DIRECT DETECTION OF PEOPLE WEARING GLASSES USING THE HAARCASCADE CLASSIFIER

[1]Billy Cornelio Alfons, [2]Robertus Setiawan Aji Nugroho
[1,2]Program Studi Teknik Informatika Fakultas Ilmu Komputer,
Universitas Katolik Soegijapranata
[2]nugroho@unika.ac.id

ABSTRACT

In this day and age, many people wear glasses to help their eyesight or to add style to make them look more attractive. In some places it is mandatory for someone not to wear glasses for certain reasons such as someone who goes to an ATM machine not to wear dark colored glasses so that his face can be seen clearly and is better known for security reasons. In this situation it is very important for some places to be able to detect a person wearing glasses using the camera directly for some reason in order to quickly recognize the person's face more clearly.

The haarcascade classification algorithm is an algorithm that can detect faces and eyes directly and quickly using a camera connected to a computer. The haarcascade that I use is the frontalface haarcascade to detect faces and the eye haarcascade to detect the eyes, and the results of the detection if the face and eyes are detected, it is certain that someone is not wearing glasses and vice versa if only a face is detected, it is certain that someone is wearing glasses. OpenCV to insert live video and processed by the library that we use.

The final result that we get in this project is an image that has been captured and has been run through a dataset, namely direct video input that has been processed using haarcascade frontalface and haarcascade eye and opencv. At the top of a person's face there will be a text that explains whether the person is wearing glasses or not, and can count the number of faces and eyes of a person recorded on the camera.

Keywords: Haarcascade Classifier, OpenCV, Live Video, Glasses

INTRODUCTION

Background

In this day and age, which is increasingly advanced and rapidly growing, many systems are developing to detect goods used by someone with the aim of facilitating community activities. In some places a person is required not to use glasses for certain activities, such as making transactions at an ATM machine. With the development of technology, this can be overcome by using image processing which can determine whether a person is wearing glasses or not. The problem in this case is detecting someone who is wearing glasses or is not wearing glasses.

The haarcascade classifier algorithm is one of the algorithms that can detect a human face, the algorithm is able to detect it quickly and in real time. The haarcascade classifier algorithm has the advantage of fast computation because it only depends on the number of pixels of a photo or video. The haarcascade classifier algorithm can be used to process video directly using haarcascade frontalface to detect faces and haarcascade eyes to detect eyes which can conclude that someone is wearing glasses or not.

In this project, video processing uses the haarcascade classifier algorithm and opencv which aims to detect a person's face and eyes in the video. In the first stage the haarcascade frontalface will detect the person's face and a green square pattern will appear around the face, then the haarcascade eye will detect the person's eyes and a square pattern will appear around the eyes. So if the person's face and eyes are detected, it is certain that the person is not wearing glasses, otherwise if only the face is detected while the eyes are not detected, it is certain that the person is wearing glasses. Caption text appears at the top of the person's face. The program also counts the number of faces and eyes of a person in the video. Therefore I want to help in detecting someone who wears glasses accurately and quickly.

Problem formulation

1. How does the haarcascade classifier algorithm work to detect someone who is wearing glasses or not wearing glasses?
2. What are the advantages of the haarcascade classifier algorithm in its use in real time video?
3. How is the success rate of detecting someone who uses glasses using live video by applying the haarcascade classifier method?

This identification project the author uses Python version 3.8.5. This project uses a library from OpenCV, as well as recording the face of a person wearing glasses using a laptop webcam. And with the hope that the live video can detect someone wearing glasses clearly and accurately and get maximum results.

The purpose of this project is to help determine whether a person wears glasses or not, somewhere it is mandatory not to wear glasses for certain purposes. By using the haarcascade classifier algorithm, it can detect using live video quickly and accurately. Therefore, the existence of this project can be helpful and beneficial for several parties.

LITERATURE STUDY

In this chapter I will describe some relevant literature studies related to the project I am working on. This journal is structured in a narrative manner with certain information, and has the same field of study as the project I will be working on, namely image processing. All journals discuss projects related to the study of the field of image processing but have different methods and final results according to the objectives of each project. As follows :

Hendy Mulyawan [1] here the author identifies objects in real time. Then apply the template matching method to identify the drink bottle in the image. The author focuses its implementation specifically on object identification of several beverage bottles that have different large and small dimensions in the image. They used a drink bottle image dataset of 50 different images. The final result of this project is being able to identify the drink bottle object in the image correctly and accurately, and being able to identify 5 image objects at once.

Dwi Prasetyo [2] here the author detects large and small pimples on the face using 30 sample images. The author focuses its implementation specifically on detecting acne on a person's face with different acne dimensions, namely large pimples and small pimples. Then apply the marking method to determine the sensitivity value in the image, if the sensitivity value in the image is above 0.5 then the acne on the face can be detected clearly. The final result of this project is quite good because 21 image samples have sensitivity values above 0.5 so that large and small pimples can be clearly detected, while the other 9 image samples have sensitivity values below 0.5 which makes small pimples difficult to detect.

Eddy Nuraharjos [3] here the author detects the color of the image on a 2D flat plane. Then apply the center plotting method of image pixels to determine colors on a 2D flat plane, this project uses several 2D sample images to compare from several variations of image file compression and several colors at once in one image object display. The author focuses its implementation specifically on color detection in 2D flat plane images with various and different colors. The final result of this project is that several samples of 2D images can be detected up to 100% maximum when tested on variations of one color object shape, but not good when used in object variations with more than one object color variation. The average color detection results in 2D flat plane images are quite good, although there are still limited plotting layouts that are more varied.

Asti Riani Putrias [4] here the author performs image processing of vehicles on the highway. Then apply the histogram equalization method to identify car and motorcycle objects in the image. The author focuses its implementation specifically on the identification of car and motorcycle objects with different dimensions of the vehicle. They used a dataset of images of car and motorcycle objects taken on the highway while driving as many as 50 different images. The end result of this project is being able to identify the object of a car or motorcycle precisely and accurately on an image that has been taken from a webcam camera on the highway.

Ihsan Nugraha Putra Mukti [5] here the author lists several types of tomatoes. Then apply the RGB detection method to determine the condition of the tomatoes in the image, this project uses 30 different tomato image samples to identify the color of the tomatoes in the image. The author focuses its implementation specifically on sorting tomatoes with various types of fruit which are identified as good and bad conditions of the tomatoes, so that this project can determine which tomatoes are suitable for consumption and not suitable for human consumption. The final result of this project is to get the highest accuracy reaching 88% when taking 20cm using preprocessing with bright light conditions, if the distance is not focused and away from the object, the accuracy decreases. The average compute time for a PC with 2gb ram is 4500ms. The average result of this project is being able to sort tomatoes that are in good condition through the color of tomatoes correctly and accurately.

Rendy Bagus Pratama [6] here the author tested the eigenface method on the parking system. Then apply the eigenface method to test several parking lots. This project uses 90 samples of different parking lot images, and the sample images are taken in different regions and times. The author focuses its implementation specifically on testing the eigenface method on the parking system whether to get maximum results or not to be able to detect parking spaces that are still available. The final result of this project obtained an average of 66.67% of 23 tests. As well as being able to detect parking spaces that are still available and parking spaces that are already fully occupied by vehicles in the image accurately.

Muhammad Taufiq Hidayat [7] here the author makes a traffic light control system. Then apply the prototype method to adjust the traffic light system, this project uses a video that has been recorded at a traffic light at a crossroads and uses a landscape lens. The author focuses specifically on the implementation of a traffic light control system using several cameras that are carried out simultaneously on traffic lights using a Raspberry. The final result of this project is that the use of the Raspberry Pi 2 B+ in the system is still not optimal because it can only use a maximum of 3 cameras and the results obtained for the traffic light control system are not optimal.

Sunu Jatmika [8] here the author detects the ripeness of apples based on their color composition. Then apply the image histogram method to detect the ripeness of apples in the image, this project uses 30 different apple image samples. The author focuses its implementation specifically on detecting the ripeness of apples with different types of apples and fruit colors to detect the ripeness of the fruit. The final result of this project is the percentage of similarity and classification of ripe fruit which includes raw (18%-100%), dried (12%-17%), and ripe (0%-11%). The average result of this project can detect the ripeness of apples based on the color composition in the image.

Dewi Permata Sari [9] here the author identifies braille. Then apply the chain code and sequence alignment methods to identify braille characters directly using a webcam camera. The author focuses its implementation specifically on the identification of various braille characters directly so that it can be seen what braille letters are being identified. The final result of this project is that braille identification can be done using a webcam camera with a maximum distance of 20 cm to get maximum results, while for distances above 20 cm, braille identification does not work optimally. The average result of this project is that image processing can quickly and accurately identify various types of braille in videos.

