

Generate Haar Feature for Face Detection Without Mask and Face with Mask

Sari Feronika Nadapdap and Good Fried Panggabean

Abstract— The COVID-19 pandemic is a problem that worries the wider community. To stop the spread of COVID-19, the mandatory health protocol to use masks is enforced. Many people do not comply with these health protocols. Based on these problems, a technology was developed to monitor the face that uses a mask or not used. This technology uses the Viola-Jones algorithm. In carrying out its detection function, this algorithm requires a classifier which is the result of training on some positive and negative image data sets. In this study, two positive image data sets were used: facial data using masks and facial data not using masks. The classifier obtained from the training process is a cascade file in XML format that will be used in the detection program. In this study, several training processes were carried out to obtain a good comparison value between positive and negative dataset samples in forming a cascade. The cascade test with the highest accuracy value was obtained from the classifier using 1000 positive samples and 2000 negative samples, namely 98.70% for face detection without a mask and 92.63% for face detection using a mask.

Index Terms— Viola Jones, Cascade Classifier, Accuracy, Detection, xml file.

I. INTRODUCTION

THIS world is facing a massive health crisis due to the COVID-19 pandemic. With the rapid transmission of COVID-19, the World Health Organization (WHO) has recommended that various countries ensure their citizens wear masks in public places [1]. Many people do not comply with these health protocols.

Based on these problems, we propose a machine learning model using the Viola-Jones algorithm in this paper. Viola-Jones is an object detection algorithm with a high degree of accuracy. The goal to be achieved in this research is to produce a haar cascade classifier feature for faces that use masks and faces that do not use masks. The Haar feature can then be used in a detection program to detect faces that are not wearing a mask or wearing a mask. To get a good detection accuracy value, some training will also be carried out on several data comparisons.

Sari Feronika Nadapdap. Electrical Engineering Study Program, Del Institute of Technology, Toba, Indonesia. (e-mail: sariferonika14@gmail.com).

Andi Pranata Lumbanraja. Electrical Engineering Study Program, Del Institute of Technology, Toba, Indonesia. (e-mail: andilumbanraja00@gmail.com).

Good Fried Panggabean. Faculty of Informatics and Electrical Engineering, Del Institute of Technology. (e-mail: good@del.ac.id).

There have been many studies conducted using the Viola-Jones Algorithm. Some of them are in a paper [2], explaining how a face detection program that uses a mask by applying the Haar cascade method will get a detection accuracy value of 93.33% [2]. Study in [3] also explains how the face detection system uses the Viola-Jones method in real time, where the detection accuracy is 67.7% [3]. Study in [4] compares the CNN detection method with Viola-Jones. This paper concludes that Viola-Jones has good accuracy and requires less computation [4].

As for what distinguishes this research from some of the studies above, it shows how the stages of the dataset are processed to create the Haar feature. This was not found in the study where they directly used the pre-existing Haar features in their detection program without knowing how to process the existing dataset into a feature that can be used to detect.

Several supporting materials are needed in creating the Haar feature, such as a Linux operating system, which will then install the OpenCV library and Python programming. It also takes datasets from faces that use masks and faces that do not use masks. The explanation of the stages, starting from the collection of data sets to the training process that produces Haar features, will be explained in the next chapter.

II. LITERATURE REVIEW

A. Viola Jones Method

The Viola-Jones method is an object detection method with a fairly high accuracy, about 93.7%. This method was proposed by Paul Viola and Michael Jones in 2001 [5]. The Viola-Jones algorithm combines four main keys, namely Haar Like Feature, Integral Image, Ada-boost Learning and Cascade Classifier.

1. Haar Like Feature

Haar Like Feature is the difference between the number of pixel values in the light area and the number of pixel values in the dark area of each rectangle [6].

$$F(\text{Haar}) = \sum F_{\text{Bright area}} - \sum F_{\text{Dark area}}$$

2. Integral Image

An integral image is a medium that is used to calculate the feature value by converting the input image into an integral image representation. The calculation process can be done using an integral image with just one scan and is fast and accurate [7].

