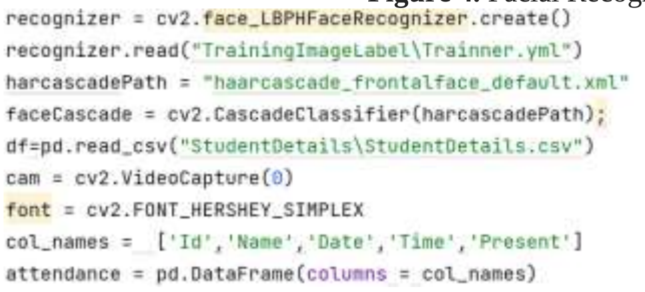


Figure 4. Facial Recognition of ID NIM



```
recognizer = cv2.face_LBPHFaceRecognizer.create()
recognizer.read("TrainingImageLabel\Trainner.yml")
harcascadePath = "haarcascade_frontalface_default.xml"
faceCascade = cv2.CascadeClassifier(harcascadePath);
df=pd.read_csv("StudentDetails\StudentDetails.csv")
cam = cv2.VideoCapture(0)
font = cv2.FONT_HERSHEY_SIMPLEX
col_names =  ['Id','Name','Date','Time','Present']
attendance = pd.DataFrame(columns = col_names)
```

By making a cascade classifier using a haar cascade to detect faces. Next is the code to make a video capture motorcycle taxi using the number 0 because this application uses an internal camera. Meanwhile, if you use the number 1, it means you are using an external camera or webcam. Next is to determine the font (size) because we will label the name of the face owner on the image so we need a font for the text. The parameters for making the font above are the font name, horizontal scale, vertical scale, spacing, line thickness, line type. Here is the code:

in this code to predict the user's passport number and the accuracy of each prediction, then the next code is to write the user's identity name which is located under the face.

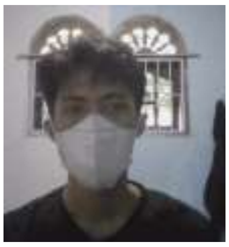
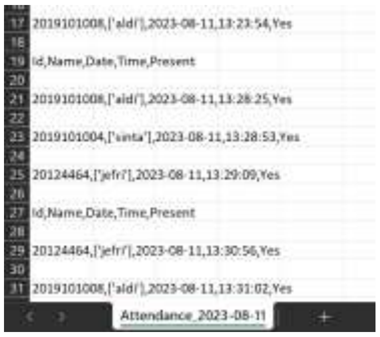


Figure 5. Unrecognized Face Using a Mask

The image above cannot be detected because it uses accessories on the face, thereby preventing the haarcascade algorithm from detecting faces



```
17 2019101008,['aldi'],2023-08-11,13:23:54,Yes
18
19 Id,Name,Date,Time,Present
20
21 2019101008,['aldi'],2023-08-11,13:28:25,Yes
22
23 2019101004,['sinta'],2023-08-11,13:28:53,Yes
24
25 20124464,['jefri'],2023-08-11,13:29:09,Yes
26
27 Id,Name,Date,Time,Present
28
29 20124464,['jefri'],2023-08-11,13:30:56,Yes
30
31 2019101008,['aldi'],2023-08-11,13:31:02,Yes
```

Figure 6. Attendance Results

4. CONCLUSIONS

The cause of failure in the face detection process is related to several factors, especially in relation to the condition of the facial image, such as orientation and clarity of the facial appearance. Meanwhile, obstacles in facial recognition through applications are mostly caused by the situation that the face is equipped with accessories such as sunglasses that cover most of the facial area, and also depend on the existing lighting level. The performance of this facial detection and recognition application is affected by the specifications of the laptop used. Using a laptop with low specifications can result in a "Not Responding" message.

REFERENCE

[1] Ningsih, AF, & Fibriany, FW (2018). Web-Based Permit and Leave Absence Information System at BPSDM KEMENHAGRI. IJCIT (Indonesian Journal on Computer and Information Technology), 3(2)

[2] T. Triyono, R. Safitri, and T. Gunawan, "Web-Based Design of Teacher and Staff Attendance Information System at SMK Pancakarya Tangerang," SENSI J., vol. 4, no. 2, pp. 153–167, 2018

[3] R. Taufiq, RR Ummah, I. Nasrullah, and AA Permana, "Design and Development of a Web-Based Employee Payroll Information System at Madrasah Ibtidaiyah Nurul Huda, Tangerang City," J.Inform. Univ. Pamulang, vol. 4, no. 4, p. 119, 2019

[4] Rhomadhona, H. (2018). Application of Web-Based QR Code Technology for Employee Attendance at BKPSDM Tanah Laut Regency. Journal of Humanities and Technology, 4(1), 1–6

[5] R. Nurmalina, JA Yani Km, T. Laut, and K. Selatan, "Planning and Development of Student Attendance Applications Using Smart Cards for Smart Campus Development (Case Study of Tanah Laut State Polytechnic)," 2017

[6] A. Priadana and M. Habibi, "Face detection using haarcascade to filter selfie face images on Instagram," Proceedings - 2019 International Conference on Artificial Intelligence and Information Technology (ICAIIT 2019), p. 6–9, 2019, doi: 10.1109/ICAIIT.2019.8834526.

[7] Hidayatulloh, P. (2017). "Digital Image Processing: Theory and Real-World Applications." Bandung: Informatics..

[8] Purwanto, Panji, et al. (2015). "Implementation of Face Identification and Face Recognition on Surveillance Cameras for Hazard Detection." Computer Systems Study Program, Faculty of Engineering, Telkom University.

[9] Adi Pamungkas, “Digital Image Processing,” 2017. https://pemrogramanmatlab.com/2017/07/26/pengolahan-citra-digital/ (accessed Jun. 12, 2020).

[10] S. Abidin, "Face Detection Use Haar Cascade Classifier Method Based on Webcam in Matlab," Journal of Electrical Technology, vol. 15, no. 1, p. 21, 2018, doi: 10.31963/elekterika.v15i1.2102.

[11] R. Wiryadinata, R. Sagita, S. Wardoyo, and Priswanto, "Face Recognition inAttendance System Using Dynamic Times, Principal Component Analysis, and Gabor Wavelet Methods,"

ISSN 1858-3075, vol. 12, no. 1, pp. 1–8, 2016.

[12] NL Fitriyani, CK Yang, and M. Syafrudin, "Real-time eye status detection system using Haar cascade classification and Circular Hough transform," IEEE Global Conference on Consumer Electronics (GCCE) 2016, p. 5–7, 2016, doi: 10.1109/GCCE.2016.7800424.

[13] Sethy, A., Raut, A.K., & Nayak, S.R. (2022). Face Recognition Based Automatic Recognition System. International Conference on Cloud Computing, Data Science & Engineering (Meeting)

[14] Kururniawati, Y. (2019). CLASS PRESENCE SYSTEM USES FACIAL RECOGNITION WITH THE HAAR CASCADE CLASSIFIER METHOD UNIVERSITY OF SEMARANG.

[15] Kakarla, S., Gangula, P., Rahul, M., Singh, C.S., & Sarma, T.H. (2020, 12 21). Intelligent Attendance Management System Based on Facial Recognition cxvi CNN. IEEE-HYDCON, 515-701.