A Deep Learning Using DenseNet201 to Detect Masked or Non-masked Face

Faisal Dharma Adhinata[1], Diovianto Putra Rakhmadani[2], Merlinda Wibowo[3], Akhmad Jayadi[4]

[1,2,3]*Faculty of Informatics, Institut Teknologi Telkom Purwokerto, Indonesia*

[4]*Faculty of Engineering and Computer Science, Universitas Teknokrat Indonesia, Indonesia*

[1]faisal@ittelkom-pwt.ac.id, [2]diovianto@ittelkom-pwt.ac.id, [3]merlinda@ittelkom-pwt.ac.id, [4]akhmad.jayadi@teknokrat.ac.id

Abstract - The use of masks on the face in public places is an obligation for everyone because of the Covid-19 pandemic, which claims victims. Indonesia made 3M policies, one of which is to use masks to prevent coronavirus transmission. Currently, several researchers have developed a masked or non-masked face detection system. One of them is using deep learning techniques to classify a masked or non-masked face. Previous research used the MobileNetV2 transfer learning model, which resulted in an F-Measure value below 0.9. Of course, this result made the detection system not accurate enough. In this research, we propose a model with more parameters, namely the DenseNet201 model. The number of parameters of the DenseNet201 model is five times more than that of the MobileNetV2 model. The results obtained from several up to 30 epochs show that the DenseNet201 model produces 99% accuracy when training data. Then, we tested the matching feature on video data, the DenseNet201 model produced an F-Measure value of 0.98, while the MobileNetV2 model only produced an F-measure value of 0.67. These results prove the masked or non-masked face detection system is more accurate using the DenseNet201 model.

Keywords: masked face, coronavirus, MobileNetV2, DenseNet201, F-Measure value

I. INTRODUCTION

Recently, the coronavirus outbreak has infected more than 930,000 people and has caused more than 26,000 deaths in Indonesia [1]. Prevention of coronavirus transmission in Indonesia can implement 3M's behavior, namely using masks, washing hands, and doing social distance. 3M's behavior is critical in breaking the chain of transmission of the coronavirus, which means protecting ourselves and protecting others [2]. At work, especially in an office environment, employees who enter the office are constantly checked for body temperature and the use of masks on their faces. Security guards always ensure that employees who want to enter the office use masks correctly. The use of this mask acts to protect oneself from coughing or sneezing and protects the surrounding air from being mixed with aerosols, which contain viruses in the air for a long time [3].

Air that can be contaminated by the coronavirus is hazardous for employees who are working in the room. In fact, it is often seen that employees wear masks not correctly, even not covering their nose and mouth. Therefore, a masked or non-masked face detection system is needed to prevent the transmission of the coronavirus. Through videos data, the system is expected to provide a warning in the form of a marker on the face that is not wearing a mask properly. Besides, this system must be able to recognize all types of masks of various colors and patterns. The use of these various motifs requires application in Artificial Intelligence [4] for training data on various mask motifs.

Computer vision is a knowledge field that processes image or video data for identification, description, and further analysis to produce meaningful information [5]. One technique that processes image data to classify faces wearing masks or not is using deep learning techniques. Several researchers have been developed masked or non-masked face detection systems using deep learning techniques [6][7][8][9]. One of the deep learning techniques is using transfer learning, where the model has been trained using the ImageNet dataset [10]. Researchers Joshi et al. [7] using transfer learning MobileNetV2 resulted in an F-measure value below 0.9. The results of this F-measure indicate that the system being built is not accurate enough.

Recently, several researchers often use transfer learning techniques for image data classification. Researchers Roslidar et al. [11] reviewed several transfer learning models (ResNet101, DenseNet201, MobileNetV2), resulting in 100% accuracy with the DenseNet201 model for classifying the thermal breast image. Researchers from Mporas and Naronglerdrit [12] also analyzed the use of several transfer learning models to identify the coronavirus through Chest X-ray images. The best performance was using the

DenseNet201 transfer learning model. Therefore, we propose using the DenseNet201 transfer learning model to classify masked or non-masked face image data. In this study, the results of F-measure using the MobileNetV2 model will also be compared. We hope to get the best F-measure for the mask wear detection system.