Calculating the value of the integral image on the feature is using the following formula:

$$s(x, y) = i(x, y) + s(x, y - 1) + s(x - 1, y) + s(x - 1, y - 1)$$

$s(x, y)$ = The value of the sum of each pixel
 $i(x, y)$ = Intensity value of the input image pixel value
 $s(x, y)$ = Pixel value on x-axis
 $s(x, y)$ = Pixel value on the y-axis
 $s(x, y)$ = Diagonal pixel value

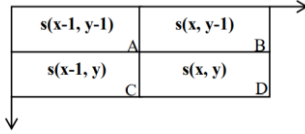


Fig. 1. Integral image calculation

To determine the value of the Haar feature, the following equation is used:

$$i(x', y') = S(A) + S(D) - S(B) - S(C)$$

3. Adaboost

Viola-jones uses a method called Adaboost. Adaboost functions is used to determine whether a feature can represent the presence or absence of a face without a mask or a face that does not use mask in an input image. The Adaboost method combines many weak classifiers to create a strong classifier. The classifier is said to be weak if the filter sequence in the classifier only gets a few correct answers. It would a strong classifier If all the weak classifiers were combined [8].

Here are the classification steps with Ada-boost in detail:

The first step is to train a weak classifier (h = hypothesis) and select the smallest error. The example in Fig. 2 shows only two classes: class + (positive) and class - (negative). First, we will look for the weakest classifier, which is the best in terms of having the smallest error value. It is called a weak classifier because the classifier is only limited by one line resulting in the impossibility of classifying classes correctly [9].

In this process, three weak classifiers will be obtained, and combined into one strong classifier.

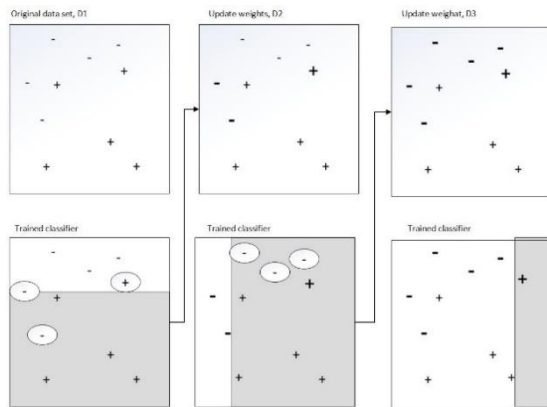


Fig. 2. Classification with adaboost

Then the training data weighting will be carried out

with the following equation [10]:

$$D_{t+1}(i) = \frac{D(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$

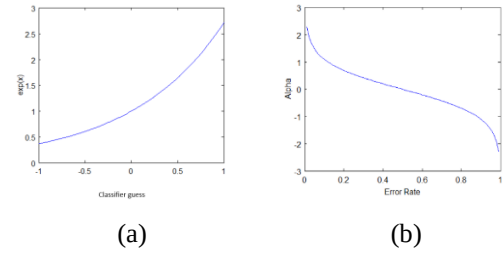


Fig. 3. (a) Training data weight graph, (b) Weight graph for classifier

The description of a graph (a) is as follows:

- If the classifier guess is correct ($y=-1, h=-1$ or $y=1, h=1$), then the value in exp is negative so that the weight result will be < 1 (shrink).
- If the guess is wrong, the value in exp will be positive, the result will be weight > 1 (increase).

Graphic (b) description:

- Weight = 0 when error = 0.5
- Weight is positive when error < 0.5
- Weight is negative when error > 0.5

The next step is to assign weights to these classes. For data that is incorrectly classified (error) in the previous stage, it will be weighed more and the correct one will be weighed less. Giving weights aims to ensure that previously classified data can be classified correctly in the next process of determining the best weak classifier.

