Real-time Image Processing-based Mask Detection using Streamlit and MobileNet Model

2023 Computer Vision (LB01)

Muhammad Zulham Firdaus
School of Computer Science
Faculty of Computer Science,
Binus University
Jakarta, Indonesia
muhammad.firdaus013@binus.ac.id

Cheryl Almeira
School of Computer Science
Faculty of Computer Science,
Binus University
Jakarta, Indonesia
cheryl.almeira@binus.ac.id

Dary Ramadhan Fajar
School of Computer Science
Faculty of Computer Science,
Binus University
Jakarta, Indonesia
dary.fajar@binus.ac.id

***Abstract*—Technology can adapt and help human problems, one of which is COVID-19. The government issued a policy for all people to wear masks. The limited ability of humans to detect is a major problem, so programs are needed to be able to detect masks, whether the person is wearing a mask or not. The research team used a CNN approach with the Keras, OpenCV, and MobileNet libraries. The use of original data as exercise material model a total of 206 images with people wearing masks and not. The researchers successfully used Streamlit and the model used to perform mask detection. The use of this program can be useful to help implement government policies to reduce the spread of the COVID-19 virus. Furthermore, development will be carried out to improve accuracy and more applicable to other issues.**

Keywords—Computer Vision, MobileNet, Image Processing, Deep Learning, Mask Detection, COVID-19.

I. Introduction

Face mask detection technology is used to determine if an individual is wearing a face covering. Scientist and the researches have proved the efficiency of using mask in order to reduce the virus spreading [1]. The implementation of mask detection systems can aid in controlling the spread of COVID-19 by ensuring compliance with mask-wearing regulations in public spaces [2]. By automatically identifying individuals not wearing masks, these systems can assist in enforcement and promote adherence to mask guidelines. This can slow the transmission of the virus, safeguard vulnerable groups, and lighten the load on healthcare systems. Additionally, the utilization of such systems can contribute to the reopening of the economy and the return to normalcy in the post-pandemic stage.

Computer vision-based mask detection systems typically involve the use of deep learning algorithms, such as convolutional neural networks (CNNs) [3], to analyze images or video streams of individuals. These algorithms can be trained to detect the presence or absence of masks in images and can be used to detect mask-wearing in real time by analyzing live video streams. One of the most utilized CNN (Convolutional Neural Network) architectures for object detection is MobileNet [4], which is designed to be efficient and perform well on mobile devices. The mask detection problem is similar to face detection, detect the face and make it as a critical subtask that will processed [5].

In the process of improving the model's ability to produce maximum accuracy, the research team conducted direct test sampling to the field and produced a total of 200 photos of people with masks and people without masks which will then go through image processing and become data for the trained model.

In this study, the object is to evaluate and compare various face mask detection techniques utilizing deep learning to determine the most effective and suitable method for practical applications and we propose a real-time image processing-based mask detection system using the MobileNet model, which can be deployed on the Streamlit platform.

The evaluation will encompass aspects such as accuracy and processing time for each method, subsequently analyzing the results to identify the optimal approach for face mask detection.

II. Literature Review

The combination of pattern recognition and image processing is called Computer Vision [3]. The combined result of the two things results in a knowledge of the understanding of an image [3]. The process of developing computer vision is growing rapidly with the existence of deep learning methods [6]. In some ways, deep learning can be a reliable solution to solve many things, especially in the field of computer vision, such as object detection, motion tracking, semantic segmentation, etc.

In the process of processing data by computer vision, several supporting things are needed such as image analysis [3]. Some of the detailed stages when analyzing the image are as follows: 1) Image formation, this process recognize and object of the images and stored it to computer; 2) Image preprocessing, before we can do the deeper analyze, the images need to be processed to eliminate objects that will ruin the analyze process, so that the results will be good; 3) Image segmentation, identified and separating the object with the background; 4) Image measurements, measuring the image features, so that the image ability will boosted to fit in models [7] ; 5) Image interpretation, interpreted the extracted images [3].

Computer vision works using pattern recognition, where the image will be processed, identified through changes, image processing, and quality improvement which will then be interpreted [3]. The process will be repeated to get maximum results. The creation of a model that will process image data will also be an important factor in the accuracy score that will be generated.

Face mask detection is one of the solutions to reduce the spread of COVID-19 with the approach of the Artificial Intelligence branch, namely Computer Vision [2]. This solution is maximized using models in deep learning, namely CNN. In classification using Convolutional Neural Network-based, objects captured by the program and directly processed by the model and then processed by the deep learning algorithms [8].

