

Implementation of the Convolutional Neural Network Method to Detect the Use of Masks

Arbiati Faizah ^{1,*}, Pujo Hari Saputro ², Augusta Jannatul Firdaus ³, Raden Nur Rachman Dzakiyullah ⁴

^{1,3} Universitas Hasyim Asy'ari, Indonesia

^{2,4} Universitas Alma Ata, Indonesia

¹ arbiatifaizah@unhasy.ac.id; ² pujo@almaata.ac.id; ³ rezafirdaus@unhasy.ac.id; ⁴ nurrachmandzakiyullah@almaata.ac.id.

* corresponding author

(Received February 5, 2021 Revised February 23, 2021 Accepted February 27, 2021, Available online March 1, 2021)

Abstract

The planet has been taken seriously by Coronavirus disease since the end of 2019. Wearing a mask in public is one of the key means of security for people. Furthermore, certain public service vendors only require clients to use the service if they wear masks correctly. However, based on image processing, there is relatively little study into the discovery of face masks. Almost everybody appears to wear a mask in order to shield themselves from the COVID-19 Pandemic. Monitoring whether people in the crowd wear face masks at the most public place, such as malls, museums, parks, has become increasingly important. The development of an AI approach to deal with if the person wears a face mask and their entrance would significantly assist society. In this article, we will use a deep learning model that is then combined with Keras / TensorFlow & OpenCV, respectively CNN or Confusional Neural Network. The accuracy of the research results obtained from this model is more than 96%.

Keywords: Facemask Detection, CNN, OpenCV, COVID-19

1. Introduction

Due to the Corona COVID-19 virus epidemic across the globe, the practice of using masks in public areas is growing. People used to wear masks before Covid-19 to safeguard their protection against air pollution. While others are self-conscious of their looks, by covering their expressions, they conceal their feelings from the public. Scientists have demonstrated that the propagation of COVID-19 [1] can be prevented by wearing a face mask. The new epidemic virus to reach human health in the last century[2] is COVID-19 (known as coronavirus). The accelerated transmission of COVID-19 will push the World Health Organization to declare COVID-19 a global pandemic in 2020. According to [3], more than five million cases in 188 countries were infected with COVID-19 in less than six months. By close contact and in cramped, overcrowded areas, the virus is spread.

The outbreak of coronavirus has significantly improved the scientific collaboration of the world. Machine learning and Deep Learning-based Artificial Intelligence (AI) can help battle Covid-19 in several respects. In order to forecast the spread of COVID-19, to act as an early warning system for possible pandemics, and to identify susceptible populations, machine learning enables researchers and doctors to analyze vast volumes of data. Provision of healthcare needs support for emerging technology to combat and predict new diseases, such as artificial intelligence, IoT, big data and machine learning. The power of AI is harnessed to tackle the Covid-19 pandemic[4], such as COVID-19 identification on medical chest X-rays[5], to help recognize infection patterns and rapidly monitor and diagnose infections.

In dealing with the spread and propagation of COVID-19[6], policy makers face multiple problems and threats. In certain nations, citizens are required by law to wear facial masks in public. In response to the rapid rise of cases and deaths in many fields, these rules and laws have been established. The method of controlling a wide number of persons, however, is getting more difficult. The method of screening requires identifying who is not wearing a face mask. In France, new AI tech has been built into the Paris Metro system's security camera[7] to ensure passengers wear face masks. The software-developed French start-up DatakaLab[8] states that the aim is not to classify or

apprehend individuals who do not wear masks, but to produce anonymous statistics that will help officials anticipate a possible epidemic of COVID-19.

Object detection is a very important problem in the development vision and challenges of data science, combining segmentation and recognition together, accuracy and real-time time are very important capabilities, in recent years, object detection is used in various applications in various fields such as driving, artificial intelligence (AI), facial recognition, however, various factors will interfere with the detection process, such as angles, occlusion, uneven illumination, and other factors. Krizhevsky [9] defines Confusional neural networks as art products in ImageNet classification, many types of Confusional neural networks have been used for object recognition since then. CNN achieved a good performance in the introduction, especially, the CNN Mask method got the latest results in object recognition, however, it is very time consuming and tiring to get the object Mask for training, so we use reasonable data available open source.

