

CHAPTER 5

IMPLEMENTATION AND RESULTS

5.1. Implementation

All modules used in this project are made using the python programming language, with the help of opencv as a library to process data in the form of videos directly. The following is an example of the source code used to run the program:

```
1 import cv2
2
3 faceCascade = cv2.CascadeClassifier("haarcascade_frontalface_default.xml")
4 eyeCascade = cv2.CascadeClassifier("haarcascade_eye.xml")
5
6 video_capture = cv2.VideoCapture(0)
7 menggunakankacamata = False
8
9 while True:
10     ret, frame = video_capture.read()
11     gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
12     wajah = faceCascade.detectMultiScale(gray, 1.1, 5)
13     roi_gray = gray
14
15     for(x, y, w, h) in wajah:
16         roi_gray = gray[y:y+h, x:x+w]
17         roi_color = frame[y:y+h, x:x+w]
18         if menggunakankacamata:
19             cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 3)
20             cv2.putText(frame, 'Glasses On', (x, y),
cv2.FONT_HERSHEY_SIMPLEX, 1.5, (0,255,0),3)
21         else:
22             cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 0, 255), 3)
23             cv2.putText(frame, 'Glasses Off', (x, y),
cv2.FONT_HERSHEY_SIMPLEX, 1.5, (0,0,255),3)
24
25     mata = eyeCascade.detectMultiScale(roi_gray, 1.18, 35,)
26     roi_gray = gray
27     for (sx, sy, sw, sh) in mata:
28         cv2.rectangle(roi_color, (sh, sy), (sx+sw, sy+sh), (255, 0,
0), 2)
29         cv2.putText(frame, 'Eyes', (x + sx, y + sy), 1, 1, (0, 255, 0),
2)
30
31     if len(mata)>0:
32         menggunakankacamata = False
33     else:
34         menggunakankacamata = True
35
36     cv2.putText(frame, 'Number Of Faces : ' + str(len(wajah)), (30,
30), cv2.FONT_HERSHEY_SIMPLEX, 1, (255,0,0),3)
```

```

37     cv2.putText(frame, 'Number Of Eyes : ' + str(len(mata)), (30,
60), cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 0, 0),3)
38     cv2.imshow('Video', frame)
39
40     if cv2.waitKey(1) & 0xFF == ord('s'):
41         break
42
43 video_capture.release()
44 cv2.destroyAllWindows()

```

In line 1, the import cv2 function is used to read datasets in the form of videos directly and haarcascade files.

Rows 3 and 4 are used to enter the haarcascade_frontalface.xml and haarcascade_eye.xml data sets which are used to detect faces and eyes of people in videos and are processed using cv2 and the cascade classifier.

Row 6 is used to open the laptop's webcam camera which is used to record a person's face and eyes. Whereas in line 7 it is used to create a variable using glasses or not with a default of false.

Lines 9 through 13 are used to loop through each frame in the video from the webcam camera, then each frame is converted to grayscale.