In a Convolutional Neural Network (CNN), the convolution layers work by creating filters that are specifically tailored to the input data [9]. This leads to a hierarchical structure of visual representations that have been optimized for a specific task. After the training process, the CNN model can respond to the intended task through its set of weights and biases. One of the major advantages of CNNs is their ability to generalize, meaning they can process new data they have never encountered before.

In approach to create lightweight class of CNN that able to run in mobile devices and embedded devices, and have a smaller model size, MobileNet is the right used model. They utilize a technique called depth-wise separable convolutions which reduces the number of parameters and computation needed compared to traditional CNNs [4].

MobileNet has sufficient architecture to perform image processing, such as 1) Depthwise Separable Convolution; 2) Network Structure and Training; 3) Width Multiplier: Thinner Models; and 4) Resoultion Multipler: Reduced Representation [4] . This architecture makes the process of filtering and combining is split into two distinct layers. This method of separation leads to a significant reduction in computation and overall model size. The MobileNet architecture is known for its compact size and fast performance. However, there may be instances where a more lightweight version of the model is needed for certain applications. To address this, a parameter called "width multiplier" (α) is introduced, which allows for the creation of even smaller and more efficient models [4].

In conclusion, face recognition and mask detection are using the similar model, the difference is on the model for processing the whole data. CNNs have proven to be the effective ways in mask detection tasks, especially with MobileNet model. However, there are still challenges. Lighting conditions, face direction, face color, object of interference, and others are the example of the challenge.

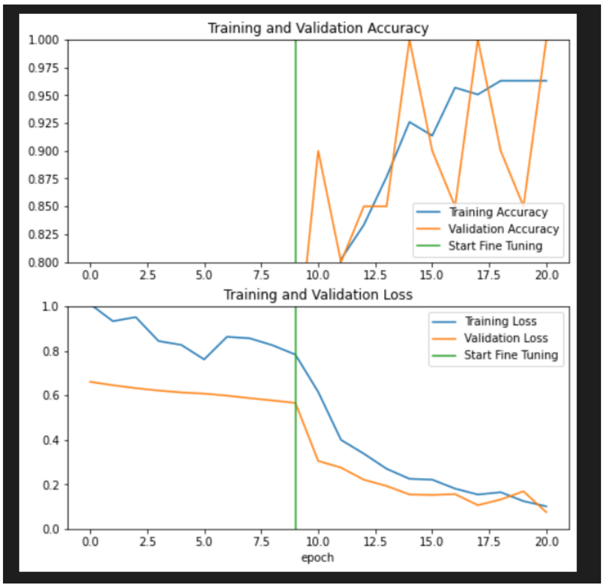
III. METHODOLOGY

A. CNN

Researchers used a pre-trained model that was developed using a dataset that includes various images of people wearing masks and those not wearing masks. This model also uses deep learning using CNN, which can extract complex features from images [7]. The libraries used to predict whether to use a mask or not are Tensorflow and Keras. Tensorflow is a library used to build and train deep learning models, while Keras is used to create architectures and write code more quickly [8]. In its development, researchers use the load_model() function from Keras to create pre-trained models that have been trained. Then the predict_on_batch() function is used to predict the detected faces. The pre-trained model used in the developed code can predict whether a face is detected wearing a mask with a relatively high degree of accuracy. This model was tested using the same dataset and optimized to improve prediction accuracy. This model can be used in various applications, such as face mask detection in healthcare, transportation, and public services.

B. MobileNet

Besides TensorFlow and Keras, another library used is image preprocessing from PIL (Python Image Library), which converts images into arrays that the model can process. Then the preprocess_input from mobilenet_v2 is used to preprocess the images to be processed so that they comply with the standard input required by the model [9].



Images 3.1. Training and Validation

Can be seen in Figure 3.1. model training experiments on MobileNet produce high accuracy in validation accuracy. In addition, loss training also decreased.

C. OpenCV

The researcher uses a pre-trained model developed by OpenCV, which uses the prototypes and weights provided by OpenCV to detect faces in images [10]. This model uses a Convolutional Neural Network (CNN) technique to extract features from complex images. The library used to detect faces is OpenCV. OpenCV is a library developed for image and video processing, which provides various functions that can be used to process images and videos, such as detecting faces, objects, and many more [11]. In the manufacturing process, researchers use the dnn.readNet() function, which is used to load a pre-trained model developed by OpenCV. Then, the setInput() function is used to set the input of the image to be processed, and the forward() function is used to execute face detection.

