Journal of Computer Networks, Architecture and High Performance Computing
Volume 6, Number 1, January 2024
https://doi.org/10.47709/cnahpc.v6i1.3460

Submitted : January 16, 2024
Accepted : January 21, 2024
Published : January 22, 2024

Integration Of Open CV LBF Model To Detect Masks In Health Protocol Surveillance Systems

Yovi Litanianda[1], Moh. Bhanu Setyawan[2], Adi Fajaryanto C[3], Ismail Abdurrozzaq Z[4], Charisma Wahyu Aditya[5]
[1)2)3)4)5)]Informatics Engineering Study Program, Faculty of Engineering, Muhammadiyah University of Ponorogo
[1)]yovi@umpo.ac.id, [2)]m.banu@umpo.ac.id, [3)]adifajaryanto@umpo.ac.id, [4)]ismail@umpo.ac.id,
[5)]bl4ck.cmd@gmail.com

ABSTRACT

The Corona Viruses Diseases pandemic that was rife in early 2020 and hit many countries caused discipline to be applied to health protocols. The prevention of physical contact between humans gave rise to new traditions in aspects of human life. Almost all public facilities in Indonesia require visitors to wear masks as a means of preventing exposure to viruses in the air. However, this advice is often ignored by some people. In addition to endangering many people, this condition also makes public facility managers need extra resources in the form of time, energy and costs to ensure this health protocol is implemented. The existence of these problems triggers the emergence of innovations to present a system that provides assurance and convenience in ensuring compliance with health protocols for the use of masks through creative and effective methods. This method is done by utilizing CCTV cameras or webcams at the entrance equipped with an Artificial Intelligence program designed to be able to detect the use of masks on visitors to public facilities, and without the need for other sensors. The detection system is built on the concept of facial biometrics and utilizes the OpenCV LBF model to detector landmarks on a person's face. Based on tests conducted through several scenario, it can be said that the open CV LBF model successfully identified the use of masks within 35 seconds, increasing the reading distance to 2 meters making the process longer. In addition, in indoor lighting conditions, the system experienced 1 detection error with a process time of 18 seconds, while for well-light outdoor conditions the system managed to detect all objects within 10 seconds

Keywords: Artificial Intelligence, Landmark Detector, Mask Detector, Opencv, LBF Model

1. INTRODUCTION

Corona virus Disease (COVID19) originated in Wuhan, China, which was confirmed to appear in December 2019. This outbreak then massively spread to many countries including Indonesia. This pandemic has had an impact that has greatly affected many aspects of human life. Fears of contagion made the Industry and Government implement a temporary suspension of activities for several days. However, along with pressure due to vital losses on the economic side, as well as public services, there was a relaxation of rules, namely restrictions on activities. This policy was taken by the central government and followed by local governments with the aim not only to keep the economy running but also as an effort to reduce the number of crowds, so as to reduce the transmission of the Covid19 outbreak.(Ongky Heri et al., 2021)

According to research conducted by Peeri NC, et al, the way of transmission of this disease is mainly through small droplets that often come out when someone coughs or sneezes. Viruses that have come out of the body can survive in the air or objects for various periods of time. Objects affected by droplets if then touched by hands will cause the transfer of the virus from one human to another. This is what causes touch to be declared as one of the ways the Covid19 Virus spreads. In addition to spreading through objects, the presence of the virus in the air around the patient is also very likely to be inhaled and spread to other people. Therefore, in order to reduce the spread of the virus, the government through the Ministry of Health issued recommendations for the public to comply with the Health Protocol, namely by means of everyone must maintain a safe distance between individuals, use masks when doing activities not alone and wash hands regularly after using or coming into contact with equipment in public facilities.(Peeri et al., 2021)

The implementation of the rule has not been fully obeyed by the community. Some people are still stubborn and do not want to wear masks during activities. Of course, this action should not be tolerated, so the management of

* Corresponding author

Submitted : January 16, 2024
Accepted : January 21, 2024
Published : January 22, 2024

public facility objects is forced to provide officers to supervise visitors. This action, although it can be successful, in fact makes the manager have to spend additional resources in the form of duty officers. This policy, of course, has an impact on additional expenses and also adds to the costs that must be incurred and time losses because there can be errors in the field.