Weight formula [10]:

$$\alpha_t = \frac{1}{2} \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$$

In order to combine the three weak classifiers into one strong classifier mathematically, it can be seen as follows [10]:

$$H(x) = \text{sign} \left(\sum_{t=1}^T \alpha_t h_t(x) \right)$$

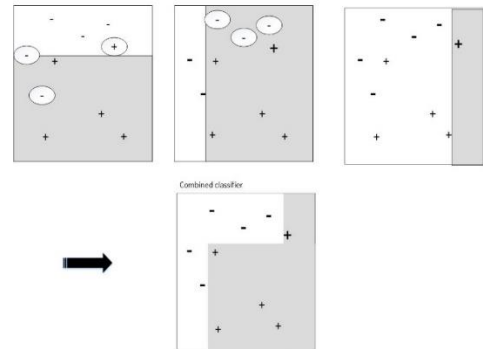


Fig. 4. Strong classifier

Adaboost on the Viola-Jones algorithm has a hypothesis

that is made limited by using a "single feature" which means choosing one feature that best separates faces and non-faces (the error is the smallest).

4. Cascade Classifier



Fig. 5. Cascade classifier Flow

The cascade classifier is a collection of several stages obtained with the Adaboost technique, which is tasked with rejecting the image areas that are not detected by faces. Each stage in the classifier marks the area of the object, whether positive or negative through a sliding window. When it is considered that the face is detected, it is given a positive sign; if it is not detected, it is given a negative sign. The results of this classification are T (True) for images that meet the values in all classifiers and F (False) if they don't [11]. In the groove shape of the cascade classifier is shown.

B. Accuracy

Performance matrices need to be explored to assess the performance of the different classifiers. The most common performance measures to be calculated are Accuracy, Precision, Recall, and F1 Score [12], and they are presented in the equation below [13] [14]:

$$TPR = \frac{TP}{TP + FN}$$

$$PPV = \frac{TP}{TP + FP}$$

$$F_{measure} = 2 \times \left(\frac{TPR \times PPV}{TPR + PPV} \right)$$

$$Accuracy = \frac{TP + TN}{(TP + FP) + (TN + FN)}$$

Explanation:

1. True Positive (TP) = Face object detected correctly as face
2. True Negative (TN) = Non-face objects are detected correctly as non-faces
3. False Positif (FP) = Face object not detected as face
4. False Negative (FN) = Face objects that are not faces are detected as faces
5. True Positive Rate (TPR) = Is the ability of the system to detect correctly
6. Positive Predicted Value (PPV) = Is the total detection accuracy of the total number that should be detected.

III. METHOD

The methodology for generating a cascade used in the face detection process that uses a mask or a face that does not use a mask is shown in Fig. 6 [15].