In this article, we present a model of face mask detection based on a branch of the model of machine learning, namely CNN or Confusional Neural Networks. By allowing the identification of people who are not wearing face masks, the proposed model may be combined with a camera to prevent transmission of COVID-19. An interaction among machine learning and the CNN algorithm is the blueprint. For extracting features, we have used deep machine learning and coupled it with 5-layer models that will be used to receive input filters.

2. Related Works

In previous research, when wearing a face mask, the focus of the study was more on facial construction and identity recognition. In this research, our emphasis is on detecting individuals who do not wear face masks to help minimize COVID-19 transmission and spread. The wearing of face masks helps to minimize the spread of COVID-19, researchers and scientists have demonstrated. In [10], a system for defining the requirements for wearing a new face mask was created by the authors. For wearing face masks, they were able to identify three types of circumstances. Correct use of face masks, incorrect use of face masks, and no use of face masks are the types. In the face detection point, the proposed observation has achieved an accuracy of 98.70 percent. To identify the human, Sabbir et al [11] applied Principal Component Analysis (PCA) to masked and exposed face recognition. They discovered that mask wear significantly affected the accuracy of facial resonance using PCA. When recognizable faces are closed, identification precision declines to less than 70 percent. In [12], PCA is also used. A procedure used to detach glasses from a picture of a human frontal face is suggested by the speaker. Using recursive error correction using PCA restoration, deleted sections are reconstructed.

In [13], for face recognition, the authors used the YOLOv3 algorithm. As the backbone, YOLOv3 utilizes Darknet-53. The system proposed achieves a precision of 93.9 percent. With more than 600,000 images, it was trained on the CelebA and WIDER FACE dataset. The Fddb dataset is the test. Nizam et al [14] proposed a new GAN-based network which, by building up the missing holes, can automatically delete masks covering facial areas and regenerate images. A full facial picture that looks normal and realistic is the production of the suggested model.

In [15], the authors provided a method for the identification in the operating room of the presence or absence of mandatory medical masks. The main objective is to eliminate false positive face detection as much as possible without missing mask detection only for medical professionals who do not wear surgical masks to cause alarms. 95 percent precision is archived in the proposed system.

An immersive tool called MRGAN was proposed by Muhammad et al [16]. This approach is focused on acquiring the user's microphone area and using the Generative Adversarial Network to reconstruct this field. In real time, Shaik et al [17] used description and identification of facial emotions. To classify seven facial expressions, they used VGG-16. On the KDEF dataset, the proposed model was trained and achieved an accuracy of 88 percent.

Due to its superior spatial feature extraction capabilities and lower computational costs, CNN played an important role in the pattern recognition task relevant to computer vision in this study [18]. In order to retrieve higher-level characteristics, CNN uses a convolutional kernel to merge initial images or feature maps. What remains an opening question, though, is how to build a better convolutional neural network architecture. The early networking suggested in [19] helped the system to learn the right kernels combinations. To deeper neural networks to be trained, A Residual

Network (ResNet), which can analyze the identity mappings of the previous layer, was proposed by K. He et al [20]. A Cellular Network (MobileNet)[21] was suggested because object detectors are typically used in handheld or embedded devices, where processing resources are very small. To remove features and wise channel convolution to balance channel numbers, it uses careful convolution, so the cost of MobileNet computing is even smaller than networks using regular convolution.

3. Method



Fig. 1. Methodology Process

3.1. Step 1



Fig. 2. Dataset Sample

The first step is to visualize the data that will be used. The dataset we use consists of 1376 images which are divided into 690 images using masks and 686 images using no masks. In order to move on to the next process we have to augment our dataset by selecting images for training, which in the end will produce a number of 560 selected images. Then we divide the details into training data containing images for the CNN model's training and testing data containing images that would be used to validate the model that was developed. Our augmented data is split into 80% (training data) and 20% (testing data).