The possibility of loss can be overcome by utilizing surveillance camera technology or commonly called CCTV (Close Circuit Television). Currently, almost all public facilities are equipped with CCTV cameras, especially at the entrance. The existence of this device is very likely to be engineered so that it is able to take on the duties of duty officers, especially if the ministry has been integrated with attendance monitoring devices. Such engineering can be done by manipulating images and videos, so that both can be recognized by the system. Engineering that can be done is by modifying the face identification feature into a mask identification implanted in the device. (Cobantoro, 2018; Cobantoro et al., 2023; Masykur et al., 2020; Oktavianto et al., n.d.; Setyawan et al., 2020)

Modification of facial identification features is done by adding face mark or landmark features. This feature is built using the LBF (Local Binary Features) model from Open CV (Computer Vision). This model works by detecting unique facial characteristics based on facial contour points obtained from the data train in the LBF model. Based on that assumption, a concept can be made if all facial features are detected by the camera, meaning that the face is fully exposed, indicating that the object is not wearing a mask. It is hoped that the creation of this Mask Detection System will be able to help offices and public facilities to monitor the use of masks easily and documented.

Of course, this concept cannot be separated from the shortcomings that may occur. Therefore, in this research it will be studied how possible for this concept to be applied. This research was limited to a laboratory scale with limited equipment and a test design that was far from perfect. This study will also not discuss the manufacture of face identification or face detection in detail and more focused on how the performance of the use of LBF models in mask detection.

2. LITERATURE REVIEW

Face Detection

Face Detection performs the process of detecting the location of faces, and distinguishing objects in the form of faces with other objects on the visual screen. Face Detection can be built using the Haar cascade Algorithm which works according to the following parameters. ***Radius*** is a parameter for creating a circular pattern from the central pixel. ***X grid*** or the number of cells calculated on the horizontal axis on the screen. ***Grid Y*** or the number of cells counted on the vertical axis on the screen. ***Neighbor*** is the number of sample points that match the radius pattern. The point is near the radius circle to detect the movement of the face as the central pixel. The Haar algorithm utilizes statistical methods in performing face detection with reference (13)*haarlike fetures using* fixed-sized images. Computation on this algorithm creates a sample image that depicts the real face based on its characteristics, then shifts the pixel window based on radius and neighbor patterns followed by converting the picture into grayscale. The last stage is taken a radius point measuring 3 x 3 pixels which in each pixel contains a matrix intensity of 0-225. Next, the process will take the center value of that matrix as the threshold for determining the Neighbor point.(Al Hajri et al., 2019; Alam, Saleh, Juliana, et al., 2020)

Face Recognition.

Unlike Face detection, *Face Recognition* is a process or procedure to match three components, namely, Train Data in the form of Grayscale Picture, Histogram Data and User Database.

* Corresponding author

Journal of Computer Networks, Architecture and High Performance Computing
Volume 6, Number 1, January 2024
https://doi.org/10.47709/cnahpc.v6i1.3460

Submitted : January 16, 2024
Accepted : January 21, 2024
Published : January 22, 2024

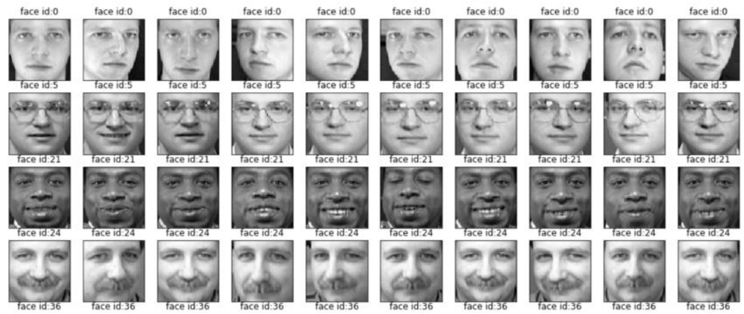

Figure 1. Data Train example.