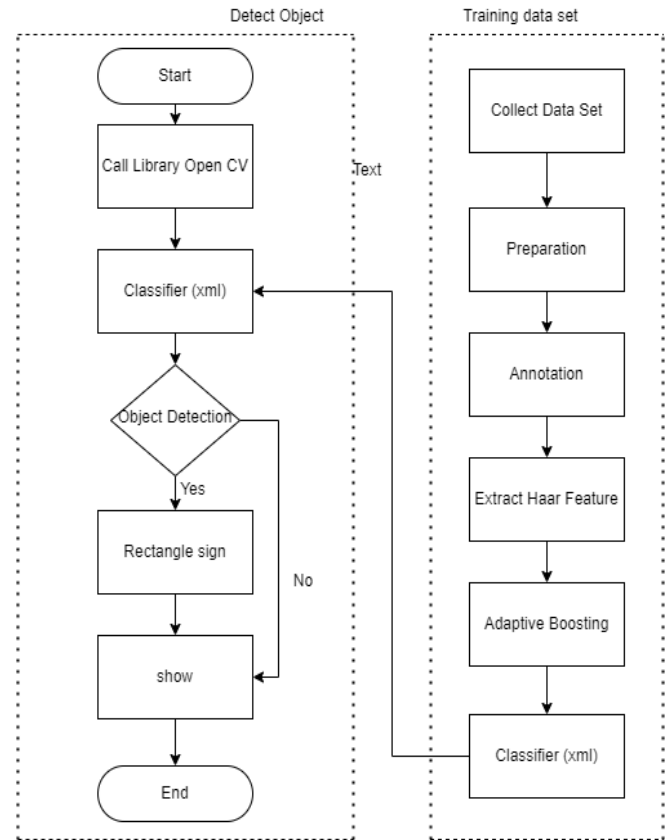


Fig. 6. The Methodology for Constructing Classification Model

The methodology starts with installing the OpenCV python library on a computer terminal. OpenCV is an open-source computer vision system that provides some image processing functions, such as image analysis functions. The use of OpenCV feature has been widely used in many areas, including human-computer interaction, recognition, ego-motion, mobile robotics, object identification, and object detection [16].

The next stage is the creation of a cascade classifier. The cascade classifier is obtained through a training process on some positive and negative data sets. Positive datasets are images in which there are objects to be detected. Negative datasets are images in which there are no objects to be detected. In this research, several data sets were used, namely 1912 facial samples using masks, 3019 negative samples, and 1,763 facial samples not using masks. The training process for this dataset is carried out in several stages, such as preparation, annotation, extracting Haar features, adaptive boosting, and cascading classifier. These stages will be discussed in more detail in the next chapter. The results of the dataset training are a cascade classifier and an xml file format, a template for face object shapes that use masks and faces that do not use masks to be detected.

After the classifier is obtained, it will then be tested in a detection program to determine whether the classifier can detect faces that use masks or faces that do not use masks properly. The classifier detects well if, in the image, faces that use masks or do not use masks are marked with square boxes. On the other hand, the classifier does not properly detect if, in

the image, faces that use masks or do not use masks are not marked with square boxes.

IV. RESULTS AND DISCUSSIONS

A. Dataset Description

In this research, several data sets were used, namely, 1912 facial samples using masks [17], 3019 negative samples, and 1,763 facial samples not using masks [18]. The data set is then divided into several categories that will be used during training and the remaining 35 image samples are used in the classifier testing process. Some examples of data sets are shown in Fig. 7.

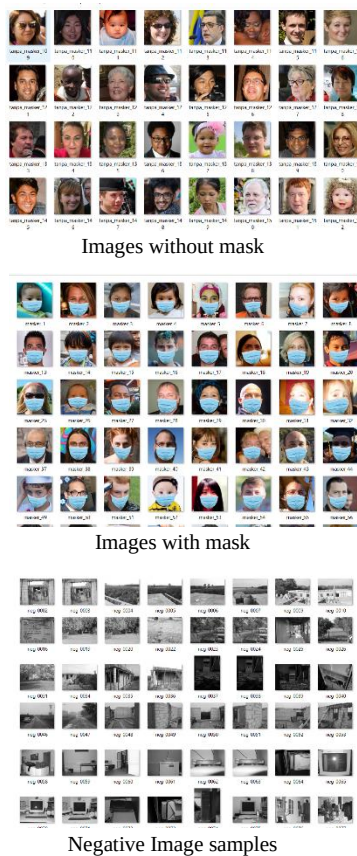


Fig. 7. sample images from the datasets

B. Make a List of Negatives Samples

The first thing to do in making the classifier is to make a list of files with a .jpg extension in the negatives folder with the list's name being bg.txt. Lists can be created using a python script like Fig. 8.

```
nm.py
1 import os
2 import cv2, glob
3 import numpy
4 img=glob.glob('*.jpg')
5 def createlst():
6     for file_type in ['negatives']:
7         for img in os.listdir(file_type):
8             if file_type == 'negatives':
9                 line = file_type+ '/' +img+'\n'
10                with open ('bg.txt', 'a') as f:
11                    f.write(line)
12
13 createlst()
```

Fig. 8. Python script to create sample list

C. Image Annotation

Image annotation is a technique used to be able to run integral images so that the search for areas that want to extract the Haar feature becomes easier and faster. The annotation process is carried out on positive samples by marking the coordinates of the object to be detected in the image [15]. Fig. 9 shows the commands performed in executing the annotation process.