Hironimus Hendra Setiawan [10] here the author performs the classification of banana species using image processing. Then apply the backpropagation method to classify types of bananas based on their shape, this project uses several sample images of types of bananas such as Ambon bananas, plantain bananas, and milk bananas. The author focuses its implementation specifically on the classification of bananas based on the types of bananas available, the classification of various types of bananas is carried out by carrying out 2 variations of the model, namely traingdx and the logsig layer transfer function and the purelin output transfer function. The test model used is a 5-fold cross validation based on epochs, goals, and learning rates from testing using holdout validation. The final result of this project is that the best accuracy of 1 hidden layer is 100% with a time of 1.6 seconds and with the highest average accuracy of 89.95% with a time of 6.2 seconds. Then use 2 hidden layers of 100% with a time of 22.1 seconds with the highest average accuracy of 89.95% and an average time of 19 seconds. The average result of this project is that image processing with the backpropagation method can classify the types of bananas based on their shape precisely and accurately.

Risfendra [11] here the author is working on a wheeled goalkeeping robot movement system. Then apply the wall follower method to run the wheeled goalie robot movement system. The author focuses specifically on the implementation of the wheeled goalie robot movement system based on the shape and color of the ball and the goalie robot moves to block the incoming ball. The end result of this project is that the tool can work well in detecting the color of the ball and the movement of the robot according to the analysis carried out. The average result of this project is the maximum linear speed traveled by the robot of 1.59 m/s. And the efficiency of the moving time analysis test with the actual distance is 86.77%.

Albert Liem [12] here the author detects an empty parking lot in a video. Then apply the Thresholding and SVM methods to identify vehicles in the parking lot. The author focuses its implementation specifically on detecting empty parking spaces or not through videos that have been taken from the parking lot using the opencv library to process the video. The end result of this project is that the program can work well in video input with bright lighting, while for video input with low lighting, the program will run less than optimally. The average result of this project is to be able to find out and tell the user whether the parking slot is empty or not using indicators.

Amelia Yolanda [13] here the author automatically counts platelet cells for early detection of the degree of dengue fever. Then apply the template matching method to identify several samples of human blood cell images. The author focuses its implementation specifically on the automatic counting of platelet cells to detect the degree of dengue fever in a person early. They used a dataset of 3 different sample images of human blood cells. The end result of this project is that the sample image needs to go through several processes, including segmentation and color manipulation where segmentation serves to separate objects and backgrounds, while color manipulation is to increase contrast and align colors. The average result of this project is that it can automatically count human platelets to detect the degree of dengue fever.

Halim Abdillah Soleh [14] here the author compares the Lampung script using the Roberts and Sobel edge detection method. Then apply the Roberts and Sobel edge detection method using a dataset of 20 different Lampung script images to identify the type of character. The author focuses its implementation specifically on the comparison of the Roberts and Sobel edge detection method to the identification of Lampung script types based on the Lampung script letter pattern. They conducted research which was divided into two stages, namely the training stage and the testing stage. The final result of this final project the author can conclude that the detection of Lampung script is better using the Sobel method than using the Roberts method because the Roberts method gets an error percentage of 28.5% while the Sobel method gets an error percentage of 14.5. %. The recognition error on the test equipment is influenced by several things, namely the shape of the character, image processing, the diversity of handwriting forms, and the determination of the parameters of the artificial neural network.

Bambang Marhaenanto [15] here the author identifies coffee roasters based on variations in roasting degrees. Then apply the RGB method to detect coffee beans based on the age of the coffee beans. The author focuses its implementation on identifying coffee roasters based on variations in roasting degrees, and detecting coffee beans based on the age of the coffee beans, the purpose of this study is to help determine the quality of roasted coffee. They used a dataset of 9 sample images of different types of roasted coffee. The final result of this project is that the predicted roasting time for standard variations of roasting degrees for Robusta coffee with a temperature of 200 degrees Celsius is 101.7 minutes for the lowest roasting rate and 158.2 minutes for the highest roasting level, while for Arabica coffee at a temperature of 200 degrees Celsius. was 31.6 minutes for the lowest roast level and 44.7 minutes for the highest roast level. The average results of this project can calculate the RGB values quantitatively for Arabica and Robusta roasted coffees.

Fajar Ananda Saputra [16] here the author conducts waste monitoring on thingspeak-based drainage. Then use the Thresholding method to convert the image to grayscale and remove the background on the image to detect junk images. The author focuses its implementation specifically on monitoring waste in the drainage and changing the image to grayscale to remove the background in the image which aims to detect garbage objects in the image. They use the dataset in the form of live video input. The purpose of this study is to have a direct waste monitoring system to minimize the occurrence of flood disasters. The end result of this project is a drainage monitoring system with thingspeak-based water level and garbage indicators along with data that will continue to be updated and will read conditions in real time.

Hendrick [17] here the author measures the water level in a box. Then apply the ESP-32 method to measure the water level using a red object in a box filled with water as an indicator. The author focuses its implementation specifically on measuring the water level in a box using a red buoy object indicator and the ESP-32 method as a water level monitoring tool and using a float as a measured object. They used a dataset of 10 samples of water level images on different boxes. The final result of this project is to get a change in pixels as a measurement tool, the pixel changes are caused by the ups and downs of objects due to changes in the water level in the box. Data retrieval is calculated through pixels from the contour area by taking from a water level of 1 cm - 10 cm, the accuracy of the measurement is 1 cm - 7 cm with a relative error value of 1.26%, while for a height of 8 cm the measurement is inaccurate with a relative error value of 250.315. The average result of this project is that the system can measure the water level in the box using a red object indicator to be processed using the ESP-32 method.

Abdul Jaliladju [18] here the author detects rupiah banknotes based on the nominal money. Then apply the thresholding method to several images of rupiah banknotes with different nominal values to determine the nominal value of the rupiah banknotes. The author focuses its implementation on detecting rupiah banknotes based on the nominal money using Raspberry Pi image processing. They used a sample image dataset of 5 different rupiah banknotes, such as the 5,000 rupiah, 10,000 rupiah, 20,000 rupiah, 50,000 rupiah and 100,000 rupiah denominations. The final result of this project is the system can identify rupiah banknotes based on their nominal value using raspberry-pi image processing.

Edy Winarnoe [19] here the author detects the edges of objects in real time video. Then apply the canny detection method to identify several digital object images. The author focuses its implementation specifically on detecting the edges of objects using live video, as well as applying the canny detection method to identify several images of digital objects such as human objects, and even inanimate objects. They use a dataset in the form of live video recorded by a webcam. The end result of this project is that live video can detect the edges of objects, such as the edges of human faces, starting from the ears, nose, and eyes. Likewise with being able to detect the edges of inanimate objects such as rolls of duct tape accurately.

Taftyani Yusuf Prahudaya [20] here the author classifies the quality of guava. Then apply the KNN method to 80 different guava sample images based on their conditions. The author focuses its implementation specifically on the quality classification stage of guava based on the condition of the fruit. The system uses image processing to extract the color and texture features of guava fruit using the KNN method. The system will classify guava into 4 qualities, namely super class, A class, B class, and outside quality. The KNN method is designed with 7 extraction features as input, namely the average RGB value, defect area, GLCM value with the output of these 4 qualities. The purpose of this study is to determine the quality of guava that is suitable for human consumption and guava whose quality is not good and not suitable for human consumption. The final result of this project is that this classification method is able to provide the best accuracy at k=3 in the KNN method with an accuracy of 91.25%. The average result of this project is that the system can determine the quality of guava based on its condition, namely defects and rot.

Khabib Abdulla [21] here the author classifies the maturity of rice. Then apply the Laplacian of Gaussian method by using several images of rice samples based on conditions and types. The author focuses its implementation specifically on the stage of classifying the level of rice maturity based on the condition and type of rice, the level classification in this final project is processed by image processing which includes the classification of good rice, medium rice, and bad rice. The purpose of this project is to determine the level of quality of good and edible rice and the level of poor quality of rice so that it is not suitable for consumption. This study uses the python programming language and uses the opencv library as an image processing of rice samples. The final result of this project is the calculation of the percentage of good rice classification with a value of 69.23%, while for medium rice it is 32%, and for bad rice it is 85.29%. The average result of this project is that the author can classify rice with good, medium, and poor quality based on the calculation of the percentage of rice eligibility classification.

Dimas Rossiawan Hendra Putra [22] here the author identifies faces based on skin color segmentation on the face. Then apply the Naive Bayes method using 70 different sample images of people. The author focuses its implementation specifically on facial recognition based on skin color segmentation on the face. The final result of this project is that from a total of 70 test data and a threshold of 0.1, the system accuracy is 90.25%. The results of the questionnaire stated that overall the applications made were quite good and could be used to identify the faces of server room workers. The average result of this project is that several facial images of people have been identified based on skin color segmentation on the face.