Face mark or Landmark.

The human face is known to have distinctive characteristics that become unique. This uniqueness can be used for comparison of one data with another data in a database. Data comparison can be characterized more specifically in a computational process with results in the form of spot points on the intended face path. The contour point of the face is called Landmark. The results of Hutamaputra's research found the result in facial recognition accuracy rates of 76.47%, 64.71%, and 47.06% and computing times of 0.5016, 0.1322, and 0.1298 seconds with values of k = 5, 7 and 9.(Hutamaputra, W., & Utaminingrum, 2021)

OPENCV and JAVACV

OpenCV (Computer Vision) is a module or more precisely called a library that helps programs in creating, processing, modifying and converting a digital image so that it can be used for many things, especially for facial recognition. While Java CV is specifically a *library* similar to OpenCV, but specifically used for Computer Vision projects as digital image processing and analysis. This library is a java program that is made capable of being used to access internal and external cameras then process digital image data such as (Bytedeco, 2021) Image Capture, *Object Tracking*, Face Recognition and Recognition of other objects.

LBF Model OpenCV.

LBF (Local Binary Features) Model, is a ready-made data train. LBF Model as a trained face mark model was developed by Computer Vision. This trained data can be downloaded on the Github repository site with the official contributor account of OpenCV via the Github Repository. Shaoqing Ren et al, in their scientific journal explained about Local Binary Features that are able to detect many facial landmarks in one process. Such systems are considered the most efficient for finding the uniqueness of each detected object(Mallick, 2017)(Mallick, 2018)(Ren et al., 2014)

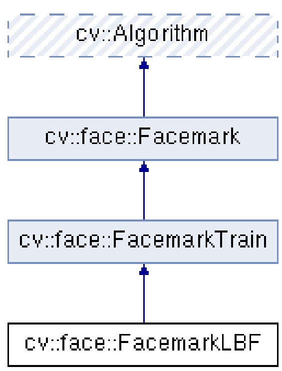

Figure 2. LBFModel Trained Data Diagram.

In this study, the ability of facial landmarks detection was used to detect the use of masks. The working idea can be described as follows:

- Let the landmark detector program with LBF Model iterate to look for facial contours from a visual streaming display.

* Corresponding author

Submitted : January 16, 2024
Accepted : January 21, 2024
Published : January 22, 2024

- LBF Model will look for the contours of the mouth, nose, cheek lines, eyes, eyebrows and hair borders.
- To code the precision command, if all the contours of the face in the previous point are visible, then the face does not use a mask. But if one of the contours of the face is not visible, it means that the face is using a mask or other face covering.

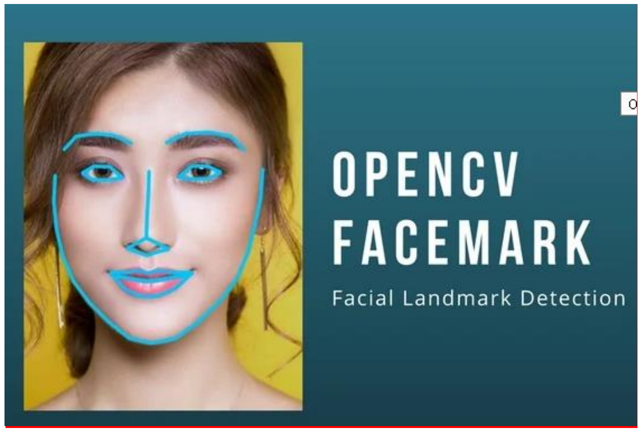

Figure 3. Example of Face Contours with Landmarks.

Based on the work ideas mentioned, it seems that it can contribute future ideas for a more practical and accurate mask detection system. By combining this method and a face detection system, it will allow for a diversity of functions and benefits to emerge within the application.