3.2. Step 2

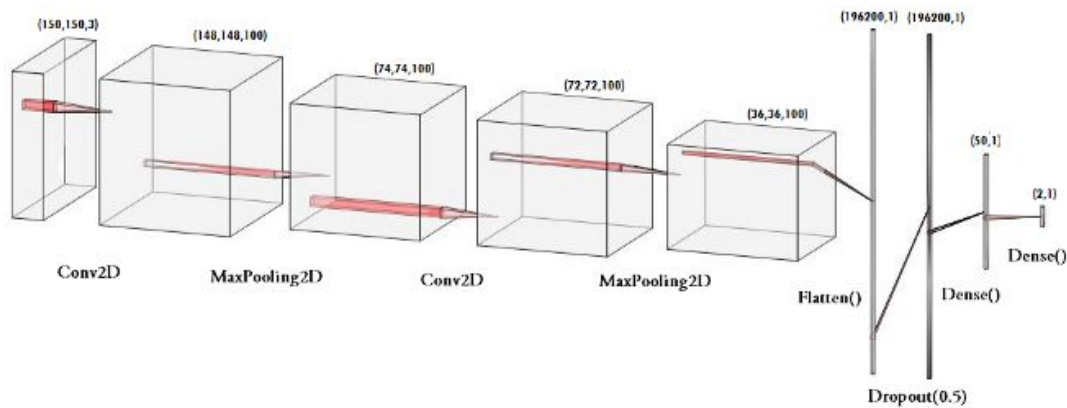


Fig. 3. CNN Model

In the next stage, after augmentation of the data and dividing it into 2 data, namely training and tests. We will start building a CNN Sequential model with several layers namely Conv2D, MaxPooling2D, Flatten, Dropout, and Dense. To make this study easier to understand each layer of the model we will discuss them one by one.

Conv2D, or better known as the 2D convolution layer, is a convolution layer that uses the properties of an image, such as 5 x 5 pixels, to "scan" the image with a filter. The convolution operation calculates the point product between the image pixel value and the weight defined in the filter for each 5 x 5 pixel area in the image. The 2D convolution layer implies that the convolution operation's input is three dimensional. This is a little complicated, since the input is supposed to be two-dimensional. But "2D" in "2D convolution" refers to a filter's movement, which crosses the image in two dimensions. For instance, a color picture with a value of three layers for each pixel: red, blue and green. The filter is then run three times over the image, once for each layer.

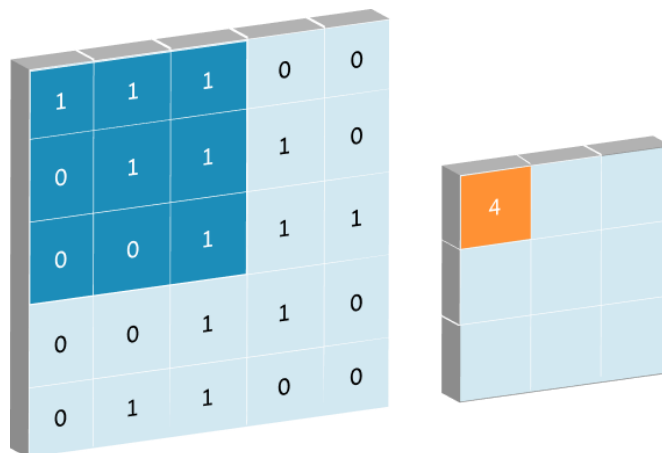


Fig. 4. Conv2D Concept

After downsampling or "merging" the previous convolution results, the same convolutionary form is used sequentially, first to classify features in the original image, and then to identify sub-features in smaller portions of the image. Finally, the aim of this method is to define essential features that can assist in classifying images. MaxPooling2D, since acquiring a convolutional layer, is a process that is carried out. A typical pattern used to organize layers in a convolutional neural network that can be replicated one or more times in a given model is the

addition of a pooling layer after a convolutional layer. In order to construct a new set of aggregated feature maps, the pooling layer works on each feature map separately.

For example, max pooling and min pooling are different forms of pooling. Maximum concatenation operates by choosing the largest value in the square by putting a 2x2 matrix on the feature diagram. The 2x2 matrix is passed around the whole feature map from left to right, taking the highest value for each pass. These values then form a new matrix called the feature map that is pooled. Max pooling works to maintain key characteristics while reducing the scale of the image as well. This helps prevent overfitting, which can occur if too much information is provided to the CNN, especially if the information is not important to the image classification.

A summary version of the features observed at the time of input is the result of using the pooling layer and generating the sampled or aggregated function diagram. They are beneficial since a slight difference in the position of the feature in the feedback observed by the convolutionary layer will result in a feature map at the same location that is merged with the feature. Model invariance to local translation is called this capacity added by coalescing. The next step is flattening, after the collected function map has been acquired. Flattening involves converting the whole matrix of the aggregated function map into a single column that is then fed for processing to a neural network.