II. METHOD

The masked or non-masked face detection system starts with the input of a masked or non-masked face image dataset. The data is divided into two parts, namely training data and validation data. All data is pre-processed using resize and augmented on data before passing them to the model. This study will compare two models, namely DenseNet201 and MobileNetV2. The result of the model building is an h5 file. The formed model is evaluated by looking at the best training loss, training accuracy, validation loss, and validation accuracy. The best model is used as a matching feature with real-time video. The video data is extracted into frames. The video frame is carried out by detecting the recorded face using the Haar Cascade classifier. The detected face is resized for processing the matching feature. The result is a masked face bounding box or not. Fig. 1 shows the proposed method in this research.

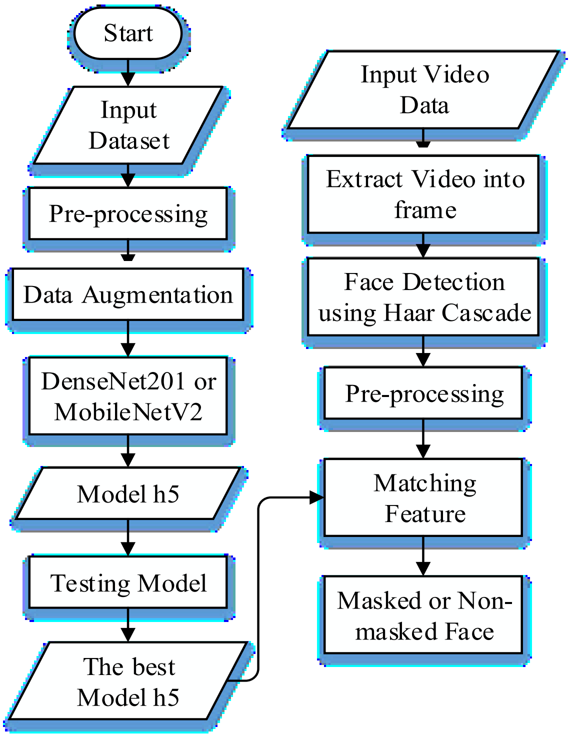


Fig. 1 Flowchart of the masked or non-masked face detection system

A. Dataset

In this study, using the Masked Face Recognition Dataset (MFRD) [13]. This dataset consists of 90,000 face images without masks, 2,203 masked faces of 525 people. Fig. 2 shows an example of the MFRD dataset. All images in this study use Red, Green, Blue (RGB) color images.

A masked or non-masked face detection system uses 2000 masked face image data and non-masked face image data, respectively. The original image is resized to a resolution of 224 x 224 on pre-processing stage. The purpose of resizing is to adjust the size for processing using the DenseNet201 and MobileNetV2 models. This study uses the preprocess_input function from DenseNet201 or MobileNetV2 to an adequate image of the model's format. The dataset is divided into two, 80% as training data and 20% as validation data.

B. Data Augmentation

This research uses data augmentation with the ImageDataGenerator class from the Keras library that incredibly facilitates the usage of geometric augmentations [14]. This study's generator or augmentation uses the parameter of rotation_range to rotate images of 20 degrees randomly, zoom_range for randomly zooming images of 0.15, width_shift_range, and height_shift_range for a horizontal shift of image and a vertical shift of image of 0.2, shear_range for shear an image of 0.15. Then augmentation also uses parameter of horizontal_flip for flipping along the vertical and fill_mode with "nearest" value, which replaces the empty area with the nearest pixel values.