3. METHOD

This research was conducted following the flow shown in figure 2 below. Starting with prototyping existing system, which is the process of duplicating a monitoring system using CCTV that has face detection based on the Haarcascade algorithm. A system face detection system is then designed for mask detection modules. The mask detection process that was designed, basically carried out using a data train that is completely different from the facial presence module because it implements the Landmark detection system using LBF Model from OpenCV. By using LBF Model, the application will search the contours of the face automatically and provide marks or marks on the faces found.(OpenCV, n.d.)

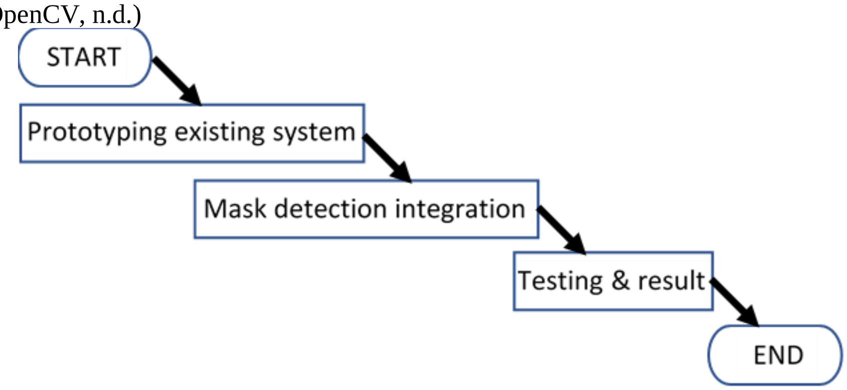

Figure 4. Research flow

The mask detection concept works by letting the landmark detector program with LBF Model iterate to look for facial contours from a visual streaming display. LBF Model will look for the contours of the mouth, nose, cheek lines, eyes, eyebrows and hair borders. This information becomes a precision command code, if all the contours of the face in the previous point are visible, then the face does not use a mask. But if one of the contours of the face is not visible, it means that the face is using a mask or other face covering. The implementation stage of mask detection

* Corresponding author

Journal of Computer Networks, Architecture and High Performance Computing
Volume 6, Number 1, January 2024
https://doi.org/10.47709/cnahpc.v6i1.3460

Submitted : January 16, 2024
Accepted : January 21, 2024
Published : January 22, 2024

begins with designing a display as shown in figure 5 below.

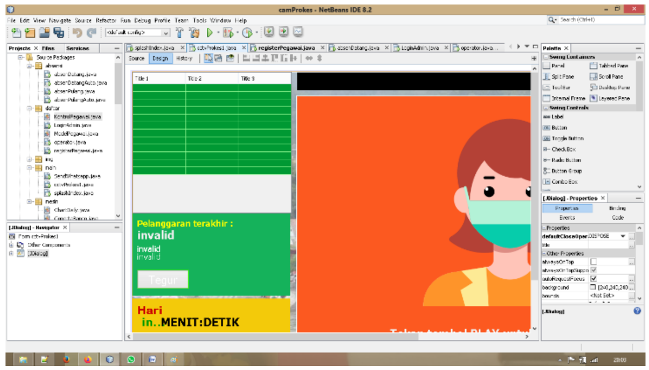

Figure 5. Implementation of CCTV Health Care Design

From the design above, a Pseudocode for Landmark Detection and Face Recognition was then made which was then translated into syntax as below

```
package nama_package;
import library;
import mesin_koneksi;
The other mesin_koneksi*
syntax database connection;
The other mesin_koneksi*
public class nama_JDialog {
variable_algoritma_haarcascade = new nama_variabel
("lokasi/haarcascade_frontalface_alt.xml");
public nama_jdialog(nilai) {
nama_variable.loadModel("lokasi/lbfmodel.yaml");
nama_variable.read("lokasi\\classifierLBPH.yml");
}
public void komponen_jDialog() {}
private void komponen_jDialog() {}
PLAY CAMERA SURVEILLANCE
private void tombol_start() {sintax start kamera pengawas}
    public void nama_sintax() {
        new Thread() {
            sintax
        }.start();
    }

MASK DETECTION PROCESS AND LANDMARKS
if (! WAJAH_TIDAK_KOSONG()) {
    if(FULFILLED) {
            for {
SHOW FACE LANDMARKS WHEN FACE CONTOUR IS DETECTED
            }}}
FACE ID PREDICTION AND FACE MATCHING PROCESS
if (!! WAJAH_TIDAK_KOSONG()) {VARIABLE VALUE}
            for {
            NILAI_VARIABEL
```