Oky Dwi Nurhayati [23] here the author identifies ordinary chicken eggs and omega chicken eggs. Then apply the thresholding method to be able to identify chicken eggs using some pictures of ordinary chicken eggs and omega chicken eggs. The author focuses its implementation specifically on the identification of ordinary chicken eggs and omega chicken eggs through three stages, namely proving the image pre-processing algorithm with the histogram equalization method, the next step is to perform the image segmentation process using the floating method, and the last step is to recognize the egg image pattern. By analyzing first-order texture statistics from the image histogram by calculating the mean, variance, skewness, and kurtosis values. The end result of this project is the program can distinguish ordinary chicken eggs. and omega chicken eggs through three stages of accurate identification.

I Gede Susrama Masdiyasamas [24] I Gede Susrama Masdiyasamas[24] here the author identifies the number plate of a motorized vehicle. Then apply the KNN and threshold methods to identify 20 vehicle number plate sample images, both light and dark. The author focuses its implementation specifically on the process of identifying vehicle number plates based on the number from the plate, both light and dark. This research goes through three stages of the identification process, namely threshold, contour, and KNN. The end result of this project is a multi-step image processing method capable of recognizing vehicle license plates with a license plate recognition accuracy of 95% using a camera with 5 megapixels or more, while for cameras with under 5 megapixels, identification of license plates becomes difficult and inaccurate. This program can detect vehicle license plates with a light base color by performing an inverse so that the character recognition algorithm runs optimally.

Riyan Latifahul Hasanah [25] here the author identifies the types of pears. Then apply the adaptive neuro fuzzy inference system (ANFIS) method by using several pictures of pears based on the type of fruit. The author focuses its implementation specifically on identifying the types of pears based on the characteristics and shapes of the pears. This identification process is divided into several stages, namely by converting the RGB image into a Lab, then the segmentation process using the K-Means Clustering algorithm. The segmented image is extracted into seven features, namely six color features (RGB and HSV) and one size feature. And the last stage is applying the ANFIS algorithm for the final classification stage, and the results show high accuracy in the identification of pears. The final result of this project is to get a high accuracy value of 90% for monster pears, and 100% accuracy values for William pears, and 95% accuracy values for whole pears. The average result of this project is that the author can classify between monster pears and william pears.

Fachri Ibrahim Nasution [26] here the author is grading the salak fruit of Padang Sidempuan. Then apply the grading method using 10 sample images of Padang Sidempuan salak fruit. The author focuses its implementation specifically on the assessment of the Padang Sidempuan salak fruit based on the characteristics and conditions of the fruit and the water content of the salak fruit. The final result of this project is a major model assessment program with a fit rate of 65.0% with an average validation test error of 16.37% and an R value of 0.833. Intermediate model with a 78.33% fit level with an average validation test error of 18.45% and an R value of 0.882. Minor model with a 70.0% match rate with an average validation test error of 12.66% and an R value of 0.8879%. The mean model with a 76.7% fit level with an average validation test error of 8.99% and an R value of 0.9528. The average results of this project, namely the grading method on Padang Sidempuan salak fruit has a high accuracy so that the weight of the fruit with the area of processing results has a very close relationship.

Zulham Effendi [27] here the author detects macronutrients in the leaves of oil palm plants. Then apply the Sobel filter method using 400 sample images of oil palm leaves based on their condition. The author focuses its implementation specifically on the process of detecting macro nutrients contained in oil palm leaves based on the condition of the oil palm leaves. The final result of this project is that the histogram value of the detection of nutrient deficiency N shows the tolerance limit value, namely the mean value of 46.69, standard deviation of 38.19, median value of 40.89. While the histogram value of the detection of healthy leaves shows the tolerance limit value, namely the mean value of 43.34, standard deviation value of 31.75, median value of 36.43. The average result of this project, namely the histogram value in detecting healthy oil palm leaves and nutrient deficient oil palm leaves, shows accurate and good results.

Yusuf Yunus [28] here the author detects the surface structure of the metal being processed. Then apply the neural network method using several iron images from the treatment. The author focuses its implementation specifically on detecting metal surfaces that are being processed based on iron images that have been processed using image processing and get matrix values. The final result of this project is to detect metal surfaces that have been heated by cooling water, cooling with oil, and cooling with air, getting a recognition percentage of around 60% for the parameters of the input unit 50, output unit 50, hidden layer 2, hidden unit 30, and 1000 iterations. The average result of this project is that the image pattern for each treatment has almost the same shape so that the resulting matrix is the same.

Zulham Effendizu [29] here the author detects yellow line disease on oil palm leaves. Then apply the Sobel filter method using 240 sample images of healthy leaves and yellow line disease leaves of oil palm plants. The author focuses its implementation specifically on the detection of yellow line disease on oil palm leaves based on the quality and condition of the oil palm leaves. This detection project is divided into three different conditions of oil palm leaves, namely mild, moderate and severe symptoms of oil palm leaf disease. The purpose of this study is to determine the level of greenness of oil palm leaves so as to determine the fertility of the oil palm plant. The final result of this project is that mild symptoms of oil palm leaf disease reached 29.6%, while for moderate symptoms of oil palm leaf attack it reached 16.7%, and severe symptoms of oil palm leaf attack reached 11.25%. The average yield of this project, namely the yellow line attack on oil palm leaves reached 30%.

Ayu Fitria Amalia [30] here the author detects the iris and pupil in the human eye. Then apply the sobel edge detection method using several different eye images. The author focuses its implementation specifically on the detection of the iris and pupil in the human eye based on the condition of the human eye. In pupillary localization, there are two stages, namely detection of sobel edges and morphological operations. As for iris localization, there are three stages, namely median filter, sobel edge detection, and morphological operations. The final result of this project is that the process of extracting the iris features using sobel edge detection has not been successful, through the pupillary extraction sobel edge detection is more successful than the iris.

Various other journals related to my current project are also used as references in this research. But most of the journals only describe studies in the same field whereas the methods and algorithms are different from the projects I worked on. The project I am working on is detecting someone who wears glasses using the haarcascade classifier algorithm and using the dataset, namely haarcascade_frontalface.xml to detect faces and haarcascade_eye to detect eyes, and use real time video input to perform the detection. This is the description of the project I am working on in this field and is the main focus of this research.

ANALYSIS, METHOD AND DESIGN

This chapter contains the steps that will be taken in working on this project. Starting from entering the dataset into the python notebook folder to the final stage of the research. The steps are as follows:

Dataset Pre-Processing

The datasets used in this research are haarcascade_frontalface_default.xml and haarcascade_eye.xml files. Here is the process of getting the haarcascade dataset into the python notebook project folder:

1. The first step is to retrieve the dataset. The dataset used in this project is taken from Github[1], and to run this project the dataset must be entered into the python notebook project folder so that the dataset can function to detect a person's face or eyes.

2. The first dataset is the Haarcascade_frontalface_default.xml file which contains code to detect faces such as using a subwindow with 24x24 dimensions, has a maximum number of 211, has 25 stage numbers, has a process threshold for face detection, has an internal node value, and has a leaf value. The haarcascade_frontalface_default.xml file also contains classification steps that go through the num and tree stages, in the tree there is one

node that contains the haar (rects) feature. The first and second column numbers in the box indicate the position of the pixel being classified, while the third and fourth column numbers indicate the width and height of the feature, and for the last column number, the constant is multiplied to each side of the feature.

3. Meanwhile, the contents of the haarcascade_eye.xml file also contain codes to detect eyes such as using a 20x20 subwindow, having 93 weak counts at most, having 24 stage numbers, having a threshold value for eye detection, and having an internal node value. The haarcascade_eye.xml file also contains a classification step that goes through the numand tree; in the tree there is one node that contains the haar (rects) feature. Each number in rects serves the same purpose as the haarcascade_frontalface_default.xml file. The purpose of the rects for the first and second column numbers is to indicate the position of the pixel being classified, while the third and fourth column numbers indicate the width and height of the feature, and for the last column number, which serves to multiply the constants on each side of the feature.

Video Stream

To detect faces and eyes by processing a dataset of xml files using a cascade classifier, I used the laptop's built-in webcam as well as live video connected to the program. Webcams also make it easy to process messages processed by the cascade classifier against xml file datasets quickly. Live video on the webcam camera is converted every frame into grayscale, so that the face and eye variables that are processed using the cascade classifier will be converted to grayscale.