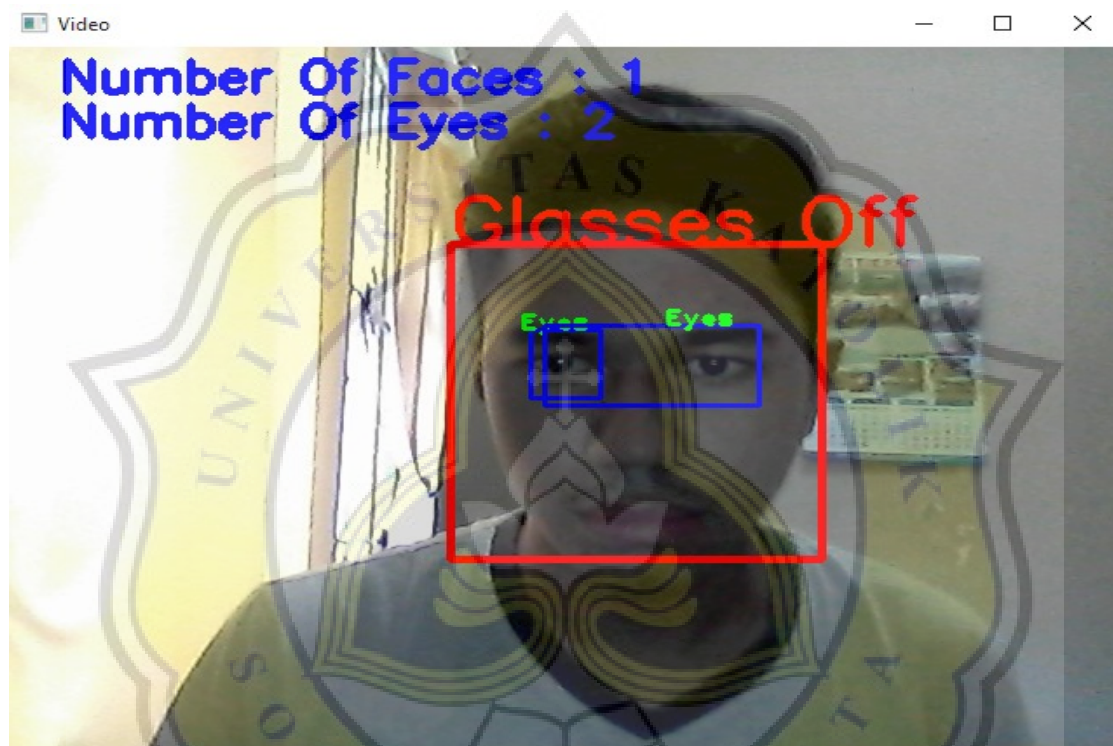
Lines 15 through 23 are used to detect faces by looping through each frame, with the faceCascade variable containing the haarcascade_frontalface.xml dataset processed using cv2 and the cascade classifier. If the wearing glasses variable is set to true, a text will appear at the top of the face i.e. Glasses is On and a green box will appear. Meanwhile, if the variable using glasses becomes the default false, a text will appear at the top of the face, namely Glasses Off and a red box will appear.

Rows 25 through 34 are used to detect eyes by looping through each frame, with the eyeCascade variable containing the haarcascade_eye.xml dataset processed using cv2 and the cascade classifier. If the eye is detected, the variable wearing glasses will default to false which means the person is not wearing glasses and a blue box appears around the eye area, whereas if the eye is not detected, the variable wearing glasses will default to true which means the person is wearing glasses.

Rows 36 through 38 are used to count the number of people's faces and eyes in live video. While in lines 40 and 41 are used to turn off the laptop webcam by pressing the s key.

5.2. Results

In the results section of this program, after running the source code, the results can detect a person wearing glasses or not, using live video through the front of the person's face. Meanwhile, from the side view, a person's face cannot be detected because the haarcascade dataset can only detect it through the front face display, while the side view cannot detect it. And the program can count the number of faces and eyes of the people in the video. Here are the results of detecting people who wear glasses:



Gambar 5.1: People Without Glasses 1

Figure 5.1 shows taking pictures of people who do not wear glasses through live video. It can be seen that with good lighting, this program can detect the person's face and eyes quickly and accurately. The haar like feature process is running well by being able to select facial and eye features for each pixel value in the black and white squares, then processed using an integral image technique that functions to calculate the pixel values in each black and white square, then processed using an adaptive boosting technique to combining weak classifiers into strong classifiers, and in the final stage, applying the cascade classifier technique to detect the accuracy of face and eye objects on video repeatedly. In the area around the face area there is a red square column and for the area around the eye area

there is a blue square column. And the output issued by the glasses his Off and can count the number of faces as 1 and the number of eyes as 2.