Besides that, OpenCV is also used to display labels above the detected faces. Researchers use the OpenCV function to display labels over the detected faces and save the processed images [12]. This function allows adding text and geometric shapes, such as boxes, to the processed images. The OpenCV library is used to display the “With Mask” or “Without Mask” label over the detected face using cv2.putText(), which is used to add text to the image. The cv2.rectangle() function adds a rectangle to the detected face. Then, the imwrite() function use to save the processed image.

Apart from OpenCV, in its development, it also uses Numpy, which is used to process arrays and perform arithmetic operations such as calculating face coordinates [13]. The cuteils library is also used to resize images and change their orientation [14]. This model has been tested with the same dataset and optimized to improve prediction accuracy.

D. Streamlit

Researchers also use the streamlit library to display images that have been processed in the form of a web interface [15]. The local_css() function is used to read the css file and apply the necessary changes to the application files. Then, the streamlit_webrtc library displays processed images in a web interface using the webrtc_streamer function.

Overall, the development of this app combines several different libraries to achieve the goal of detecting whether a mask is used or not with a high degree of accuracy. OpenCV is used to detect faces, Tensorflow and Keras are used to make predictions, PIL is used for image preprocessing, and OpenCV, streamlit, and streamlit_webrtc are used to display detection results and store processed images. This app can be used in various places, such as detecting someone wearing a mask or not in the health, transportation, or public services sector.

IV. Experimental Result and Discussion

The experiment results are divided into several explanations, namely system overview, data pre-processing, data split, building model, implementation, and website building using streamlit, each of which is explained as follows.

A. System Overview

The system created is a system to detect the use of masks. Two ways can be used by users, namely, using images or videos in real time. The resulting output predicts whether or not the user uses a mask. The methods used are Keras, MobileNet, OpenCV, and Streamlit. All three are used for model building and model training.

B. Data Collecting

The dataset is needed to conduct training on data that uses masks and does not use masks. Later, the model will differentiate the use of masks and not. The dataset collected comes from 100 data samples totaling 200 data. One hundred data use masks, and 100 data do not use masks. At this stage, the image has a portrait orientation by focusing the object on being visible on the face. After the data is collected, the data is grouped into two parts: masks and without masks.

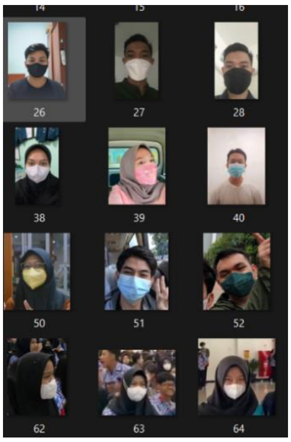


Image 4.1. WithMask

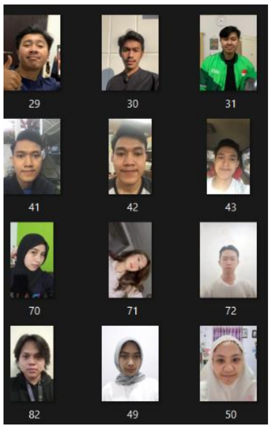


Image 4.2. WithoutMask

C. Split Data

After the data is collected, the data is grouped again. Data is divided into propositions 80%, 10%, and 10%. The distribution is 80% for train data, 10% for test data, and 10% for validation data. From these data, each has a label using a mask and without using a mask.

D. Data Pre-Processing

The next step in the experiment is to do data pre-processing. At this stage, the image size is changed to 160 x 160 with a batch size of 32. After that, buffer settings are also used in the train dataset and dataset validation. Then the results will be displayed by taking one of the samples.

Image 4.3. Data Pre-Processing

E. Building the Model

Then the model is built to continue the experiment. At this stage, several steps are carried out. Namely, the next stage is to build a model. There are six steps in building the model: creating a training image generator for augmentation, the basic model with mobilenetv2, adding model parameters, compiling the model, training the model, and saving the model for the prediction process. The results of the training model using MobileNet were 56% for the first ten epochs and 96% for the second 10 epochs. For a detailed evaluation of predictions, it can be seen in table 1.