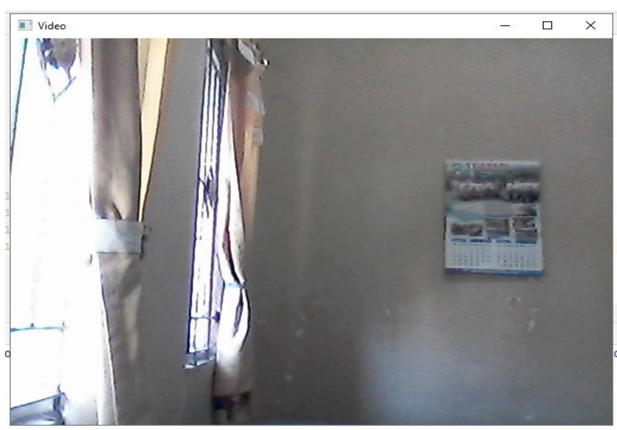

Figure 1. The Process Of Turning On The Webcam

Eye Detection

In detecting faces in video, each frame will be processed directly on CV2, and the cascade classifier functions to read the haarcascade_frontalface_default.xml dataset quickly and accurately so that it can detect images of objects on faces. And by providing the code cv2.rectangle to provide a box image around the face area, and cv2.putText code to provide a description of the face above the box image. As for eye detection, each frame will be processed on CV2, and the cascade classifier functions to read the haarcascade_eye.xml dataset file so that it can detect images of objects in the eye. And by providing the code cv2.rectangle to provide a box image around the eye area, and cv2.putText code to provide an eye description above the box image.

By detecting a person's face and eyes in a live video on the webcam, it can be tested whether someone is wearing glasses or not. This way, if face and eyes are detected, it ensures that a person is not wearing glasses, as well as the code cv2.rectangle to provide a square image around the red face area and the code cv2.putText to provide information that the person is not wearing glasses. Meanwhile, if only the face is detected while the eyes are not detected, then it is certain that the person is wearing glasses, as well as the cv2.rectangle code to provide an image of the box around the green and cv2 face area. putText code to provide information about a person wearing glasses. The project also counts the number of faces and eyes of a person in a live video.

Analysis

The main objective of making this project is to be able to detect someone who is wearing glasses or not by using video input directly, and to apply the haarcascade classifier method in detection. Based on the problem formulation that has been described, the first is how the haarcascade classifier algorithm works to detect someone who is wearing glasses or not. In that case, it is necessary to discuss how the haarcascade classifier algorithm works itself. As described below:

1. Convert video directly to grayscale.
2. After converting the video to grayscale, then use the edge feature, line feature, four rectangle feature which is used to detect features on the face and eyes.

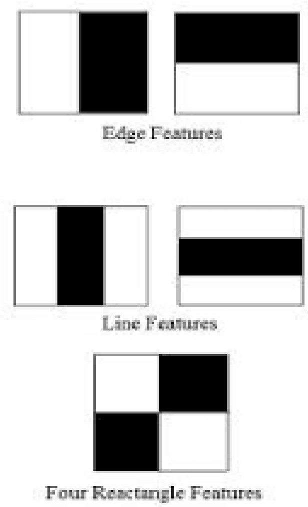

Figure 2. Haar-Like Feature

3. The next stage is the formation of a 24x24 sub-window according to the contents of the haarcascade classifier dataset. As follows:

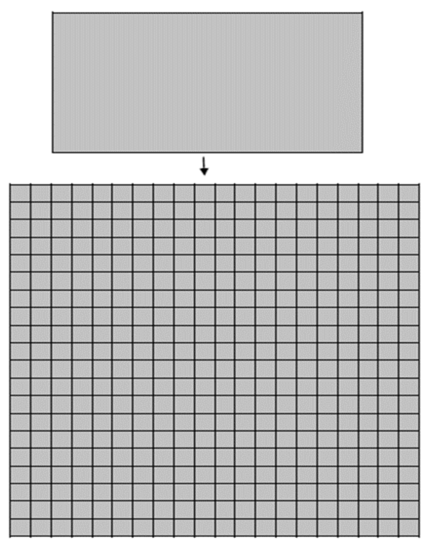

Figure 3. Haarcascade Sub Window Formation

4. After the 20x20 sub-window has been created, the next step is to determine the haar features at stage 0 and tree 0. Black pixels are with feature value = -1, while white pixels are pixels with feature value = (-1+2) = 1. Below are examples of haar features at stage 0 and tree 0:

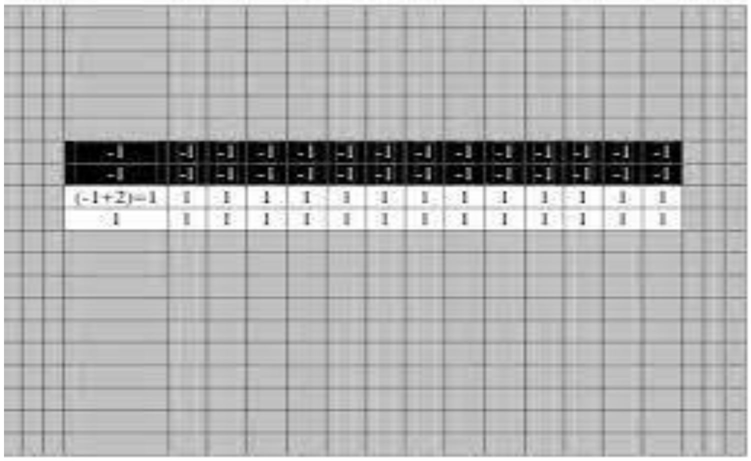

Figure 4. Haar Features At Stage 0

5. After determining the haar feature at stage 0, the next step is to calculate the pixel value using an integral image and compare the pixel values in the light and dark areas. If there is a difference in pixel values in the light and dark areas, it is called the threshold value, the threshold value is declared to have features or objects in the video.

6. The next stage is to apply the adaptive boosting technique to combine many weak classifiers to form a good classifier combination, the results of this adaptive boosting produce strong classifiers and basic classifiers.

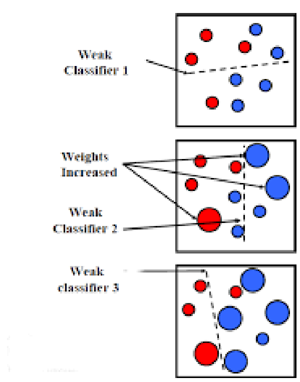

Figure 5. Adaptive Boosting Classifier

7. After generating a strong classifier and a basic classifier, the next step is to develop a stratified classification that aims to determine whether there are facial and eye features in the video.

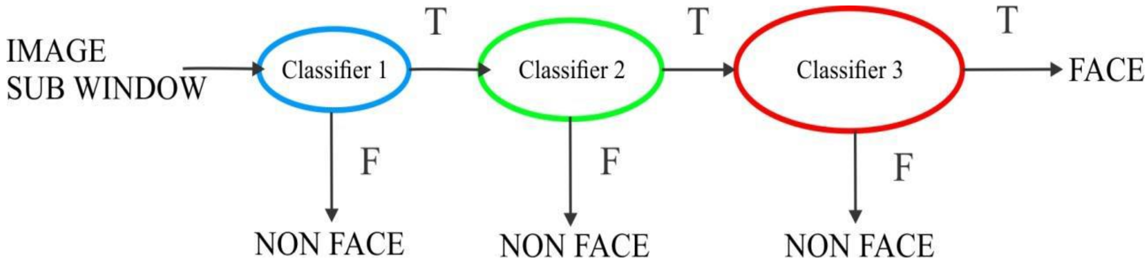

Figure 6. Cascade Classifier

8. If the face and eye detection stage in the cascade classifier has been carried out, then the next step will determine the results of a person using glasses or not, if the face and eyes are detected, it can be ascertained that the person is not wearing glasses. Meanwhile, if only the face is detected then the person is wearing glasses. The following are the results of face and eye detection that have been carried out:

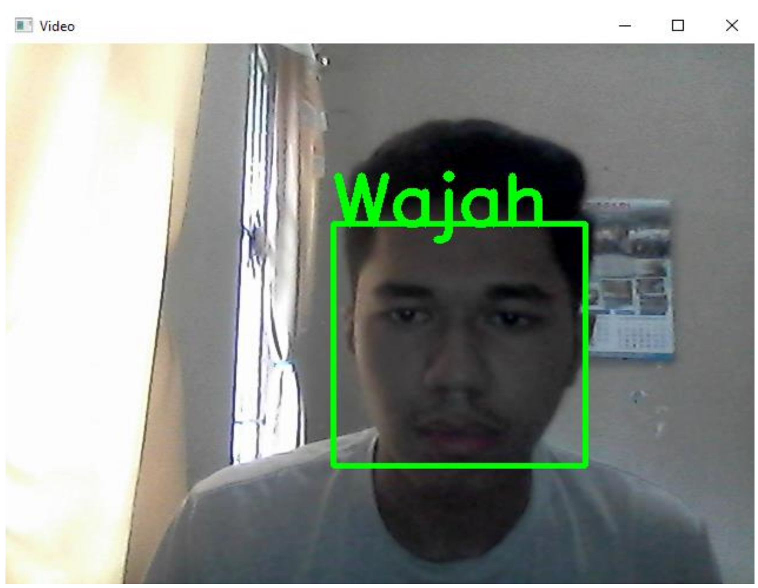

Figure 7. Face Detect

Figure 7, the results of detecting a person's face have been carried out with a haar-like feature process that functions to determine black pixels and white pixels on each face object, then processed using integral image techniques accompanied by adaptive boosting. In the last stage, a cascade classifier technique is applied which functions to determine face objects in the video directly. In determining the object of a person's face, the program will form a green square column around the face area along with face writing at the top of the square column.