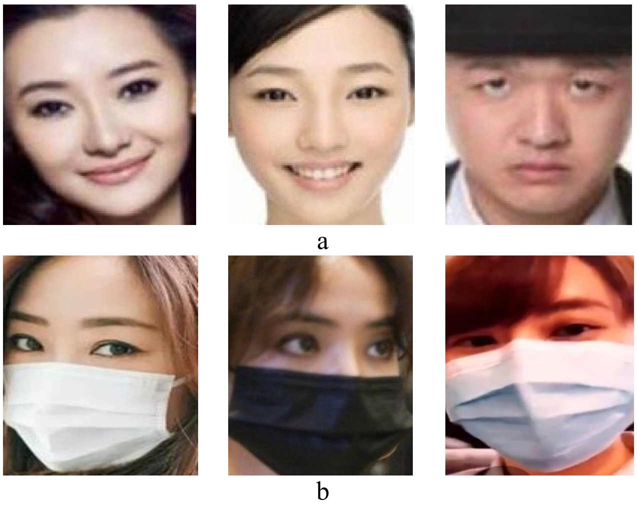


Fig. 2 Samples of MFRD, a) face image without mask, b) face image with mask [13]

C. CNN Framework

Convolutional Neural Network (CNN) is often used to process data in the form of many arrays, for example, a two-dimensional color image with three color channels, namely Red, Green, and Blue. There are various kinds of data processing on CNN, namely one dimension for voice signals, usually used for language, two dimensions for image data processing, and 3 dimensions for video data processing [15]. Based on the CNN architectural concept, various transfer learning models that have been trained using the ImageNet dataset have been developed. The two models that will be compared in this study are DenseNet201 and MobileNetV2. A pre-trained model is a network that has been trained previously on a large dataset, typically for image classification. Researchers can either use the pre-trained model as-is or use transfer learning to tailor it to a specific task. Transfer learning for image classification is based on the premise that if a model is trained on a sufficiently large and diverse dataset, it can effectively serve as a generic model of the visual world.

D. DenseNet201

The Dense Convolutional Network (DenseNet) architecture connects each layer directly to the other in a feed-forward fashion. Meanwhile, the traditional Convolutional Neural Network (CNN) with the L layer has an L relationship, where the relationship with each other has a direct relationship L (L + 1) / 2. DenseNet contains a very narrow layer (12 filters per layer) with a small set of feature maps included in the network's collective information [11]. The advantages of DenseNet are that it is light on the problem of gradients, feature deployment, encourages feature reuse, and its functionality reduces the number of parameters [16].

The DenseNet-201 is a 201-layer convolutional neural network. It uses the ImageNet database to load a pre-trained network that has been trained on over a million images. The network will classify images into 1000 different object categories, including keyboards, mice, pencils, and various animals. As a result, the network has learned extensive feature representations for a variety of image types. The network's image input size is 224 x 224. Fig. 3 shows that each layer contains batch normalization (BN), ReLU activation, and convolution with a $3x3$ filter. In each block, there is an input in the form of a matrix that corresponds to the image pixel, which then goes to the batch normalization stage, which helps reduce overfitting during training. ReLu activation to change the x value to 0 if the x value is negative, but vice versa for the x value does not change if it is not less than 0. Convolutional with a $3x3$ filter to process a matrix image that has passed the ReLu activation stage will be multiplied by a convolution matrix with a filter $3x3$, and the resulting output is a previously processed matrix value.

E. MobileNetV2

MobileNetV2 uses the ImageNet dataset, which yields better accuracy than MobileNetV1 with fewer parameters [17]. The MobileNetV2 architecture still uses depthwise and pointwise convolution on MobileNetV1. Additional features on MobileNetV2 are linear bottlenecks and shortcut connections between bottlenecks. Fig. 4 shows the MobileNetV2 architecture. There are input and output sections between the models in the bottleneck component. Simultaneously, the inner layer encapsulates the function of the model to change the input from a lower-level concept (pixels) to a higher-level descriptor (image classification). Thus, as with residual connections in traditional CNN architectures, shortcuts between bottlenecks will make the training process faster and with better accuracy.