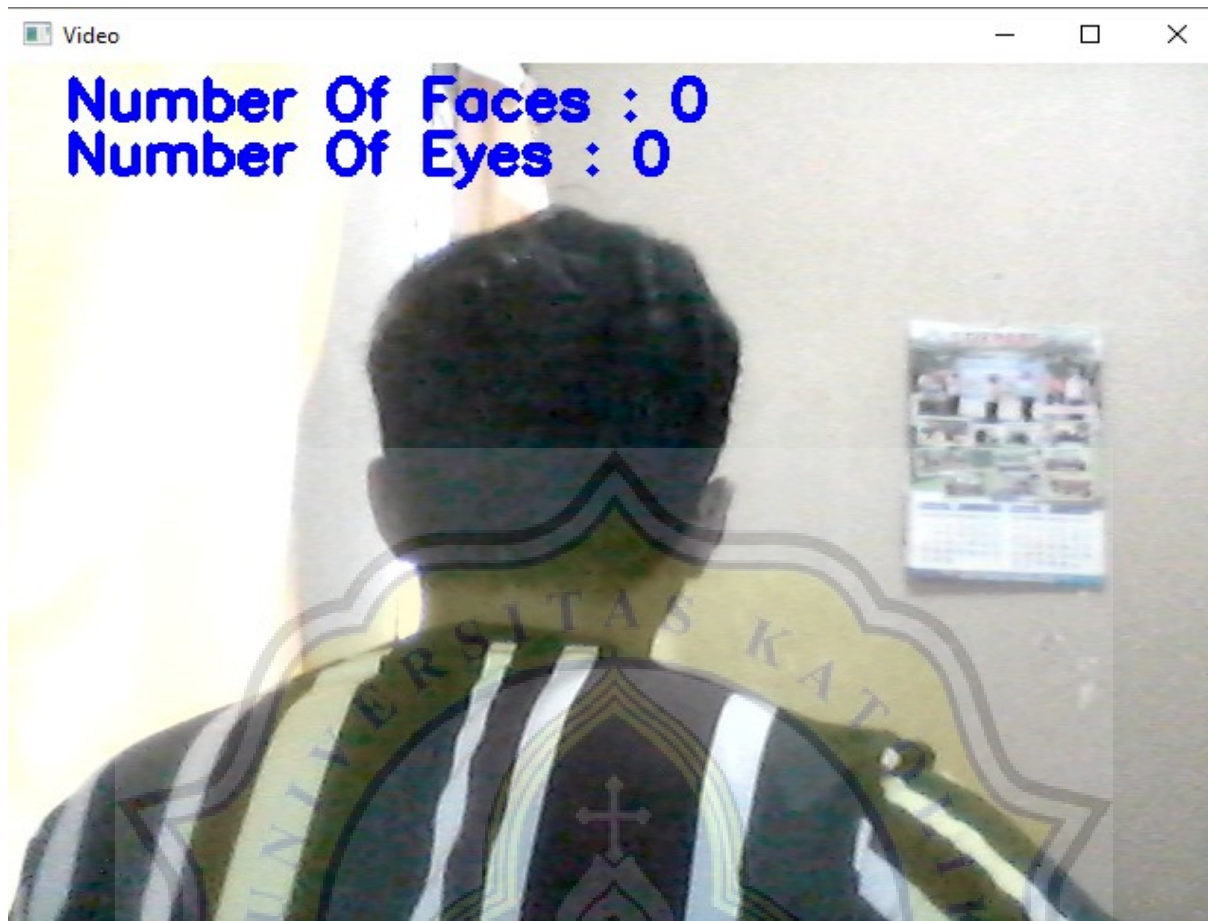
Gambar 5.2: Person with Clear Glasses 1

Figure 5.2, taking pictures of people with glasses using clear colored lenses through live video, can be seen with good lighting so this program can detect the person's face quickly and accurately. The haar like feature process runs well by being able to select facial and eye features for each pixel value in the black and white box, then processed using an integral image technique that functions to calculate the pixel value in each black and white box, then processed using an adaptive enhancement technique to combine classifiers. . . weak into a strong classifier, and in the final stage, applying the cascade classifier technique to detect the accuracy of face and eye objects in the video repeatedly. In the area around the face area there is a green square column, and the output issued by the glasses is On and can count the number of faces is 1 and the number of eyes is 0, because the eye is not detected.