TABLE I. Perfomance of MobileNet

	Precision	***Recall***	***f-1 Score***	***Support***
WithMask	1.00	1.00	1.00	10
WithoutMask	1.00	1.00	1.00	10

	Precision	*Recall*	*f-1 Score*	*Support*
Accuracy			1.00	20
Macro avg	1.00	1.00	1.00	20
Weightted avg	1.00	1.00	1.00	20

It can be seen in the table that the evaluation of the model produces very accurate accuracy with an average accuracy of 100%. In addition, it can be seen in the confusion matrix in Figure 4.4. has a TP value of 10, TF 0, FN 0, and TN 0.

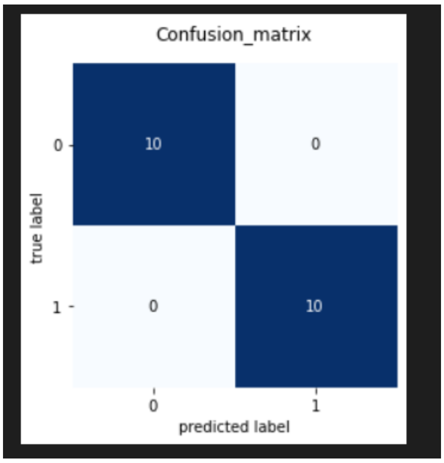


Image 4.4. Confussion Matrix

The next step is to do a prediction test on the data where the predicted data and the labels have the same accuracy, as shown in Figure 4.5. The prediction proves that the model is well-built.

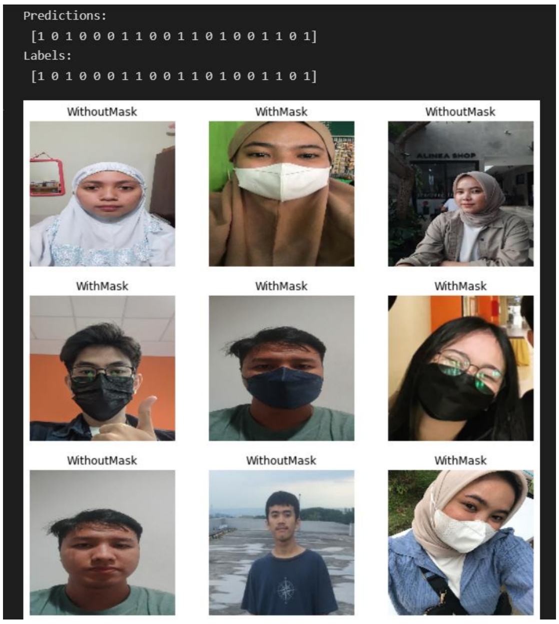


Image 4.5. Image Prediction Test

F. *Implementation*

After the model is saved, the model is implemented on two things: testing on images and video. Can be seen in Figure 4.6. and 4.7. for the results of model implementation from images and videos.

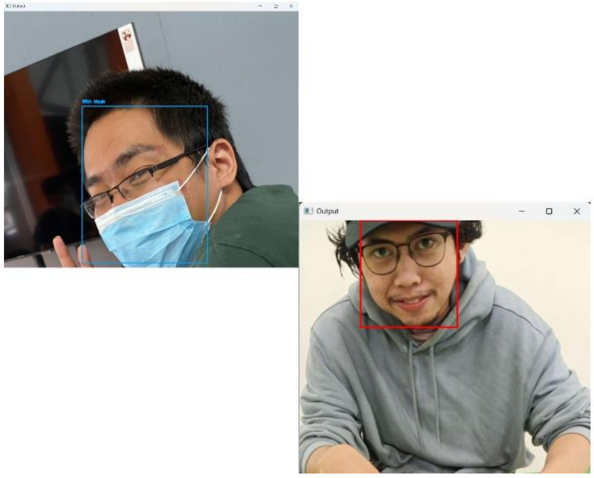


Image 4.6. Image Prediction Result

In the image prediction processing algorithm, the process is reading the shape in the image. Then the model will be matched. The final result is in the form of detecting whether the input image uses a mask or not using a mask.