Dropout is a method of regularization which provides an estimation of the training in parallel of large numbers of neural networks with different architectures. A number of output layers are skipped or "dropped out." arbitrarily during preparation. This has the effect of making layers appear alike and are viewed as layers with a different number of nodes and connectivity to the previous layer. As a result, any changes to a layer during training are carried out in another "view" of the configured layer.

The dropout has the effect of normalizing the training process, requiring the nodes in the layer to take on more or less responsibility for the input in probabilistic terms. This conceptual model implies that it could be a dropout scenario in which the network layers adjust to correct the previous layer's mistakes together, making the model more robust in turn.

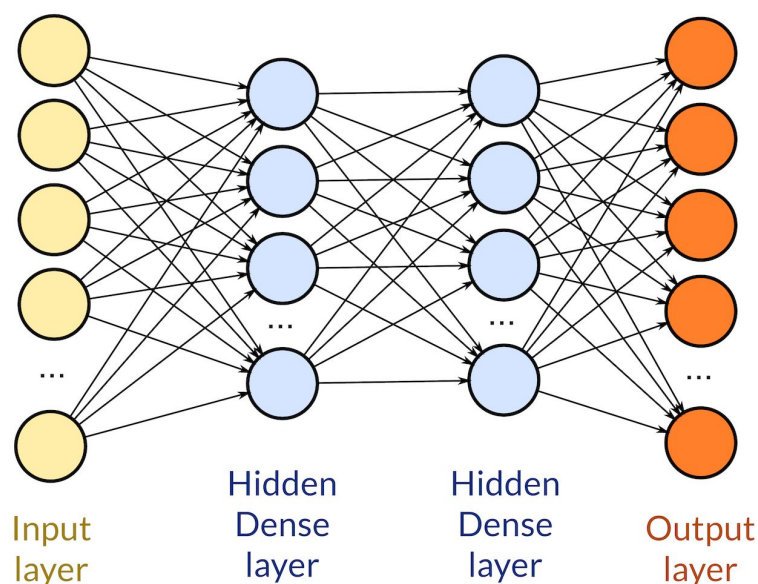


Fig. 5. Dense Layer Model

A deeply connected neural network layer is a dense, dense layer, which indicates that each neuron in the dense layer receives input from all neurons in the previous layer. The material most widely used in models is the dense layer. The thick layer executes matrix-vector multiplication in the background. The values used in the matrix are essentially parameters which, with the help of back propagation, can be trained and modified. A vector with dimension 'm' is the output generated by the dense layer. So, to change the dimensions of the vector, the dense layer is essentially used.

Furthermore, in the pre-training model phase we can see that the data has been separated into 2, there are a total of 556 in the training data and 203 in the testing data. with this we can start the process of the most important part which

is training the CNN model. In the training set and the test set, we modify our drawing to the Sequential model we have generated using the hard library. We have planned to train the 30-epoch model (iterations). For a larger number of epochs, however, we may train to gain greater precision to prevent over-fitting. We see that our model has 98.86 percent accuracy with the training set after the 30th epoch, and 96.19 percent precision with the test set. This means that without over fitting, it is well trained.

```
Epoch 30/30
56/56 [=====] - 51s 919ms/step - loss: 0.1145 - acc: 0.9642 - val_loss: 0.0650 - val_acc: 0.9852
```