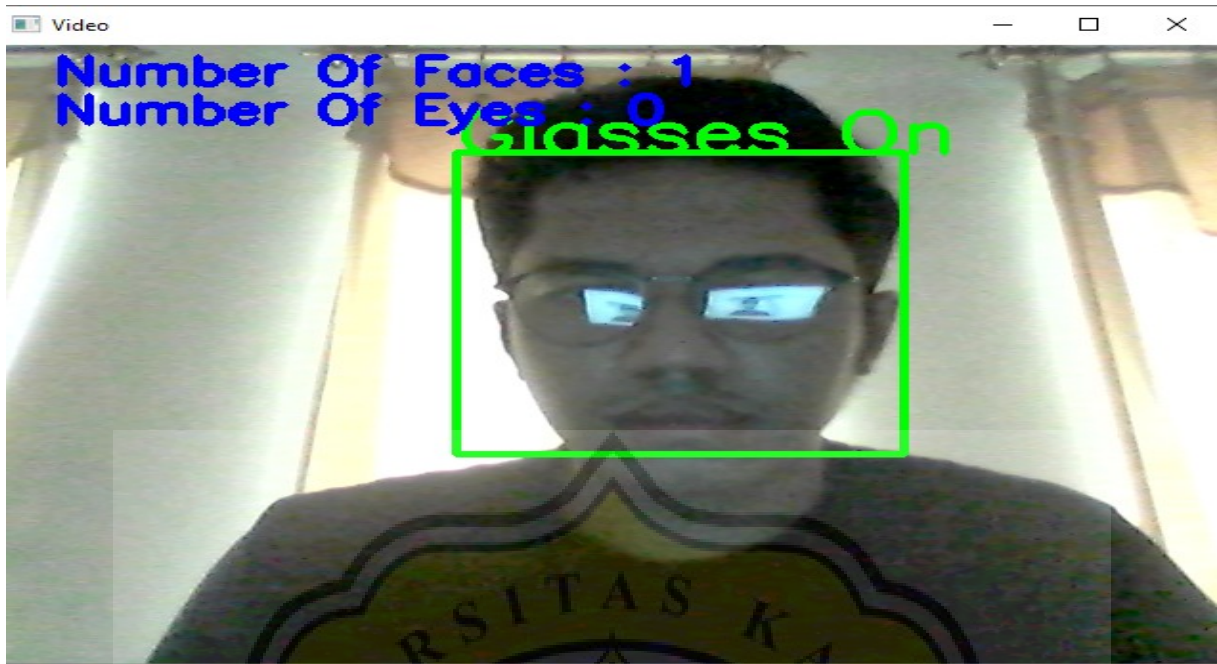
Gambar 5.3: Side View of People with Glasses

Figure 5.3 is a picture of a person with glasses through a live video from a side view of the person's face. It can be seen that the haarcascade classifier method cannot detect a person's face or eyes from a side view because the haar like feature process cannot select facial and eye features at each pixel value in the black and white box, if the haar like feature process does not run then for the integral image process, adaptive boosting, cascade classifier also doesn't work so the glasses detection can't work. And the system cannot count the number of faces and eyes of the person.