* Corresponding author

Journal of Computer Networks, Architecture and High Performance Computing
Volume 6, Number 1, January 2024
https://doi.org/10.47709/cnahpc.v6i1.3460

Submitted : January 16, 2024
Accepted : January 21, 2024
Published : January 22, 2024

```
        if (CONDITION) {
            VALUE WHEN DATA TRAIN NOT FOUND
                  SHOW RECTANGLE
                  SHOW UNKNOWN TEXT
            }else {
                  VALUE WHEN DATA TRAIN IS FOUND
                  SHOW RECTANGLE
                  SHOW NAMA_ID
                  }
}}
```

Important variable declaration:

```
Sample
Facemark facemark = FacemarkLBF.create();
CascadeClassifier cascade = new
CascadeClassifier("xml/haarcascade_frontalface_alt.xml");
LBPHFaceRecognizer recognizer = LBPHFaceRecognizer.create();
public cctvProkes1(java.awt.Frame parent, boolean modal) {
facemark.loadModel("xml/lbfmodel.yaml");
recognizer.read("D:\\XAMPP\\htdocs\\cctv\\dataset\\photos\\classifierLBPH.yml
");
recognizer.setThreshold(213);
}
Sample
```

Face ID Detection

```
for (int i = 0; i < detectedFace.size(); i++) {
Sample
    if (prediction == -1) {
    putText(cameraImage, "Visitor", new Point(x, y), FONT_HERSHEY_TRIPLEX, 1,
new Scalar(0, 255, 0, 3));
    idPerson = 0;
    }else {
    putText(cameraImage, firstNamePerson, new Point(x, y),
FONT_HERSHEY_TRIPLEX, 1, Scalar.YELLOW);
    idPerson = prediction;
    rec();
    inputAttendance();
    }
}
Sample
```

4. RESULT

System testing in this research was carried out with several test parameters, namely: Object distance, lighting (indoor/outdoor), the number of landmarks and recognition that was successful detected and the time needed to detect. CCTV Surveillance Mask testing on this was carried out by utilizing the Netbeans 8.2 IDE as a developer tool to run Java code and LBF Model data trains. The involvement of the data train is done by adding variable declarations to the Java program code as below.

* Corresponding author

Journal of Computer Networks, Architecture and High Performance Computing
Volume 6, Number 1, January 2024
https://doi.org/10.47709/cnahpc.v6i1.3460

Submitted : January 16, 2024
Accepted : January 21, 2024
Published : January 22, 2024

```
Sample
Facemark facemark = FacemarkLBF.create();
CascadeClassifier cascade = new
CascadeClassifier("xml/haarcascade_frontalface_alt.xml");
LBPHFaceRecognizer recognizer = LBPHFaceRecognizer.create();
public cctvProkes1(java.awt.Frame parent, boolean modal) {
facemark.loadModel("xml/lbfmodel.yaml");
recognizer.read("D:\\XAMPP\\htdocs\\cctv\\dataset\\photos\\classif
ierLBPH.yml");
recognizer.setThreshold(213);
}
//Sample
```

Testing of face distance parameters

In this test, the accuracy of the LBF Model will be tested as a Landmark Detector on several objects with different distances. Previously, it was necessary to add the following code so that the log output could be obtained.

```
putText(cameraImage, "ALARM MASKER", new Point(30,
40), FONT_HERSHEY_TRIPLEX, 1, Scalar.YELLOW);
System.out.println("Detect Alarm Masker");
}
```

The code above is added to the Public Void Landmark Detector section, while the Public Void Face Detector section needs to be added the following code:

```
idPerson = prediction;
System.out.println("Detect " + firstNamePerson);
rec();
inputAbsensi ();
}
```

Distance parameter testing was carried out on a day with sufficient lighting conditions without additional lighting or direct sunlight. The results of CCTV Mask Surveillance testing with distance parameters obtained the accuracy of Face Contour detection (Facial Landmark) and the accuracy of Face Recognition (Face Recognition) based on the distance that can be reached by the camera shown in table 1 below.