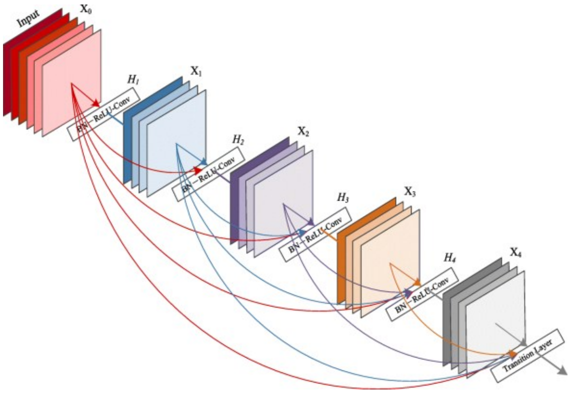


Fig. 3 DenseNet architecture [16]

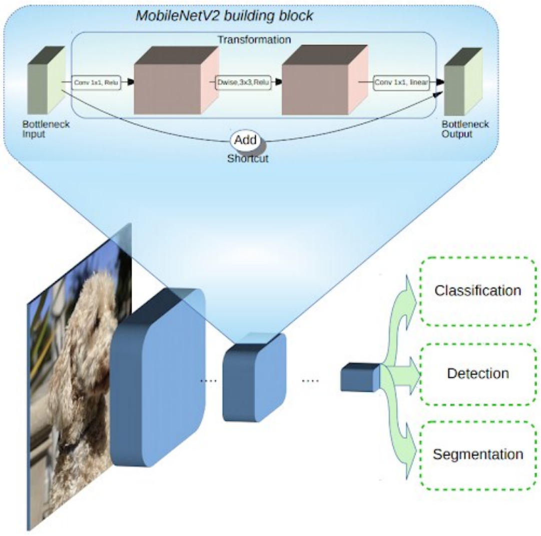


Fig. 4 MobileNetV2 architecture [18]

1) Inverted Residuals: Residual blocks use a skip connection to connect the beginning and end of a convolutional block. By combining these two states, the network gains access to previous activations that were not modified during the convolutional block. This approach proved to be critical for the development of large-scale networks. A residual block creates a skip connection (shortcut) between two wide layers, while the layers in between are narrow, as shown in Fig. 4. When we examine the skip connection in greater detail, we notice that an original residual block takes a wide → narrow → wide approach to channel count. The input has a high channel count, which is compressed using a low-cost 1x1 convolution. Thus, the subsequent 3x3 convolution has significantly fewer parameters. Another 1x1 convolution increases the number of channels to add input and output at the end.

2) Linear Bottlenecks: Because multiple matrix multiplications cannot be reduced to a single numerical operation, MobileNetV2 uses non-linear activation functions in neural networks. It enables the construction of neural networks with multiple layers. Simultaneously, the activation function ReLU, which is frequently used in neural networks, discards values less than zero. This information loss can be mitigated by increasing the number of channels on the network to increase its capacity. With inverted residual blocks, it does the inverse and compresses the layers containing the skip connections (shortcut).

3) ReLu6: The structure of a convolutional block incorporating inverted residuals and linear bottlenecks is illustrated in Fig. 4. The first aspect simply adds Batch Normalization to each convolutional layer. The second addition is not as prevalent. The creators use ReLU6 rather than ReLU, which caps the value of activations at well 6. As long as the value is between 0 and 6, the activation is linear. It is advantageous when working with fixed-point inference. It restricts the information left of the decimal point to three bits, ensuring that the precision right of the decimal point is guaranteed.

F. Format Naskah

In research on masked or non-masked face detection systems, the discussion results are based on the value of training loss, training accuracy, validation loss, and validation accuracy. All training and validation results will be recorded at each specific epoch. Processing time is used to compare the training process speed of the MobileNetV2 model with DenseNet201. After the model training is complete, we will test with the recall, precision, and F-Measure values for each video frame. The recall, precision, and F-Measure formulas are shown in (1), (2), and (3) [19].