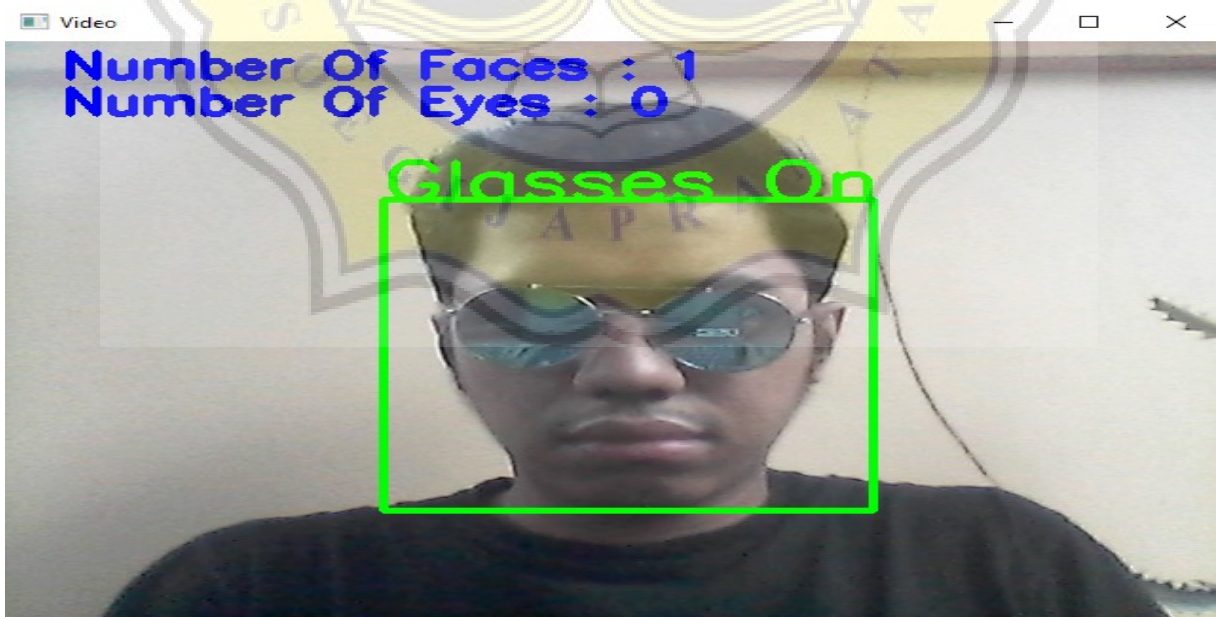
Gambar 5.4: Back View of People with Glasses

Figure 5.4 is a picture of a person with glasses through a live video from the back view of the person's face. It can be seen that the haarcascade classifier method cannot detect a person's face and eyes from the rear view of the person's face because the haar like feature process cannot select facial and eye features for each pixel value in the black and white box, if the haar like feature process does not run then to The integral image process, adaptive boosting, and cascade classifier also don't work, so the glasses detector can't work. And the system cannot count the number of faces and eyes of the person.