Table 1. Tests with Distance Parameters

No	Distance (Meter)	Iteration	Detection	Time	Information
1	1 Meter	26	Landmark: 26 Recognition: 25	35 Second	• Landmark Detector works accurately without any errors • Recognition, there is 1 error.

* Corresponding author

Journal of Computer Networks, Architecture and High Performance Computing
Volume 6, Number 1, January 2024
https://doi.org/10.47709/cnahpc.v6i1.3460

Submitted : January 16, 2024
Accepted : January 21, 2024
Published : January 22, 2024

```
run:
Detect Alarm Masker
Detect null
Detect Alarm Masker
Detect Charisma
Detect Alarm Masker
Detect Charisma
Detect Alarm Masker
Detect Charisma
Detect Alarm Masker
Detect Charisma
Detect Alarm Masker
Detect Charisma
BUILD STOPPED (total time: 35 seconds)
```

2	2 Meters	19	Landmark : 19 Recognition : 18	55 Second	• Landmark detector gets error-free accuracy. • Recognition, there is 1 error with a longer detection time.

```
run:
Detect Alarm Masker
Detect null
Detect Alarm Masker
Detect Charisma
Detect Alarm Masker
Detect Charisma
Detect Alarm Masker
Detect Charisma
Detect Alarm Masker
Detect Charisma
Detect Alarm Masker
Detect Alarm Masker
Detect Charisma
Detect Charisma
Detect Alarm Masker
Detect Charisma
Detect Alarm Masker
Detect Charisma
Detect Alarm Masker
Detect Charisma
Detect Alarm Masker
Detect Charisma
Detect Alarm Masker
Detect Charisma
BUILD STOPPED (total time: 55 seconds)
```

3	3 Meters	N/A	N/A	20 seconds	• At this distance it does not manage to detect faces.

```
run:
loading data from : xml/lbfmodel.yaml
[ WARN:1] terminating async callback
BUILD STOPPED (total time: 20 seconds)
```

Testing with Light Parameters (Indoor / Outdoor)

In this test, still using the same method, only two parameters were carried out, namely indoor and outdoor. This has the aim of knowing the detection accuracy data related to the level of illumination. The test is

* Corresponding author

Journal of Computer Networks, Architecture and High Performance Computing
Volume 6, Number 1, January 2024
https://doi.org/10.47709/cnahpc.v6i1.3460

Submitted : January 16, 2024
Accepted : January 21, 2024
Published : January 22, 2024

carried out at a camera distance of 50 cm to the object.

Table 2. Tests with Light Intensity parameters

No	Light	Iteration	Detection	Time	Information
1	Indoor	15	Landmark : 15 Recognition : 14	18 Second	Landmark Detector gets accuracy without errors. Recognition getting 1 error. It's describe in fig. 7

```
run:
Detect Alarm Masker
Detect null
Detect Alarm Masker
Detect Charisma
Detect Alarm Masker
Detect Charisma
Detect Alarm Masker
Detect Charisma
Detect Alarm Masker
Detect Charisma
Detect Alarm Masker
Detect Charisma
Detect Alarm Masker
Detect Charisma
Detect Alarm Masker
Detect Charisma
Detect Alarm Masker
Detect Charisma
Detect Alarm Masker
Detect Charisma
Detect Alarm Masker
Detect Charisma
loading data from : xml/lbfmodel.yaml
[ WARN:1] terminating async callback
BUILD STOPPED (total time: 18 seconds)
```