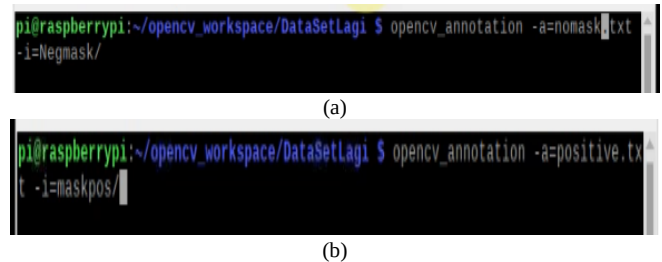


Fig. 9. Commands run the annotation process (a) without a mask, (b) with a mask

The explanation of the command is to annotate images in the Negmask folder and the resulting maskpos will be created with the names nomask.txt and positive.txt. Opencv_annotation is a tool that marks the coordinates of the object detected in an image to create a classifier. Fig. 10 shows the process carried out in running annotations.

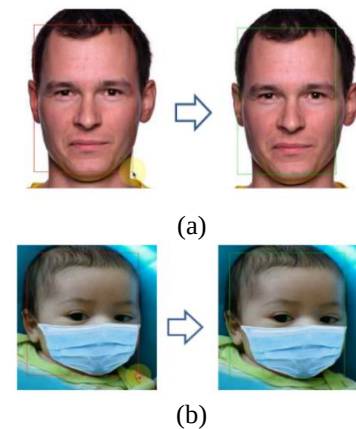


Fig. 10. Positive sample annotation process (a) without mask, (b) with mask

By using the mouse, mark the area that contains the object to be detected so that a box with a red line will appear. If the marking position is correct, use the C (confirm) button to confirm that the area is correct so that the box with the red line will turn into a box with the green line. If the confirmed area is incorrect use the D (delete) key to delete it. If it is correct and you want to move to the next picture, use the N (next) button. This process is carried out to be able to run an integral image so that the process of finding the area that you want to extract the Haar feature becomes easier and faster because the coordinates have been determined.

D. Create Vec File

Annotated files are needed to create a vec file. The vec file is the result of extracting the Haar features into positive samples to be used in making classifiers.


```
pi@raspberrypi:~/opencv_workspace/DataSetLagi $ opencv_createsamples -info positive.txt -num 1000 -w 24 -h 24 -vec pos1.vec
```

Fig. 11. Command to create vector file

The command explains that a vec file with the name pos1.vec will be created using 1000 samples marked in the positive.txt file with a sliding window size of 24X24 pixels to extract the Haar feature. Opencv_createsamples is a tool to extract the Haar features into positive samples so they can be used later in making classifiers.

E. Create Classifier

After the vec file has been successfully obtained, the classifier can be created with the opencv_traincascade command, a tool for creating classifiers. The command used is as follows:

```
pi@raspberrypi:~/opencv_workspace/DataSetLagi $ opencv_traincascade -data classifiermask2 -vec pos1.vec -bg bg.txt -numPos 500 -numNeg 1000 -numStages 15 -w 24 -h 24
```

Fig. 12. Command to create classifier

The command explains that a classifier will be created from pos1.vec for positive samples and bg.txt for negative samples. The number of positive samples used is 500 and the number of negative samples is 1000, and it is expected to make 15 stages of the cascade classifier. The training process results will be stored in a folder with the name classifiermask2. This training process usually takes a long time depending on the number of samples that want to be detrained. After the training process is complete, a file with the extension .xml is generated, which contains a classifier that has been successfully created.

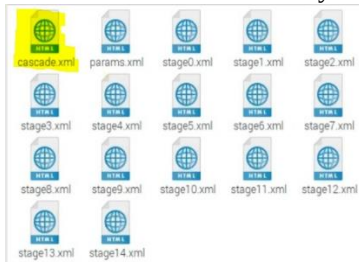


Fig. 13. Classifier that has been successfully created

F. Detection Result

In this research, the classifier obtained from the training process was tested against 35 random images outside the dataset. The following are the results of tests conducted by the author.