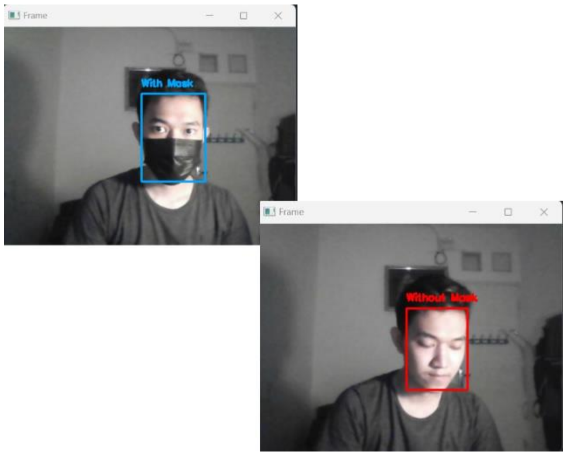


Image 4.7. Video prediction Result

If it is video, then the algorithm reads the frame for each frame in real-time, so face detection for masks will work. Then, from frame detection, re-preprocessing will be carried out, including resizing images, converting them to arrays, and pre-processing input using the MobileNet model. After being processed in the model, the resulting output is a detection that can be seen on the rectangle whether a person is wearing a mask or not.

G. Website Building using Streamlit

To make it easier for users to operate a face mask detection application, the research team used Streamlit as an application platform. Development is done using a model and then taking samples from the image.

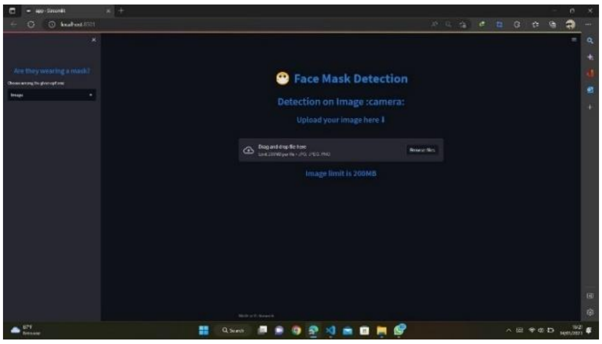


Image 4.8. Upload Form Image

In figure 4.8. there is a website display in the form of an upload for pictures. Users can click browse the file to enter input data.

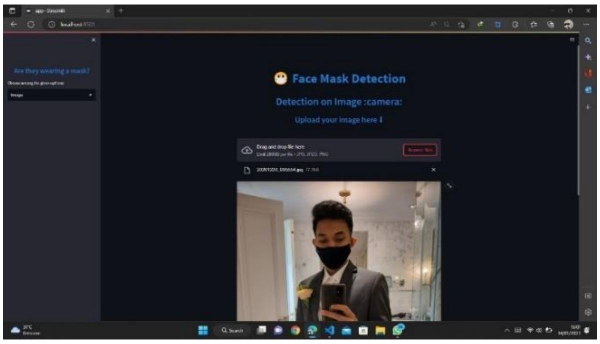


Image 4.9. Image Upload

In figure 4.9. the website display shows that the image has been successfully uploaded. Then, if the user is sure of the image to be used, the user can click the confirm button.

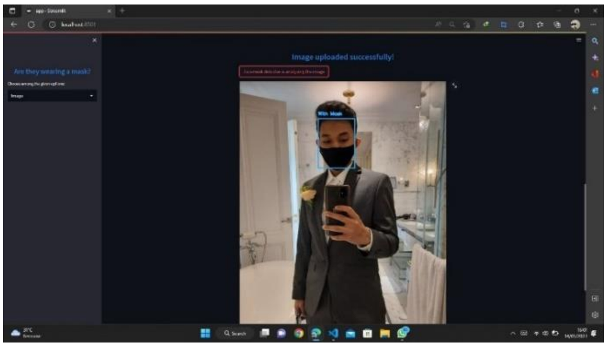


Image 4.10. Image Detection

In figure 4.10. there are results from image detection that show predictions. In this image, the predicted result is an image without a mask.

Image 4.11. Mask Detection using Webcam

In Figure 4.11. is a website display that shows object detection using each user's webcam camera. To start, the user can click the start button so that the program will run. In this image, the prediction result is the detection of wearing a mask.

Image 4.12. Mask Detection using Webcam

In Figure 4.12. is a website display that shows object detection using each user's webcam camera. To start, the user can click the start button, so the program will run. In this image the prediction result is detection of not wearing a mask.

Overall, the application of the MobileNet model to the dataset that has been collected shows promising results with an accuracy of more than 90%. This research can be implemented primarily in areas requiring mask detection, such as public places. In addition, in the future, the development of this research will not only detect masks but will also make it possible to detect anomalies in a person's body. However, to achieve this, further research is needed in the future.