$$Recall = \frac{TP}{TP+FN} \tag{1}$$

$$Precision = \frac{TP}{TP+FP} \tag{2}$$

$$F - Measure = 2\,x\,\frac{recall\ x\ precision}{recall+precision} \tag{3}$$

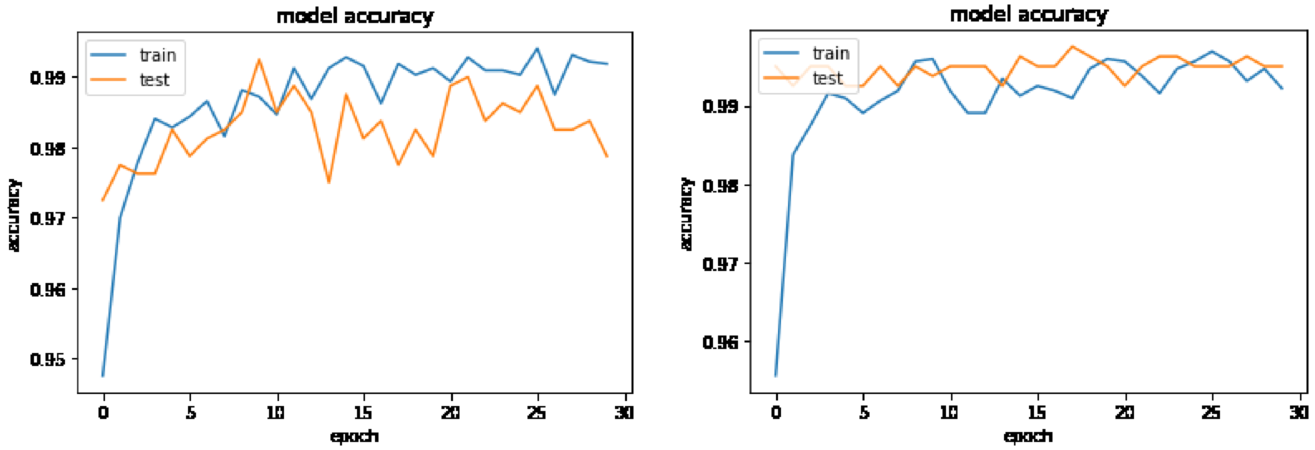
True positive (TP) is the result if the model correctly predicts the positive class. Likewise, true negative (TN) is the result if the model correctly predicts the negative class. False positives (FP) result from the model incorrectly predicting the positive class and false negative (FN) result if the model incorrectly predicts the negative class.

III. RESULTS AND DISCUSSION

In this research, experiments use an image 224 x 224 x 3. Number 3 is a three-color channel, namely Red, Green, and Blue. This size adapts to CNN models, namely DenseNet201 and MobileNetV2. The CNN model is implemented to classify two classes of masked faces and non-masked faces. We tuned the parameter epoch 10, 20, and 30 to see the loss and accuracy performance. We save a model with a smaller training loss value at this training stage than the previous epoch. This study using Adam optimizers and using batch size 16. Table I and Fig. 5 are the results of training data using DenseNet201 and MobileNetV2.

TABLE I
THE TRAINING RESULTS OF MODEL BUILDING WITH MOBILENETV2 AND DENSENET201

Model CNN	10 Epochs				20 Epochs				30 Epochs			
	Train Loss	Train Acc.	Val Loss	Val Acc.	Train Loss	Train Acc.	Val Loss	Val Acc.	Train Loss	Train Acc.	Val Loss	Val Acc.
MobileNetV2	0.0315	0.9872	0.0486	0.9925	0.0255	0.9912	0.0667	0.9787	0.0277	0.9919	0.0688	0.9787
DenseNet201	0.0154	0.9959	0.0123	0.9937	0.0132	0.9959	0.0165	0.995	0.0108	0.9969	0.0163	0.995



a b

Fig. 5 Graph of training results, a) model accuracy using MobileNetV2, b) model accuracy using DenseNet201

A. Training Result Analysis

In the training analysis, several metrics are recorded, as shown in Table I. The metrics to be seen are training loss, training accuracy, validation loss, and validation accuracy. This study uses batch size 16, which means that the dataset will be divided into batches for training by the Neural Network, where one batch is 16 data.

Based on Table I, the DenseNet201 model produces higher accuracy in training and validation than the MobileNetV2 model. Likewise, the loss values in both training and validation, the results from the DenseNet201 model, are smaller than those from MobeliNetV2. Comparing the two models on epoch 20 and 30 shows that the DenseNet201 model is better than MobileNetV2 both in loss and accuracy values. The results of training accuracy and validation (testing) accuracy are shown through the graph in Fig. V. The accuracy graph on the MobileNetV2 model looks a little overfitting at epoch 10 to 30. It shows that the increasing epoch value on the MobileNetV2 model will make the model's validation accuracy value decrease.