TABLE I
COMPARISON NUMBER OF POSITIVE AND NEGATIVE SAMPLES

Object detected	Number of samples		True Detect	False Detect		Accuracy (%)
	Positive (P)	Negative (N)		N	P	
Face without mask	500	250	39	96	1	44.57
	500	500	37	33	1	68.52
	500	1000	36	6	3	88.89
	1000	500	39	84	1	47.85

Face with mask	1000	1000	37	24	1	74.75
	1000	2000	38	1	0	98.70
	500	250	42	48	8	62.22
	500	500	35	15	14	70.71
	500	1000	36	14	10	75.00
	1000	500	40	22	10	71.43
	1000	1000	37	10	11	77.89
	1000	2000	44	7	4	92.63

Table 1 above shows that the number and comparison between positive and negative samples affect the accuracy value to be obtained. The number of negative samples less than the number of positive samples will result in a lower accuracy value in detecting faces than if the number of negative samples is more than the number of positive samples.

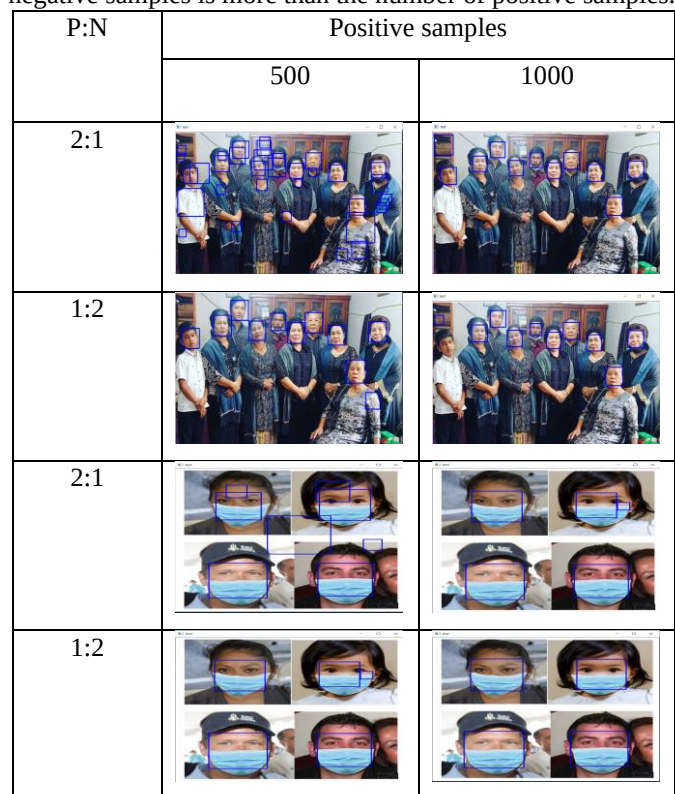


Fig. 14. Detection result

Fig. 14 above proves that the comparison of positive and negative samples has an effect on accuracy or accuracy in detecting objects. This can be seen from the large number of detection errors that occur when the number of negative samples is less than the number of positive samples. The detection error is marked by many blue boxes that are not correctly detected on the desired object, namely faces that use masks or do not use masks.

V. CONCLUSION AND SUGGESTION

In this research, the Viola-Jones algorithm was used to perform the function of detecting faces that use masks and faces that do not use masks. Viola-Jones requires a classifier

from training on some positive and negative image datasets. Classifiers can be created by extracting the Haar features into a data set and proceeding to the training process to generate a cascade in .xml file format. The xml file will be used in the detection program. Several training processes are carried out based on comparing the amount of positive and negative image data provided to determine the effect of the number of data sets used on the accuracy value. The highest accuracy value was obtained from the classifier that used 1000 positive samples and 2000 negative samples, namely 98.70% for face detection without a mask and 92.63% for faces using a mask. These results indicate that a good accuracy value is obtained when the number of negative samples used is greater than the number of positive samples.