V. Conclusion

The spread of Covid-19 has made the government make many regulations, one of which is regarding the use of masks. The use of masks is needed to reduce the spread of the virus. In this study, an approach was taken to detect masks in real-time and images. The learning model used is MobileNet combined with a Convolutional Neural Network (CNN), which is applied to the dataset that has been collected. The dataset we collected contains 200 individual images with and without masks.

The results showed that using the MobileNet model achieved a classification accuracy of 96 in the dataset. In addition, the accuracy of predictions reaches 100%. Another result is the successful application of the model to real-time image and video predictions.

ACKNOWLEDGMENT

This study involved three students and one assistant lecturers at Bina Nusantara University, Jakarta, Indonesia, 2023. This research was conducted without receiving funds assistance from any agency.

REFERENCES

[1] M. Loey, G. Manogaran, M. H. N. Taha, and N. E. M. Khalifa, "A hybrid deep transfer learning model with machine learning methods for face mask detection in the era of the COVID-19 pandemic," *Measurement (Lond)*, vol. 167, Jan. 2021, doi: 10.1016/j.measurement.2020.108288.

[2] S. Sethi, M. Kathuria, and T. Kaushik, "Face mask detection using deep learning: An approach to reduce risk of Coronavirus spread," *J Biomed Inform*, vol. 120, Aug. 2021, doi: 10.1016/j.jbi.2021.103848.

[3] V. Wiley and T. Lucas, "Computer Vision and Image Processing: A Paper Review," *International Journal of Artificial Intelligence Research*, vol. 2, no. 1, p. 22, Jun. 2018, doi: 10.29099/ijair.v2i1.42.

[4] A. G. Howard *et al.*, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications."

[5] X. Fan and M. Jiang, "RetinaFaceMask: A Single Stage Face Mask Detector for Assisting Control of the COVID-19 Pandemic."

[6] A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, "Deep Learning for Computer Vision: A Brief Review," *Computational Intelligence and Neuroscience*, vol. 2018. Hindawi Limited, 2018. doi: 10.1155/2018/7068349.

[7] C. Z. Basha, B. N. L. Pravallika, and E. B. Shankar, "An efficient face mask detector with pytorch and deep learning," *EAI Endorsed Trans Pervasive Health Technol*, vol. 7, no. 25, pp. 1–8, 2021, doi: 10.4108/eai.8-1-2021.167843.

[8] P. Nagrath, R. Jain, A. Madan, R. Arora, P. Kataria, and J. Hemanth, "SSDMNV2: A real time DNN-based face mask detection system using single shot multibox detector and MobileNetV2," *Sustain Cities Soc*, vol. 66, Mar. 2021, doi: 10.1016/j.scs.2020.102692.

[9] J. Boulent, S. Foucher, J. Théau, and P. L. St-Charles, "Convolutional Neural Networks for the Automatic Identification of Plant Diseases," *Frontiers in Plant Science*, vol. 10. Frontiers Media S.A., Jul. 23, 2019. doi: 10.3389/fpls.2019.00941.

[10] I. Hartono, A. Noertjahyana, and L. W. Santoso, "Deteksi Masker Wajah dengan Metode Convolutional Neural Network."

[11] V. Pisarevsky, "Introduction to OpenCV."

[12] X. Zhu and D. Ramanan, "Face detection, pose estimation, and landmark localization in the wild," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 2879–2886. doi: 10.1109/CVPR.2012.6248014.

[13] S. T. Ratnaparkhi, A. Tandasi, and S. Saraswat, "Face Detection and Recognition for Criminal Identification System," in *2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, 2021, pp. 773–777. doi: 10.1109/Confluence51648.2021.9377205.

[14] H. T. Wibowo, E. Prasetyo Wibowo, and R. K. Harahap, "Implementation of Background Subtraction for Counting Vehicle Using Mixture of Gaussians with ROI Optimization," in *2021 Sixth International Conference on Informatics and Computing (ICIC)*, 2021, pp. 1–6. doi: 10.1109/ICIC54025.2021.9632950.

[15] S. Raghavendra, "Introduction to Streamlit," in *Beginner's Guide to Streamlit with Python : Build Web-Based Data and Machine Learning Applications*, Berkeley, CA: Apress, 2023, pp. 1–15. doi: 10.1007/978-1-4842-8983-9_1.