Fig. 6. Model Training Result

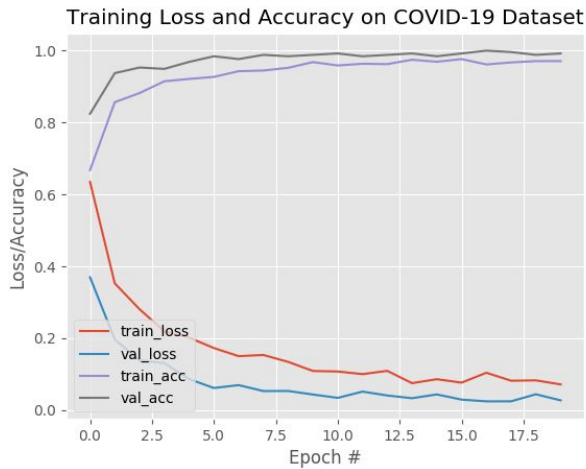


Fig. 7. Training Loss and Accuracy on COVID-19 Datasets

3.3. Step 3

The next step will begin with labeling the obtained result. "For the results, we have prepared two probability labels, namely "With Mask" and "Without Mask". We intend to use it after this to identify whether we use our PC webcam to wear a face mask. We first need to implement face recognition for this. We are using Haar's Feature Dependent Cascade Classifier to detect facial characteristics in this situation. OpenCV developed this cascade classifier to detect frontal faces through the training of thousands of photographs. An .xml file can be downloaded and used for face recognition for the same purpose. And in the final step, we run an infinite loop using the OpenCV library to use our camera to detect faces using the Cascade Classifier. The model would estimate the probability that both groups will have (with or without masks). Labels will be picked and shown around our face depending on which probability is greater. We see from the resulting picture below that the model is able to detect accurately whether or not the person is wearing a mask and shows the same thing on the label.

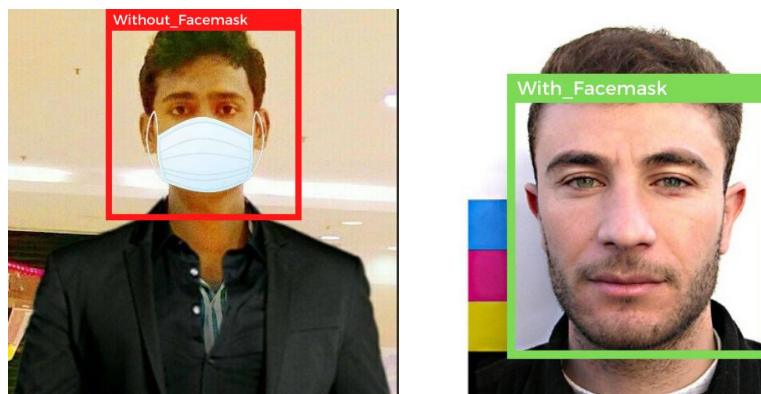


Fig. 8. Model Result

4. Results and Conclusion

We have succeeded in developing a CNN model in this article to detect whether or not anyone is wearing a mask. In different implementations, it can be used. Given the COVID-19 crisis, wearing a mask may be appropriate in the future and this method of detecting if the individual is wearing a face mask may be very helpful. The pandemic COVID-19 corona virus is triggering a global health crisis. Governments are battling this form of virus around the world. According to the World Health Organisation (WHO), protection from COVID-19 infection is a required precautionary step. A machine learning model is presented in this paper for face mask detection. Our model had an accuracy of 98.86 percent with the training set in the data set checked, and an accuracy of 96.19 percent with tests specified. The literature review has presented comparable findings. In terms of testing accuracy, the model used can be assumed to be passed. In future studies, the primary objective to be accomplished is the machine learning approach to attain the lowest consumption time and maximum precision. Using a deeper machine learning model for feature extraction and using the neurotrophic domain is one of the possible future activities as it demonstrates promising promise in problems of classification and detection.