In contrast to the accuracy graph generated from the DenseNet201 model, the graph of training accuracy and validation (testing) accuracy appear coincided, which means it shows the optimum model. The results of these two models will be used for testing video data on masked or non-masked faces.

B. Testing Result Analysis

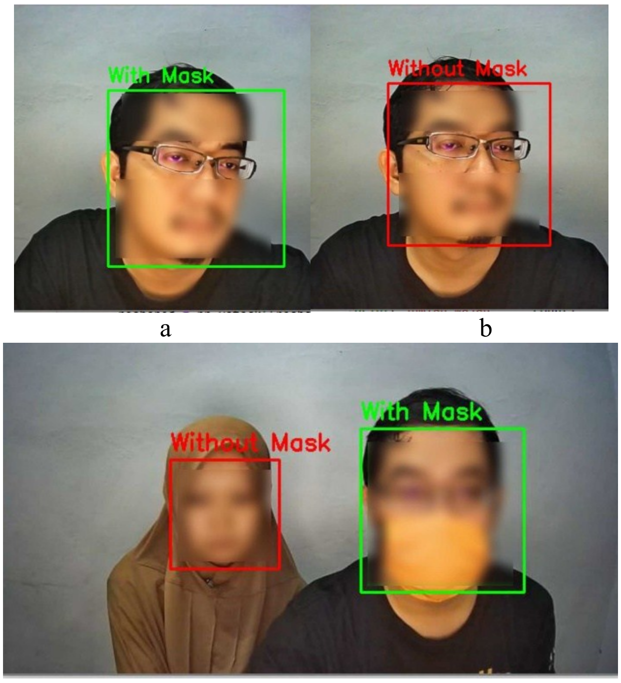
This research, testing on video data, use 30 fps video. There are objects of humans wearing masks and not wearing masks on their faces. The test results are shown in Table II. The DenseNet201 model produces an F-Measure value of 0.98, which is very high compared to using MobileNetV2.

The recall value for using the MobileNetV2 model is 0.50 because it predicts people wearing glasses as masked people, as shown in Fig. 6a. The low recall value also has an impact on the F-Measure value, which is only 0.67. The matching feature results with the DenseNet201 model produce the optimum F-measure value of 0.98. It proves that the masked or non-masked face classification using the DenseNet201 model is better than using the MobileNetV2 model. These results prove that face masks' detection using the DenseNet201 model is better than previous research [7]. The number of parameters for the DenseNet201 model is 20 M while the MobileNetV2 model is only 3.5 M. Fig. 6 shows an example of testing results using real-time video data.

This study has a drawback where the detected face is still limited to the face facing the camera's front. Therefore, future research can modify face detection, which can detect faces from all directions. It is hoped that the detection of mask use can be more accurate with all recorded facial directions.

TABLE II
THE TESTING RESULTS OF MODEL BUILDING WITH MOBILENETV2 AND DENSENET201

Model CNN	Recall	Precision	F-Measure
MobileNetV2	0.50	1	0.67
DenseNet201	0.97	1	0.98



a b

c

Fig. 6 The example of testing results, a) bespectacled face using MobileNetV2, b) bespectacled face using DenseNet201, c) real-time video result

IV. CONCLUSION

This research focuses on improving the accuracy of masked face detection systems or not. Previous research using the MobileNetV2 model resulted in inaccurate accuracy. Then we use the DenseNet201 model, which has more parameters than MobileNetV2. The result is that the detection system produces an F-measure value of 0.98. Further research will focus on masked or non-masked face detection by modifying face detection methods to detect faces in all directions. So it can produce masked or non-masked face detection more accurate.

REFERENCES

[1] WHO Indonesia, "Coronavirus Disease Situation Report World Health Organization," 2020.

[2] C. Arumsari, E. Yulianto, E. N. Afifah, U. M. Tasikmalaya, and U. Siliwangi, "SOSIALISASI DALAM RANGKA MEMELIHARA KESADARAN WARGA PADA KESEHATAN DI MASA PANDEMI COVID-19 Pendahuluan," vol. 2, no. 1, pp. 272–276, 2021, doi: 10.31949/jb.v2i1.676.