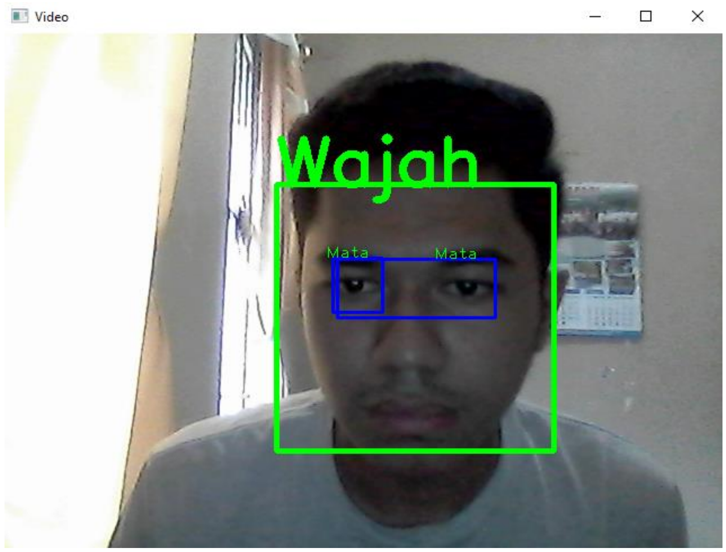

Figure 8. Detect Face And Eyes

Figure 8, the results of detecting a person's face and eyes have been carried out through a haar-like feature process that functions to determine black pixels and white pixels on each face object or eye object. Furthermore, it is processed using an integral image technique along with an adaptive boost technique. In the last stage, applying the cascade classifier technique which serves to determine the face and eye objects in the video directly. In determining a person's face and eyes, the program will form a green square column around the face area and a blue square column around the eye area, along with face and eye writing at the top of the square column.

The second problem formulation discusses the advantages of the haarcascade classifier algorithm or method in its use in real time video. In the case of detecting objects as well as faces and the human eye using the haarcascade classifier method in real time video, it has advantages as described below:

1. The haarcascade classifier algorithm is able to detect quickly because it only depends on the number of pixels in the square of an image or video.
2. The haarcascade classifier algorithm can detect objects in real time and accurately because real time videos are processed using the haar-like feature to compartmentalize each area on the video which aims to find whether there are objects in the video, after going through the haar-like feature, it is processed using integral image technique that aims to calculate the pixel value in the video, then applies an adaptive boosting technique that aims to produce a strong classifier, and the last is processed using a cascade classifier technique which aims to measure the accuracy of object detection on video.
3. The haarcascade classifier algorithm can detect objects in real time using a pixel camera that is not very good, for example a laptop webcam camera.

The last problem formulation discusses the success rate of detecting someone who uses glasses using live video by applying the haarcascade classifier method. In this project, the success rate can be stated as good because it can detect a person's face and eyes using the laptop's built-in webcam which has less high pixels. And in doing the detection it must be with bright lighting conditions so that the program can run smoothly, as well as detecting someone who uses glasses can only be detected from the front view of the person's face while from the side it can't detect glasses.

Design

The use of flowcharts aims to determine the process or procedure of a program so that it can make it easier to understand the program to be run. The system flowchart can be described as follows:

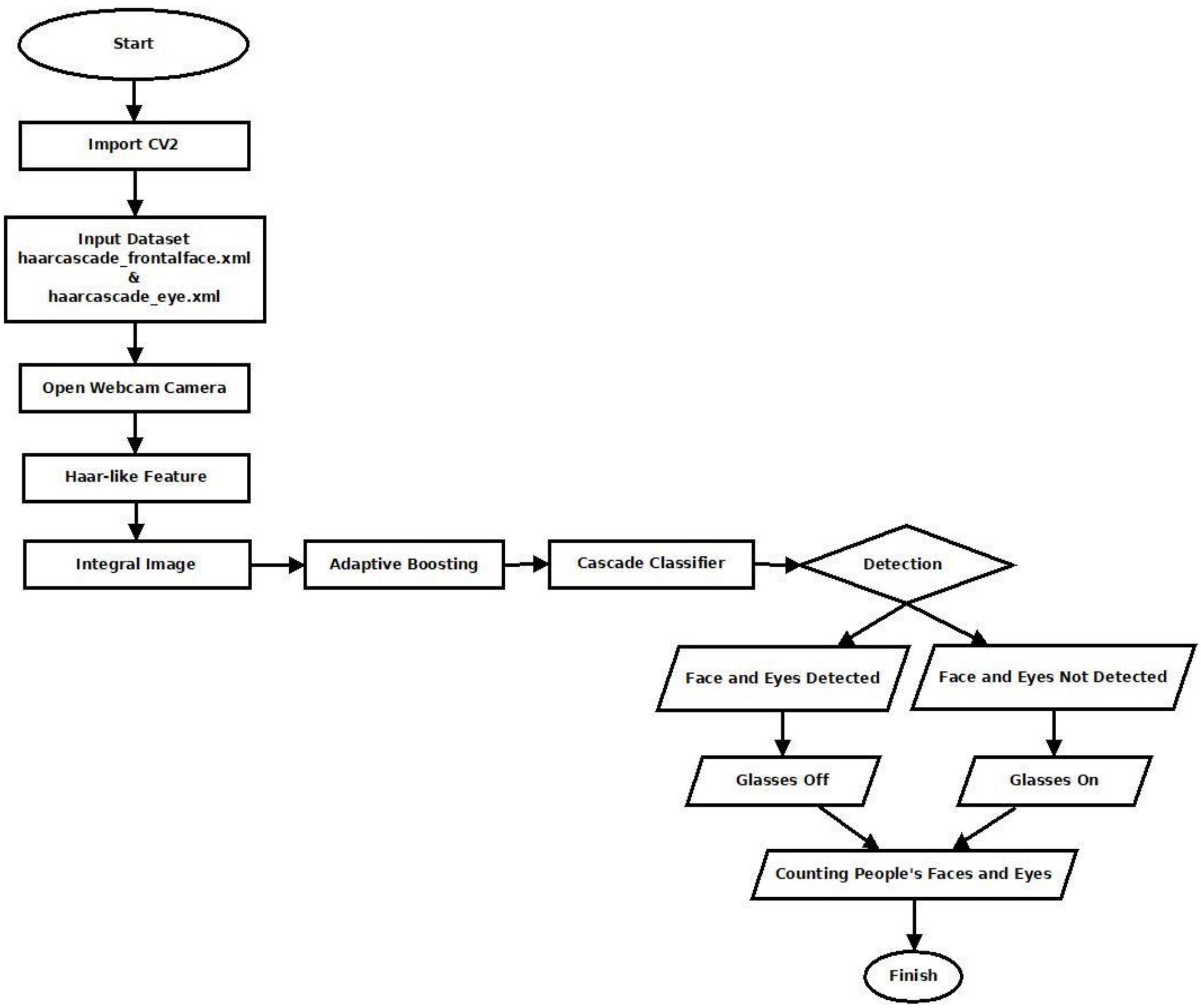

Figure 9. Glasses detection flowchart

Figure 9 is used as an illustration or flow of this project system. In general, the glasses detection system using the haarcascade classifier algorithm does not directly detect the glasses but goes through the stages of detecting the face and eyes first as below:

1. **Import CV2** : OpenCV is used to process data in the form of images or videos, while CV2 is used for face detection.
2. **Input Dataset haarcascade_frontalface.xml and haarcascade_eye.xml** : haarcascade_frontalface.xml is used to detect someone's face, while haarcascade_eye.xml is used to detect someone's eyes.
3. **Open Camera Webcam** : opens the laptop's webcam camera to detect the person's face and eye objects.
4. **Haar-like feature** : the video will change the scale to gray and select facial features and eye features and then determine the pixel value in each black and white box even though the process is not visible.
5. **Integral image** : calculate the difference in pixel values in each black box and white box and get a threshold value which can be interpreted as an object in the video.
6. **Adaptive Boosting** : combining weak classifiers to make a better classifier, and the adaptive boosting process produces a strong classifier.
7. **Cascade Classifier** : multilevel classification that aims to detect the accuracy of a person's face and eye objects in the video until it is successful.
8. **Face And Eyes Detected** : the object of a person's face and eyes is detected, it is certain that the person is not wearing glasses, and the output produced is glasses off.
9. **Face And Eyes Not Detected** : only the object of a person's face is detected while the eyes are not detected, it is certain that the person is wearing glasses, and the output produced by glasses is on.
10. **Counting People Face And Eyes** : count the number of faces and eyes of the person in the video.