No	Light	Iteration	Detection	Time	Information
2	Outdoor	9	Landmark : 9 Recognition : 9	10 Second	All detections succeed accurately

```
run:
Detect Alarm Masker
Detect Charisma
Detect Alarm Masker
Detect Charisma
Detect Alarm Masker
Detect Charisma
Detect Alarm Masker
Detect Charisma
Detect Alarm Masker
Detect Charisma
Detect Alarm Masker
Detect Charisma
Detect Alarm Masker
Detect Charisma
Detect Alarm Masker
Detect Charisma
Detect Alarm Masker
Detect Charisma
loading data from : xml/lbfmodel.yaml
[ WARN:1] terminating async callback
BUILD STOPPED (total time: 10 seconds)
```

The number of iterations per second that can be achieved in indoor testing is less when compared to the number of iterations per second performed in outdoor testing. The data can be interpreted as that light intensity affects the accuracy and speed of detection from face detection systems with LBPH

* Corresponding author

Journal of Computer Networks, Architecture and High Performance Computing
Volume 6, Number 1, January 2024
https://doi.org/10.47709/cnahpc.v6i1.3460

Submitted : January 16, 2024
Accepted : January 21, 2024
Published : January 22, 2024

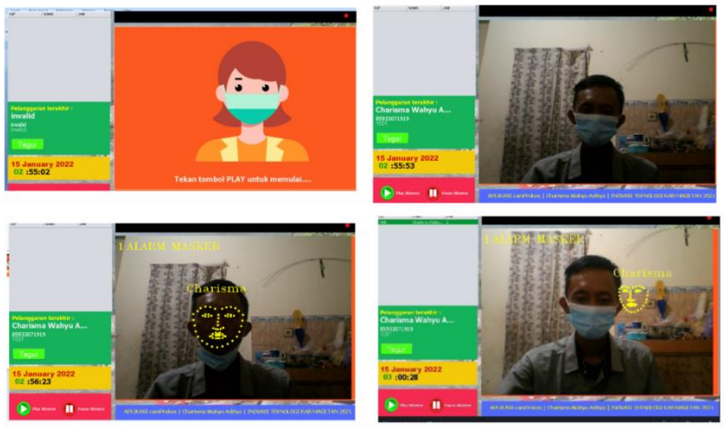

Figure 7: The mask detector has a detection error

5. DISCUSSIONS

Information based on the results of the test above is at a distance of 1 meter the camera managed to detect Face Landmarks even though there was 1 detection error. At a distance of 2 meters, the camera can detect Landmarks and Face ID but the Face Recognition section takes longer with the result that there is also 1 detection error. At a distance of 3 meters the camera cannot detect faces. From these data it can be concluded that, the distance between the object and the camera affects the speed and accuracy of detection. Tests on Indoor Light found detection errors in the Face Recognition system, while tests on Outdoor Light were accurate without any errors. From these results, it can be concluded that Light Intensity affects the accuracy of Face Detection and Face Recognition systems.

6. CONCLUSION

Based on the test data, it can be concluded that the use of OpenCV Model LBF for Mask Detection in the Health protocol surveillance system has been successfully applied, proven by being able to identify the use of masks within 35 seconds at a distance of 1 meter. For a read distance of 2 meters, the detection capability decreases with an indication of the time required to 55 seconds and the system fails to detect at an object distance of 3 meters. In addition, factor lighting also affects the detection ability of the system, in sufficient light conditions (outdoor) the system manages to detect 100% of objects within 10 seconds. While in the condition (indoor) the system takes 18 seconds with the result of 1 failed to detect.

7. REFERENCES

Al Hajri, E., Hafeez, F., & Ameer Azhar, N. V. (2019). Fully automated classroom attendance system. *International Journal of Interactive Mobile Technologies*, *13*(8), 95–106. https://doi.org/10.3991/ijim.v13i08.10100

Alam, H., Saleh, K., Juliana, K. B., & Juliana BRPane, K. (2020). *Pengenalan Wajah Secara Real-Time Menggunakan Algoritma LBPH Pada Raspiberry PI*. https://doi.org/10.30596/rele.v1i1.4471