[3] S. E. Hwang, J. H. Chang, O. Bumjo, and J. Heo, "Possible Aerosol Transmission of COVID-19 Associated with an Outbreak in an Apartment in Seoul, South Korea, 2020," *Int. J. Infect. Dis.*, no. xxxx, pp. 0–3, 2020, doi: 10.1016/j.ijid.2020.12.035.

[4] K. Podbucki, J. Suder, T. Marciniak, and A. Dabrowski, "CCTV based system for detection of anti-virus masks," *Signal Process. - Algorithms, Archit. Arrange. Appl. Conf. Proceedings, SPA*, vol. 2020-Septe, pp. 87–91, 2020, doi: 10.23919/spa50552.2020.9241303.

[5] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 580–587, 2014, doi: 10.1109/CVPR.2014.81.

[6] M. R. Bhuiyan, S. A. Khushbu, and M. S. Islam, "A Deep Learning Based Assistive System to Classify COVID-19 Face Mask for Human Safety with YOLOv3," *2020 11th Int. Conf. Comput. Commun. Netw. Technol. ICCCNT 2020*, 2020, doi: 10.1109/ICCCNT49239.2020.9225384.

[7] A. S. Joshi, S. S. Joshi, G. Kanahasabai, R. Kapil, and S. Gupta, "Deep learning framework to detect face masks from video footage," *arXiv*, pp. 435–440, 2020, doi: 10.1109/CICN.2020.78.

[8] I. B. Venkateswarlu, "Face mask detection using MobileNet and Global Pooling Block," pp. 0–4, 2020.

[9] A. Oumina and N. El Makhfi, "Control The COVID-19 Pandemic : Face Mask Detection Using Transfer Learning," in *International Conference on Electronics, Control, Optimization and Computer Science (ICECOCS)*, 2020, pp. 0–4.

[10] L. Fei-Fei, J. Deng, and K. Li, "ImageNet: Constructing a large-scale image database," *J. Vis.*, vol. 9, no. 8, pp. 1037–1037, 2010, doi: 10.1167/9.8.1037.

[11] R. Roslidar, K. Saddami, F. Arnia, M. Syukri, and K. Munadi, "A study of fine-tuning CNN models based on thermal imaging for breast cancer classification," *Proc. Cybern. 2019 - 2019 IEEE Int. Conf. Cybern. Comput. Intell. Towar. a Smart Human-Centered Cyber World*, pp. 77–81, 2019, doi: 10.1109/CYBERNETICSCOM.2019.8875661.

[12] I. Mporas and P. Naronglerdrit, "COVID-19 Identification from Chest X-Rays," *Proc. Int. Conf. Biomed. Innov. Appl. BIA 2020*, pp. 69–72, 2020, doi: 10.1109/BIA50171.2020.9244509.

[13] Z. Wang *et al.*, "Masked face recognition dataset and application," *arXiv*, pp. 1–3, 2020.

[14] C. Shorten and T. M. Khoshgoftaar, "A survey on Image Data Augmentation for Deep Learning," *J. Big Data*, vol. 6, no. 1, 2019, doi: 10.1186/s40537-019-0197-0.

[15] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015, doi: 10.1038/nature14539.

[16] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 2261–2269, 2017, doi: 10.1109/CVPR.2017.243.

[17] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, “MobileNetV2: Inverted Residuals and Linear Bottlenecks,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 4510–4520, 2018, doi: 10.1109/CVPR.2018.00474.

[18] T. Choudhury, A. Anggarwal, and R. Tomar, “A Deep Learning Approach to Helmet Detection for Road Safety,” *J. Sci. Ind. Res. (India).*, vol. 79, no. June, pp. 509–512, 2020.

[19] T. Fawcett, “An introduction to ROC analysis,” *Pattern Recognit. Lett.*, vol. 27, no. 8, pp. 861–874, 2006, doi: 10.1016/j.patrec.2005.10.010.