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.

Figure 10 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.

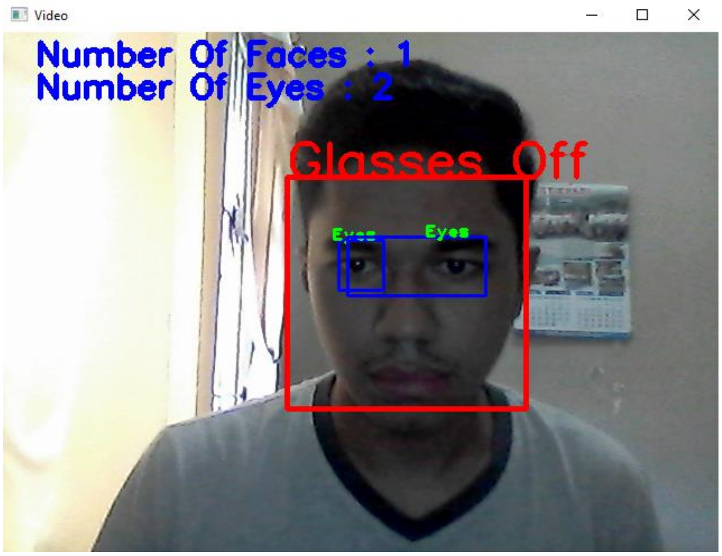

Figure 10. People Without Glasses 1

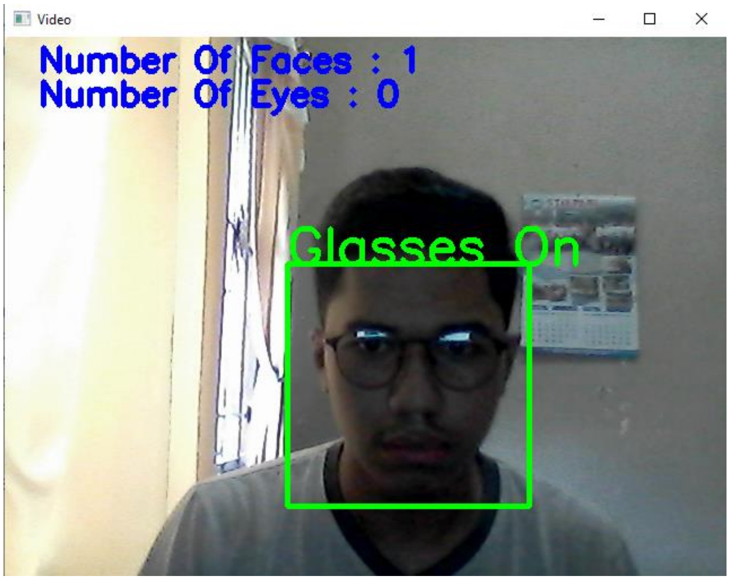

Figure 11. Person with Clear Glasses 1

Figure 11, 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.

Figure 12. Side View of People with Glasses

Figure 12 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.

Figure 13 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.

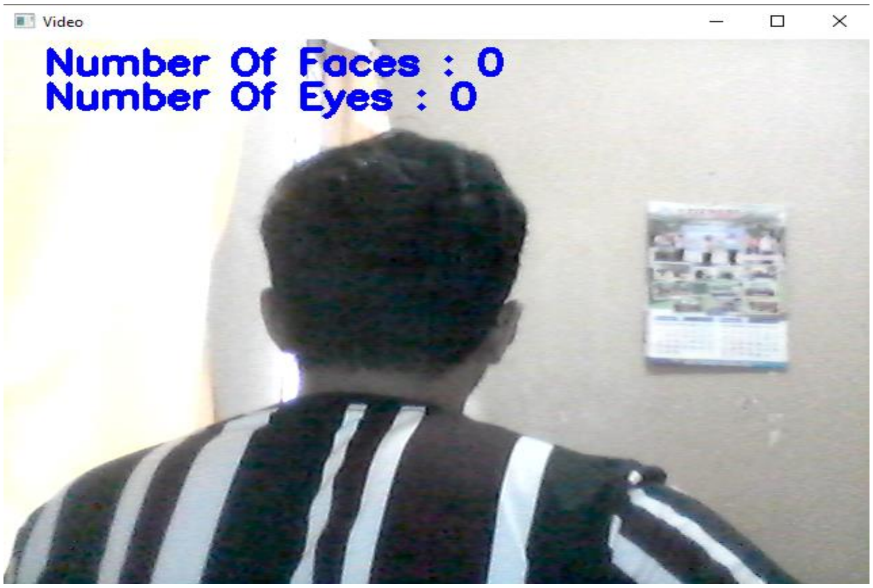

Figure 13. Back View of People with Glasses

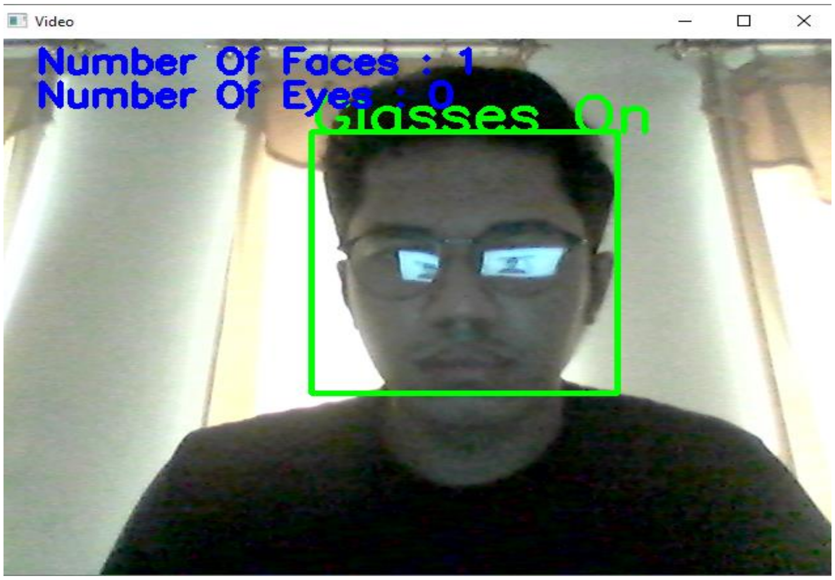

Figure 14. Person with Clear Glasses 2

Figure 14 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.

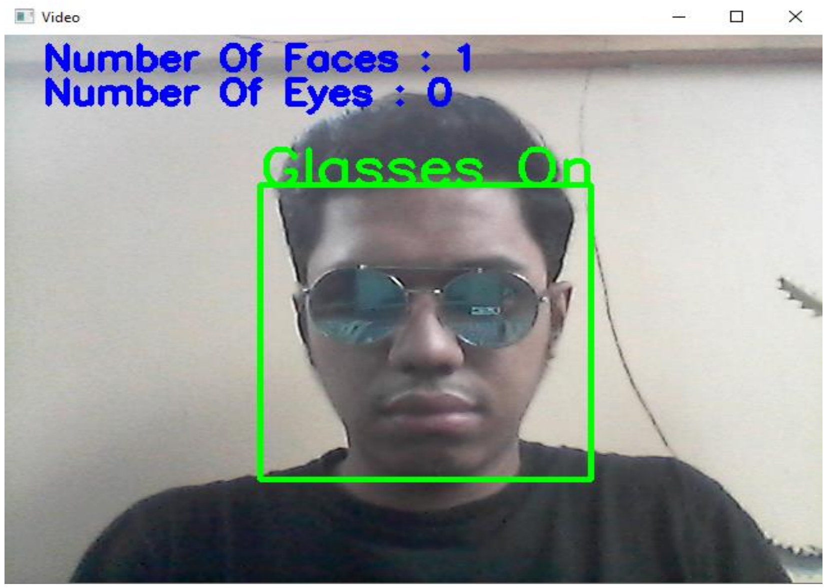

Figure 15. Person with Dark Glasses 1

Figure 15 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.