REFERENCES

- [1] J. T. Atmojo, A. P. Putri, S. Kuntari, R. T. Handayani and A. T. Darmayanti, "The Use of Masks in the Prevention and Management of COVID-19: Rationality, Effectiveness and Current Issues," *Journal of Health Research*, vol. 3, 2020.
- [2] F. L. Ahmad, A. Nugroho and A. F. Suni, "Deteksi Pemakai Masker Menggunakan Metode Haar Cascade Sebagai Pencegahan COVID 19," *Edu Elekrika Journal*, vol. 10, 2021.
- [3] R. E. Putri, T. Matulatan and N. Hayaty, "Sistem Deteksi Wajah Pada Kamera Real-Time dengan Menggunakan Metode Viola-Jones," *Jurnal Sustainable: Jurnal Hasil Penelitian dan Industri Terapan*, vol. 08, 2019.
- [4] M. Sivakumar, N. Saranprasath, N. S. Sridharan and V. S. Praveen, "Comparative analysis of CNN and Viola-Jones for Face Mask Detection," *Journal of Physics*, 2021.
- [5] P. A. P. E and S. D, "Deteksi Wajah dengan Berbagai Posisi Sudut Pada Sekumpulan Orang dengan Membandingkan Metode Viola Jones dan Kanade-Lucas-Tomas," vol. 5, p. 136, 2016.
- [6] P. Viola and M. Jones, "Robust Real-Time Face Detection," *International Journal of Computer Vision*, pp. 137-154, 2004.
- [7] Gupta, "Face detection and recognition using Raspberry Pi," pp. 1-5, 2017.
- [8] Jusia, "Face Recognition Menggunakan Metode Algoritma Viola Jones Dalam Penerapan Computer Vision," vol. 11, pp. 635-675, 2019.
- [9] A. Umam, Director, *Detail Cara Kerja Viola Jones untuk Deteksi Wajah (Deteksi Objek)*. [Film]. Indonesia.2019.
- [10] N. Zerrouki and F. Harrou, "Vision-based Human Action Classification Using Adaptive Boosting Algorithm," *IEEE SENSORS JOURNAL*, p. 4, 2018.
- [11] F. L. Ahmad, A. Nugroho and A. F. Suni, "Deteksi Pemakai Masker Menggunakan Metode Haar Cascade Sebagai Pencegahan COVID 19," *Edu Elekrika Journal*, vol. 10, 2021.
- [12] G. Cyril and G. Eric, "A Probabilistic Interpretation of Precision, Recall and F-Score, with Implication for Evaluation," in *Advances in Information Retrieval, 27th European Conference on IR Research, ECIR 2005, Santiago de Compostela, Spain, March 21-23, 2005, Proceedings*, 2005.
- [13] M. F. Naufal and S. F. Kusuma, "Pendeteksi Citra Masker Wajah Menggunakan Cnn Dan Transfer Learning," *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK)*, 2021.
- [14] S. o. C. Science, "Evaluation : From Precision Recall And F-Measure To Roc Informedness , Markedness & Correlation," vol. 2, pp. 37-63, 2011.
- [15] R. Lumbantobing, "Implementation Of Viola Jones Algorithm On Raspberry Pi 4 To Detect Car," pp. 22-23, 2020.
- [16] K. D. Irianto, G. Ariyanto and D. Ary, "Motion Detection Using Opencv With Background Subtraction And Frame Differencing Technique," *Simposium Nasional RAPI XVIII 2019*, p. 75, 2019.
- [17] P. Bhandary, "Github," 2020. [Online]. Available: https://github.com/prajnasb/observations/tree/master/experiments/data/with_mask. [Accessed March 2021].
- [18] P. Bhandary, "Github," 2020. [Online]. Available: https://github.com/prajnasb/observations/tree/master/experiments/data/without_mask. [Accessed March 2021].