References

- [1] S. Feng, C. Shen, N. Xia, W. Song, M. Fan, and B. J. Cowling, "Rational use of face masks in the COVID-19 pandemic," *Lancet Respir. Med.*, vol. 8, no. 5, pp. 434–436, 2020, doi: 10.1016/S2213-2600(20)30134-X.
- [2] X. Liu and S. Zhang, "COVID-19: Face masks and human-to-human transmission," *Influenza Other Respi. Viruses*, vol. 14, no. 4, pp. 472–473, 2020, doi: 10.1111/irv.12740.
- [3] S. E. Eikenberry *et al.*, "To mask or not to mask: Modeling the potential for face mask use by the general public to curtail the COVID-19 pandemic," *Infect. Dis. Model.*, vol. 5, pp. 293–308, 2020, doi: 10.1016/j.idm.2020.04.001.
- [4] Z. Wang and K. Tang, "Combating COVID-19: health equity matters," *Nat. Med.*, vol. 26, no. 4, p. 458, 2020, doi: 10.1038/s41591-020-0823-6.
- [5] T. Learning, "SS symmetry Within the Lack of Chest COVID-19 X-ray Dataset : A Novel Detection Model Based on GAN and Deep," 2020.
- [6] D. M. Altmann, D. C. Douek, and R. J. Boyton, "What policy makers need to know about COVID-19 protective immunity," *Lancet*, vol. 395, no. 10236, pp. 1527–1529, 2020, doi: 10.1016/S0140-6736(20)30985-5.
- [7] R. Raghavendra and C. Busch, "Novel presentation attack detection algorithm for face recognition system: Application to 3D face mask attack," *2014 IEEE Int. Conf. Image Process. ICIP 2014*, no. 1, pp. 323–327, 2014, doi: 10.1109/ICIP.2014.7025064.
- [8] G. J. Chowdary, N. S. Punn, S. K. Sonbhadra, and S. Agarwal, "Face mask detection using transfer learning of inceptionV3," *arXiv*, pp. 1–11, 2020, doi: 10.1007/978-3-030-66665-1_6.
- [9] Y. Zhang, J. Gao, and H. Zhou, "Breeds Classification with Deep Convolutional Neural Network," *ACM Int. Conf. Proceeding Ser.*, pp. 145–151, 2020, doi: 10.1145/3383972.3383975.
- [10] B. Qin and D. Li, "Identifying facemask-wearing condition using image super-resolution with classification network to prevent COVID-19," *Sensors (Switzerland)*, vol. 20, no. 18, pp. 1–23, 2020, doi: 10.3390/s20185236.
- [11] M. S. Ejaz, M. R. Islam, M. Sifatullah, and A. Sarker, "Implementation of Principal Component Analysis on Masked and Non-masked Face Recognition," *1st Int. Conf. Adv. Sci. Eng. Robot. Technol. 2019, ICASERT 2019*, no. Icasert, 2019, doi: 10.1109/ICASERT.2019.8934543.
- [12] J. S. Park, Y. H. Oh, S. C. Ahn, and S. W. Lee, "Glasses removal from facial image using recursive error compensation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 5, pp. 805–811, 2005, doi: 10.1109/TPAMI.2005.103.
- [13] C. Li, R. Wang, J. Li, and L. Fei, *Face detection based on YOLOv3*, vol. 1031 AISC. 2020.
- [14] N. Ud Din, K. Javed, S. Bae, and J. Yi, "A Novel GAN-Based Network for Unmasking of Masked Face," *IEEE Access*, vol. 8, pp. 44276–44287, 2020, doi: 10.1109/ACCESS.2020.2977386.

- [15] R. Paredes, J. S. Cardoso, and X. M. Pardo, "Pattern recognition and image analysis: 7th Iberian conference, IbPRIA 2015 Santiago de Compostela, Spain, June 17–19, 2015 proceedings," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 9117, pp. 138–145, 2015, doi: 10.1007/978-3-319-19390-8.
- [16] M. K. J. Khan, N. Ud Din, S. Bae, and J. Yi, "Interactive removal of microphone object in facial images," *Electron.*, vol. 8, no. 10, 2019, doi: 10.3390/electronics8101115.
- [17] P. L. Tu and J. Y. Chung, "A new decision-tree classification algorithm for machine learning," *Proc. - Int. Conf. Tools with Artif. Intell. ICTAI*, vol. 1992-November, pp. 370–377, 1992, doi: 10.1109/TAI.1992.246431.
- [18] A. Cutler, D. R. Cutler, and J. R. Stevens, "Ensemble Machine Learning," *Ensemble Mach. Learn.*, 2012, doi: 10.1007/978-1-4419-9326-7.
- [19] E. Mangalova and E. Agafonov, "Wind power forecasting using the k-nearest neighbors algorithm," *Int. J. Forecast.*, vol. 30, no. 2, pp. 402–406, 2014, doi: 10.1016/j.ijforecast.2013.07.008.
- [20] I. Naseem, R. Togneri, and M. Bennamoun, "Robust regression for face recognition," *Proc. - Int. Conf. Pattern Recognit.*, vol. 32, no. 11, pp. 1156–1159, 2010, doi: 10.1109/ICPR.2010.289.
- [21] T. Meenpal, A. Balakrishnan, and A. Verma, "Facial Mask Detection using Semantic Segmentation," *2019 4th Int. Conf. Comput. Commun. Secur. ICCCS 2019*, pp. 1–5, 2019, doi: 10.1109/CCCS.2019.8888092.