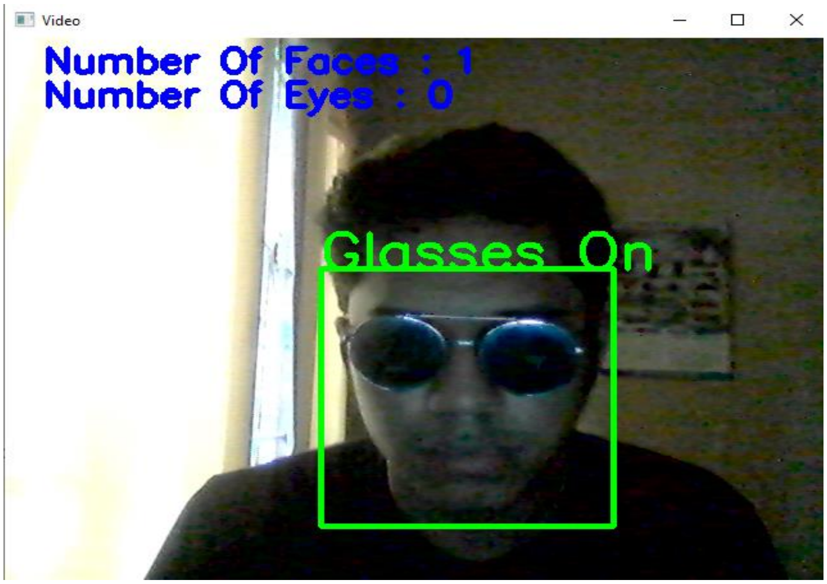

Figure 16. People with Dark Glasses 2

Figure 16 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.

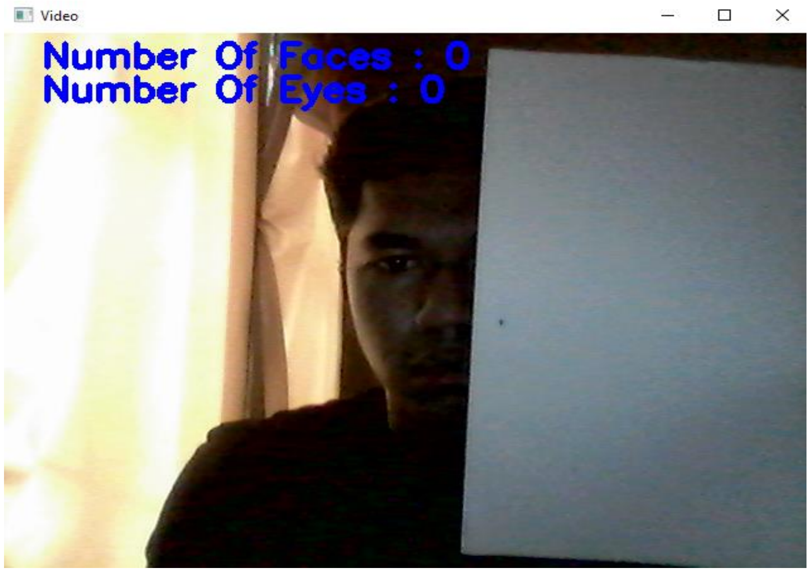

Figure 17. Other Objects Blocking Face

Figure 17 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.

CONCLUSION

Based on research conducted on the project of detecting a person using glasses through live video and the application of the haarcascade classifier method, it can be concluded that:

1. The results of detecting a person using glasses are going quite well but still not perfect because the detection program for someone using glasses does not detect it accurately because the eyeglass lens has light reflecting off so that the program does not run optimally. The solution that can be done is by detecting someone who is wearing glasses by using non-reflective eyeglass lenses so that the program can run properly and according to the program's workflow.
2. The results of detecting a person using glasses via live video in low light conditions get quite good results but are still not perfect because if the detection of glasses is carried out in too dark light conditions, the program cannot detect the face and eyes of the person in the video. So that the detection of glasses can not run optimally. The solution that can be done is that while the program is running, it must be carried out in bright enough light conditions so that the program can run optimally and detect a person's face and eyes to be able to conclude that the person is wearing glasses or not.
3. In this project, the detection of people with glasses can only be done through the front view of the person's face, while from the side and back view of the person's face it cannot be done because the program can only detect the face and eyes of that person. The solution that can be done is by making improvements to the program by detecting the glasses frame

used by the person so that the program can detect the person through the side and back view of the person's face.

4. As long as the program is running in detecting the glasses used by the person, the face and eye objects must not be blocked by other objects so that the program can run optimally, whereas if there are other objects that block the face and eye objects, the program cannot run in performing the task. glasses detection.

Based on the conclusions above, the following are suggestions that can be made for further research:

1. The project of detecting people who use glasses through live video can be done by using glasses that have non-reflective lenses so that the program can detect glasses properly, because if there are lenses that reflect light, the program will not be optimal in detecting the glasses.
2. The project of detecting people who use glasses through live video can be developed by using CCTV cameras or cameras that have higher or better pixel quality, so that in the program the detection of someone who uses glasses can run more accurately and optimally.
3. The development of a project to detect someone who uses glasses can be done by detecting the frame of the glasses worn by that person, so that the program can detect it through the side and back view of the person's face.

REFERENCES

[1] H. Mulyawan, M. Z. H. Samsono, and Setiawardhana, “Identifikasi Dan Tracking Objek Berbasis Image,” Institut Teknologi Sepuluh Nopember, pp. 1–5, 2011, [Online]. Available: http://repo.pens.ac.id/1324/1/Paper_TA_MBAH.pdf.

[2] D. Prasetyo, “Aplikasi Pendeteksi Jerawat Di Wajah Dengan Menggunakan Teknik Pengolahan Citra Pada Foto,” U. Islam Indonesia, “Fakultas Teknologi Industri,” p. 52, 2018. https://dspace.uii.ac.id/bitstream/handle/123456789/10492/Laporan%20Tugas%20Akhir.pdf?sequence=1&isAllowed=y

[3] E. Nuraharjo, Z. Budiarso, D. Irawan, and T. Utomo, “Implementasi Metode Center Plotting Of Pixel Untuk Mendeteksi Warna Citra Bidang Datar 2-D,” U. Stikubank and U. Semarang, “Fakultas teknologi informasi,” pp. 2019–2020, 2019. https://www.unisbank.ac.id/ojs/index.php/fti1/article/view/6088/1953

[4] A. R. Putri, “Pengolahan Citra Dengan Menggunakan Web Cam Pada Kendaraan Bergerak Di Jalan Raya,” STKIP PGRI Tulungagung, *JIPI (Jurnal Ilm. Penelit. dan Pembelajaran Inform.*, vol. 1, no. 01, pp. 1–6, 2016, doi: 10.29100/jipi.v1i01.18. https://jurnal.stkippgritulungagung.ac.id/index.php/jipi/article/download/18/8

[5] I. Nugraha, Suwandi, and H. Bethaningtyas, “Sistem Otomasi Dalam Penyortiran Tomat Dengan Image Processing Menggunakan Metode Deteksi Rgb Automated System in Tomato Sorting With Image Processing Using Rgb Detection Method,” U. Telkom, “Fakultas Teknik Elektro,” vol. 2, no. 3, 2015. https://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/download/559/529

[6] R. B. Pratama, “Penerapan Metode Eigenface Pada Sistem Parkir,” U. Amikom Yogyakarta, *J. DISPROTEK*, vol. 9, no. 2 pp. 86–96, 2018. https://ejournal.unisnu.ac.id/JDPT/article/download/801/1061

[7] M. Hidayat, “Sistem Pengatur Lampu Lalu Lintas Menggunakan Image Processing,” U. Islam Negeri Alaudin Makassar, “Fakultas Sains dan Teknologi,” vol. 9, no. August, p. 10, 2016, [Online]. Available: http://repositori.uin-alauddin.ac.id/9637/1/Muhammad%20Taufiq%20Hidayat.pdf

[8] S. Jatmika and D. Purnamasari, “Rancang Bangun Alat Pendeteksi Kematangan Buah Apel Dengan Menggunakan Metode Image Processing Berdasarkan Komposisi Warna,” STMIK Asia Malang, *J. Ilm. Teknol. Inf. Asia*, vol. 8, no. 1, pp. 51–58, 2014. https://jurnal.stmikasia.ac.id/index.php/jitika/article/view/128/99

[9] D. P. Sari, S. Rasyad, E. Evelina, and A. Amperawan, “Identifikasi Huruf Braille Berbasis Image Processing Secara Real Time, ”Politeknik Negeri Sriwijaya,” *J. Ampere*, vol. 2, no. 2, p. 68, 2017, doi: 10.31851/ampere.v2i2.1765. https://jurnal.umj.ac.id/index.php/semnastek/article/download/1908/1560