Alam, H., Saleh, K., & Pane, K. J. B. (2020). Pengenalan Wajah Secara Real-Time Menggunakan Algoritma LBPH Pada Raspiberry PI. *RELE (Rekayasa Elektrikal Dan Energi) : Jurnal Teknik Elektro*, *2*(2), 95–100. https://doi.org/10.30596/rele.v2i2.4471

Bytedeco. (2021). *JavaCV*. Bytedeco. https://github.com/bytedeco/javacv

Cobantoro, A. F. (2018). Analisa Qos (Quality Of Service) Pada Jaringan Rt-Rw Net Dengan Kendali Raspberry Pi. *Network Engineering Research Operation*, *4*(1), 31–36. https://doi.org/10.21107/nero.v4i1.109

Cobantoro, A. F., Masykur, F., & Sussolaikah, K. (2023). Performance Analysis Of Alexnet Convolutional Neural Network (Cnn) Architecture With Image Objects Of Rice Plant Leaves. *JITK (Jurnal Ilmu Pengetahuan Dan Teknologi Komputer)*, *8*(2), 111–116. https://doi.org/10.33480/jitk.v8i2.4060

Hutamaputra, W., & Utaminingrum, F. (2021). Implementasi Facial Landmark dalam Pengenalan Wajah pada Sistem Pembayaran Elektronik. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, *5*(5), 2058–2064.

* Corresponding author

Journal of Computer Networks, Architecture and High Performance Computing
Volume 6, Number 1, January 2024
https://doi.org/10.47709/cnahpc.v6i1.3460

Submitted : January 16, 2024
Accepted : January 21, 2024
Published : January 22, 2024

Mallick, S. (2017). *GSOC2017/data/lbfmodel.yaml*. Github. https://github.com/spmallick/GSOC2017/blob/master/data/lbfmodel.yaml

Mallick, S. (2018). *Facemark : Facial Landmark Detection using OpenCV*. Learnopencv.Com. https://learnopencv.com/facemark-facial-landmark-detection-using-opencv/

Masykur, F., Prasetyo, A., Widaningrum, I., Cobantoro, A. F., & Setyawan, M. B. (2020). Application of Message Queuing Telemetry Transport (MQTT) Protocol in the Internet of Things to Monitor Mushroom Cultivation. *2020 7th International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE)*, 135–139. https://doi.org/10.1109/ICITACEE50144.2020.9239118

Oktavianto, H., Satria Bakti, B., Cobantoro, A. F., Jember, M., Karimata, J., Jember, S., & Timur, J. (n.d.). Untuk Segmentasi Wilayah. *Bina Insani ICT Journal*, *10*(2), 145–153.

Ongky Heri, T., Wahyudiono, N., Dziqie, M., & Al-Farauqi, A. (2021). Proses Sekuritisasi pandemi Covid-19 di Indonesia. *Jurnal Ilmu Politik Dan Komunikasi*, *XI*(2), 32–47.

OpenCV. (n.d.). *cv::face::FacemarkLBF Class Reference*. OpenCV. Retrieved January 12, 2022, from https://docs.opencv.org/3.4/dc/d63/classcv_1_1face_1_1FacemarkLBF.html

Peeri, N. C., Shrestha, N., Siddikur Rahman, M., Zaki, R., Tan, Z., Bibi, S., Baghbanzadeh, M., Aghamohammadi, N., Zhang, W., & Haque, U. (2021). The SARS, MERS and novel coronavirus (COVID-19) epidemics, the newest and biggest global health threats: what lessons have we learned? In *International Journal of Epidemiology* (Vol. 49, Issue 3, pp. 717–726). Oxford University Press. https://doi.org/10.1093/IJE/DYAA033

Ren, S., Cao, X., Wei, Y., & Sun, J. (2014). Face alignment at 3000 FPS via regressing local binary features. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *1*(1), 1685–1692. https://doi.org/10.1109/CVPR.2014.218

Setyawan, M. B., Cobantoro, A. F., Prasetyo, A., Informatika, P. S., Teknik, F., & Ponorogo, U. M. (2020). *Prototype Untuk Monitoring Presensi Siswa Menggunakan*. *13*(1).

* Corresponding author