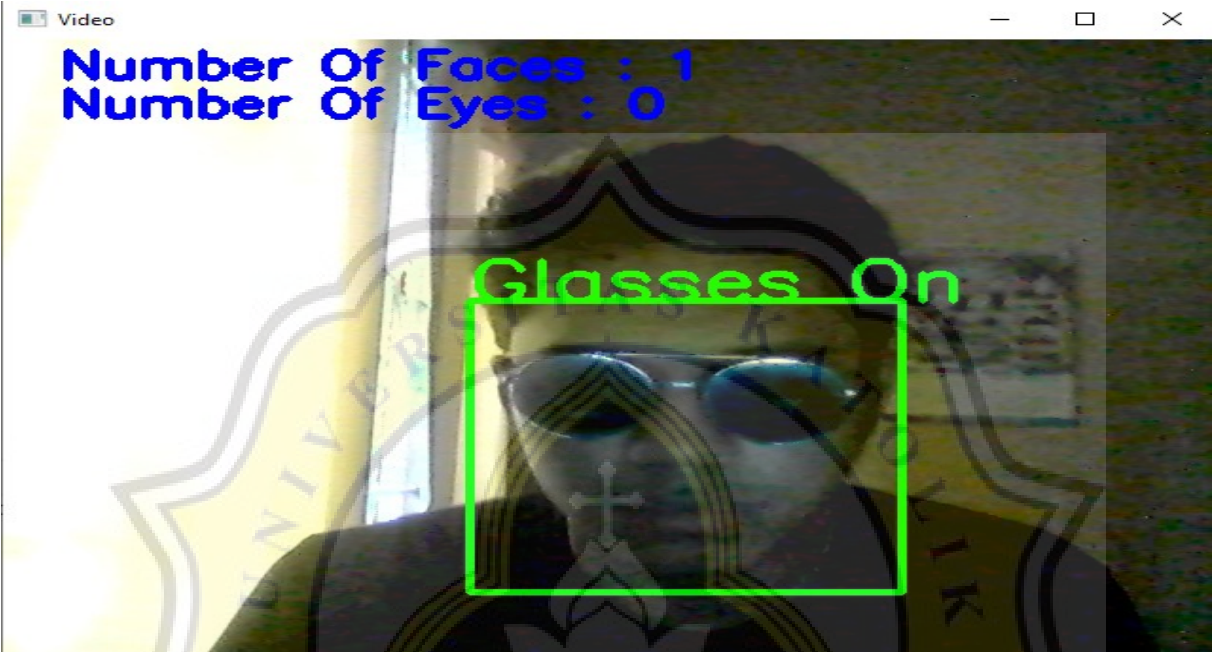
Gambar 5.5: Person with Clear Glasses 2

Figure 5.5 is a shot of a person with glasses via live video using different clear-lensed glasses, and taking a picture with a different background. It can be seen that with good lighting this program can detect a person's face and the glasses being used quickly and accurately.



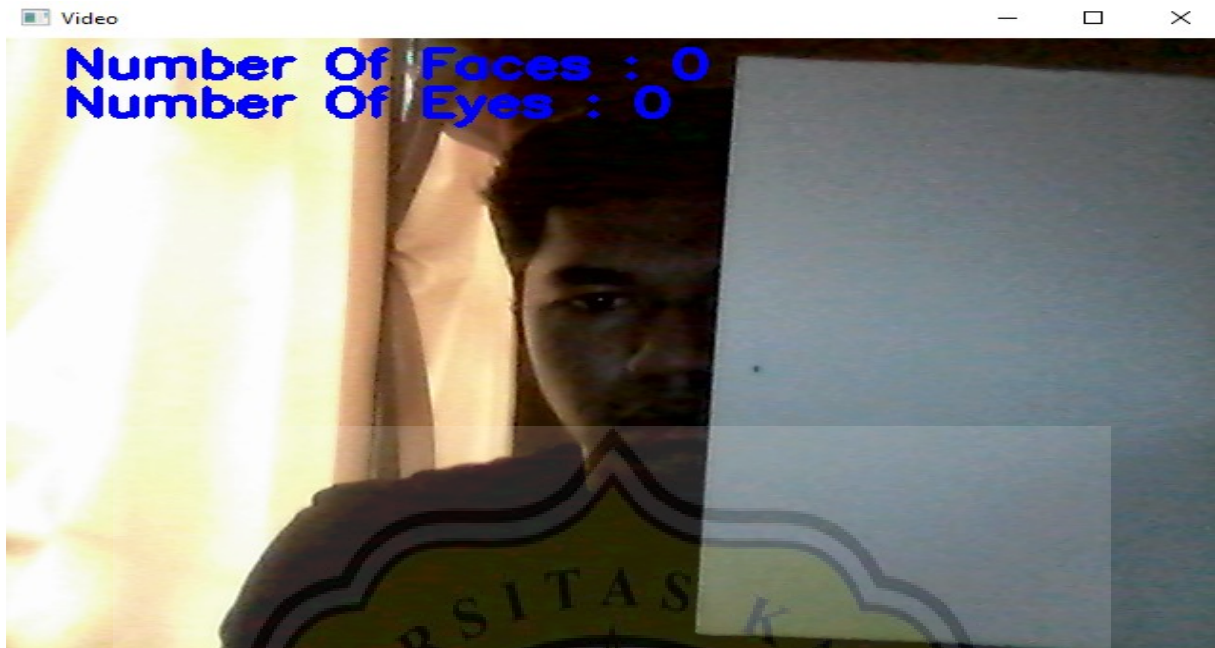
Gambar 5.6: Person with Dark Glasses 1

Figure 5.6 is a shot of a person with glasses via live video using a dark-lensed spectacle frame. It can be seen that with good lighting this program can detect a person's face and the glasses being used quickly and accurately.



Gambar 5.7: People with Dark Glasses 2

Figure 5.7 is a shot of a person with glasses via live video using a dark-lensed spectacle frame. Visible in poor lighting or dark this program can detect a person's face and the glasses being used quickly and accurately.



Gambar 5.8: Other Objects Blocking Face

Figure 5.8 is a shot of a person without glasses via live video. It was seen that there were other objects blocking the face which had an impact on the glasses detection program not running in the video.