[10] H. H. Setiawan, “Klasifikasi Jenis Buah Pisang Dengan Image Processing Menggunakan Methode Classification Of Type Of Banana Fruits With Image Processing Using Backpropagation Method,” *Univ. Sanata Dharma*, 2018. http://repository.usd.ac.id/31630/2/135314126_full.pdf

[11] R. Risfendra, A. Asfinaldi, H. Habibullah, and J. Julisardi, “Sistem Pergerakan Robot Kiper Beroda Menggunakan Metode Wall Follower Berbasis Image Processing,” U. Negeri Padang, “Fakultas Teknik Elektro,” *Elkha*, vol. 12, no. 1, p. 1, 2020, doi: 10.26418/elkha.v12i1.35245. https://jurnal.untan.ac.id/index.php/Elkha/article/download/35245/pdf

[12] A. Liem, N. Sandjaja, and R. Adipranata, “Sistem Pendeteksian Lahan Parkir Kosong pada Static Image dengan Image Processing,” U. Kristen Petra, “Fakultas Teknologi Industri,” vol. 5, no. 2, pp. 2–5, 2017. http://publication.petra.ac.id/index.php/teknik-informatika/article/download/5717/5215

[13] A. Y. -, D. P. -, and A. R. -, “Template Matching dalam Otomatisasi Penghitung Sel Keping Darah Berbasis Image Processing untuk Deteksi Dini Derajat Penyakit Demam Berdarah,” Politeknik Negeri Padang, *J. Ilm. Poli Rekayasa*, vol. 14, no. 1, p. 1, 2018, doi: 10.30630/jipr.14.1.109. https://ejournal2.pnp.ac.id/index.php/jipr/article/view/48/25

[14] H. A. Sholeh, Y. Mulyani, and H. D. Septama, “Sobel a Comparative Study Lampung Script Characters Recogniton Based Edge Detection Method of Roberts,” U. Lampung, “Fakultas Teknik,” vol. 6, no. 3, pp. 261–272, 2018. https://jurnal.balitbangda.lampungprov.go.id/index.php/jip/article/download/94/96

[15] B. Marhaenanto, D. W. Soedibyo, and M. Farid, “Penentuan lama Sangrai Kopi Terhadap Variasi Derajat Sangrai Menggunakan Model Warna Rgb Pada Pengolahan Citra Digital (Digital Image Processing),” U. Jember, “Fakultas Teknologi Pertanian,” *J. Agroteknologi*, vol. 09, no. 02, pp. 1–10, 2015, [Online]. Available: https://jurnal.unej.ac.id/index.php/JAGT/article/download/3536/2746/.

[16] F. A. Saputra, D. E. Sintyaningrum, and I. Kamil, “Monitoring sampah pada drainase berbasis thingspeak dengan metode based on image processing,” U. Negeri Malang, “Fakultas Teknik,” vol. 4, no. 2, pp. 43–50, 2020. http://e-journals.unmul.ac.id/index.php/SAKTI/article/download/1842/pdf

[17] H. Fitri and dan Ivan Finiel Hotmartua Bagariang, "Pemanfaatan Esp32-Cam Untuk Mengukur Ketinggian Air Menggunakan Metode Image Processing," Politeknik Negeri Padang, *Semin. Nas. Terap. Ris. Inov. Ke-6 ISAS Publ. Ser. Eng. Sci.*, vol. 6, no. 1, pp. 762–769, 2020. https://proceeding.isas.or.id/index.php/sentrinov/article/view/547/226

[18] A. Jalil, "Sistem Kontrol Mesin Penukaran Uang Kertas Rupiah Berbasis Pengolahan Citra Dan Raspberry PI," STMIK Handayani Makassar, vol. 14, no. 2, pp. 13–19, 2014, [Online]. Available: http://jurnal.fikom.umi.ac.id/index.php/ILKOM/article/download/272/138

[19] E. Winarno, "Aplikasi Deteksi Tepi pada Realtime Video menggunakan Algoritma Canny Detection," U. Stikubank, *J. Teknol. Inf. Din.*, vol. 16, no. 1, pp. 44–49, 2011. https://unisbank.ac.id/ojs/index.php/fti1/article/view/349/226

[20] T. Y. Prahudaya and A. Harjoko, "Metode Klasifikasi Mutu Jambu Biji Menggunakan Knn Berdasarkan Fitur Warna Dan Tekstur," U. Gajah Mada, "Fakultas Matematika Dan Ilmu Pengetahuan Alam," *J. Teknosains*, vol. 6, no. 2, p. 113, 2017, doi: 10.22146/teknosains.26972.
https://jurnal.ugm.ac.id/teknosains/article/download/26972/17069

[21] K. Abdullah and A. A. Riadi, "Klasifikasi Kelayakan Beras Menggunakan Metode Laplacian of Gaussian," U. Muria Kudus, "Fakultas Teknik," vol. 6, no. 2, pp. 312–320, 2014. https://tunasbangsa.ac.id/ejurnal/index.php/jurasik/article/download/353/332

[22] D. R. H. Putra, F. Marisa, and I. D. Wijaya, "Identifikasi Wajah Berbasis Segmentasi Warna Kulit Wajah Menggunakan Naive Bayes Classifier," U. Widyagama Malang, "Fakultas Teknik," *J. Teknol. Inf.*, vol. 9, no. 2, pp. 99–106, 2018. http://ejurnal.stimata.ac.id/index.php?journal=TI&page=article&op=view&path%5B%5D=329&path%5B%5D=319

[23] O. D. Nurhayati, "Mengenali Jenis Telur Ayam Biasa Dan Telur Ayam Omega-3," *J. Sist. Komput.*, no. Lab.Multimedia, Prodi Sistem Komputer, Fakultas Teknik Universitas Diponegoro, Jl. Prof.H.Soedarto, Tembalang Semarang, Email, vol. 10, pp. 84–93, 2015. https://ejournal.undip.ac.id/index.php/jsinbis/article/download/29127/pdf

[24] S. W. Masdiyasa IGS, S. Bhirawa, "Identifikasi plat nomor kendaraan bermotor menggunakan Metode multi-step image processing berbasis android," U. Narotama, vol. 5, no. 1, pp. 17–25, 2019, [Online]. Available: https://jurnal.narotama.ac.id/index.php/narodroid/article/view/862/556.

[25] R. L. Hasanah, "Identifikasi Jenis Buah 'Pyrus' (Pir) Menggunakan Algoritma Adaptive Neuro Fuzzy Inference System (ANFIS)," STMIK Nusa Mandiri, vol. 9, no. 1, pp. 120–128, 2021. https://ejournal.bsi.ac.id/ejurnal/index.php/evolusi/article/download/10318/4908

[26] F. I. Nasution, Andasuryani, and R. E. Putri, "PENGEMBANGAN METODE IMAGE PROCESSING UNTUK GRADING BUAH SALAK PADANG SIDEMPUAN (Salacca sumatrana) U. Andalas, 2020. http://tpa.fateta.unand.ac.id/index.php/JTPA/article/download/474/205

[27] Z. Effendi, I. O. Y, and M. H. A. Sembiring, "Deteksi Unsur Hara Makro N,P,dan K pada Daun Tanaman Kelapa Sawit dengan Menggunakan Metode Image Processing berdasarkan Filter Sobel," STIPAP Medan, vol. II, no. 1, pp. 42–49, 2018. https://ejurnal.stipap.ac.id/index.php/JAE/article/view/38

[28] M. Y. Yunus, "Deteksi struktur permukaan logam hasil heat treatment dengan image processing dan neural network (sub image processing)," Politeknik Negeri Ujung Pandang, vol. 8, no. 1, pp. 13–21, 2010. http://jurnal.poliupg.ac.id/index.php/Sinergi/article/download/1050/pdf

[29] Z. Effendi, S. Manurung, and S. M. Ayu, “Deteksi Penyakit Garis Kuning (Patch Yellow) Pada Daun Tanaman Kelapa Sawit (Elaeis Guineensis Jacq) Dengan Menggunakan Metode Image Processing Berdasarkan Filter Sobel,” U. Muhammadiyah Tapanuli Selatan, “Fakultas Pertanian,” vol. 5, no. 1, pp. 43–56, 2020, [Online]. Available: https://core.ac.uk/download/pdf/327184963.pdf.

[30] A. F. Amalia and H. Saputro, “Analisis deteksi iris mata menggunakan metode deteksi tepi sobel,” U. Sarjanawiyata Tamansiswa, “Fakultas Keguruan Dan Ilmu Pendidikan,” vol. 4, no. 1, pp. 41–48, 2018. https://jurnal.ustjogja.ac.id/index.php/sciencetech/article/view/2482/1456