IJCCS (Indonesian Journal of Computing and Cybernetics Systems)
Vol.16, No.1, January 2022, pp. 67~78
ISSN (print): 1978-1520, ISSN (online): 2460-7258
DOI: https://doi.org/10.22146/ijccs.72075

Behind the Mask: Detection and Recognition Based-on Deep Learning

Ade Nurhopipah*[1], Irfan Rifai Azziz[2], Jali Suhaman[3]
[1,2,3] Department of Informatics, Universitas Amikom Purwokerto, Purwokerto, Indonesia
e-mail: ***[1]ade_nurhopipah@amikompurwokerto.ac.id**, [2]rifaiirfan41@gmail.com,
[3]jalisuhaman2018@gmail.com

Abstrak

Berbagai prosedur pencegahan COVID-19 dilakukan untuk mendukung pelayanan publik dan kelangsungan bisnis dalam situasi pandemi. Pemantauan penggunaan masker secara manual tidak efisien karena membutuhkan sumber daya untuk memantau orang setiap saat. Oleh karena itu, tugas ini dapat didukung oleh sistem pengawasan otomatis berbasis Deep Learning. Kami melakukan deteksi masker dan pengenalan wajah untuk dataset dari lingkungan nyata. YOLOV3 sebagai detektor satu tahap diimplementasikan untuk menghasilkan bounding box wilayah wajah dan prediksi kelas secara bersamaan. Dalam pengenalan wajah, kami membandingkan kinerja tiga model pre-trained, yaitu ResNet152V2, InceptionV3, dan Xception. Deteksi masker menunjukkan hasil yang menjanjikan dengan MAP=0.8960 pada pelatihan dan MAP=0.8957 pada validasi. Kami memilih model Xception dalam pengenalan wajah karena kualitas yang sebanding dengan ResNet152V2 tetapi memiliki parameter yang lebih sedikit. Xception mencapai nilai loss minimal dalam validasi sebesar 0,09157 dengan akurasi sempurna pada citra wajah yang lebih besar dari 100 piksel. Secara keseluruhan, sistem ini memberikan hasil yang menjanjikan dan dapat mengidentifikasi wajah, bahkan jika wajah tersebut bermasker.

Kata kunci*— Deep Learning, deteksi masker, model pre-trained, pengenalan wajah, YOLO*

Abstract

COVID-19 prevention procedures are executed to support public services and business continuity in a pandemic situation. Manual mask use monitoring is not efficient as it requires resources to monitor people at all times. Therefore, this task can be supported by automated surveillance systems based on Deep Learning. We performed mask detection and face recognition for a real-environment dataset. YOLOV3 as a one-stage detector was implemented to simultaneously generate a bounding box of the face area and class prediction. In face recognition, we compared the performance of three pre-trained models, namely ResNet152V2, InceptionV3, and Xception. The mask detection showed promising results with MAP=0.8960 on training and MAP=0.8957 on validation. We chose the Xception model for face recognition because it has equal quality as ResNet152V2 but has fewer parameters. Xception achieved a minimal loss value in the validation of 0.09157 with perfect accuracy on facial images larger than 100 pixels. Overall the system delivers promising results and can identify faces, even those behind the mask.

Keywords— *Deep Learning, face recognition, mask detection, pre-trained model, YOLO*

Received January 8th,2022; Revised January 30th, 2022; Accepted January 31th, 2022

1. INTRODUCTION

The Corona Virus Disease (COVID-19) pandemic, which began to spread in December 2019, made all countries tighten health protocols in their respective regions. The governments make regulations on personal protective equipment, one of which is wearing masks in public places. Supervision of regulatory compliance is usually done manually by paying attention to people in the observed environment. This manual control is not efficient because officers have to monitor everyone at all times. Therefore, the manual monitoring method can be replaced by automatic monitoring through the image processing system. In psychology and social communication, masks affect the accuracy of humans in recognizing faces [1], [2], even if the mask used is transparent [3]. This problem also happens in image processing systems for face authentication. Many studies have been carried out to recognize faces under challenging conditions due to differences in pose, light, image quality, occlusion, etc. However, research on face detection using masks has only bloomed in the last two years.

Convolutional Neural Network (CNN), one of the concepts of Deep Learning, has been widely used in pattern recognition. Using CNN, researchers have carried out studies to classify faces, whether masked or unmasked [4]–[7]. A comparison of five different types of CNN architecture is also implemented in research [8], resulting in InceptionV3 and Xception as the most outstanding models. Analysis [9] found an approach that was considered more efficient by removing the face area covered by a mask. Furthermore, they applied three types of pre-trained CNNs, namely, VGG-16, AlexNet, and ResNet-50, to extract features from the acquired areas (eye and forehead area). The experimental results on the "Real-World-Masked-Face-Dataset" show the highest accuracy by 91.3%. Research [10] proposed an approach to detect face masks in videos using the MTCNN face detection model. A MobileNetV2 object detector then follows to identify whether a face is masked or not. The face detection achieved an accuracy of 81.84%, and the face mask detection achieved 81.74%.

The detection problem requires a classifier that can determine whether the face is masked or not, and we need to know the location of this face. Mask detection is challenging because existing detections have only been developed for faces without masks. Many traditional detection algorithms are widely used based on handcrafted feature extraction. With advances in Deep Learning, neural networks can now learn features automatically using some architectures such as R-CNN, Deep MultiBox, OverFeat, MultiGrasp, You Only Look Once (YOLO), Single-Shot Detector (SSD), and Retina Net. There are two types of detectors, namely one-stage detectors and two-stage detectors. The one-stage detector recognizes the region proposal as a simple regression problem by taking an input image, resulting in the class probabilities and bounding box coordinates. Meanwhile, the two-stage detector looks for bounding box and probability separately. First, they find the region proposal in the image, then apply a classifier to this region to find the class probabilities [11]. Detection using a one-stage detector is an efficient approach that can detect the use of masks.

Face recognition can follow the mask detection process. We can do face recognition even though a mask covers the face. Research [12] has performed face detection with seven types of occlusion, including masked faces using the Multi-Task Cascaded Convolutional Neural Network (MTCNN). The Google FaceNet embedding model detected extracted faces. Finally, the classification was done with the Support Vector Machine (SVM). This study resulted in a maximum accuracy of 98.50% on the test set. Research [13] used a classification model based on the MobileNetV2 architecture. The FaceNet model was used as a feature extractor, and a feedforward multilayer perceptron does the face recognition. The experimental results show an accuracy of 99.65% in determining whether a person wears a mask or not. The face recognition of 10 people using masks achieved an accuracy of 99.52%, while face recognition without masks achieved an accuracy of 99.96%. In research [15], two pre-trained deep learning architectures, VGG16 and MobileNetV2, have been implemented for masked face recognition. Histogram of Gradients (HOG) technique was used to extract the relevant features,

and Support Vector Machines (SVM) was used for classification. MobileNetV2 was the best model with an accuracy of 96.8%.

This study will perform mask detection using YOLO and face recognition using a pre-trained model. YOLO is one of the popular one-stage detector architectures that has been carried out in mask detection and shows its efficient and high-quality accuracy [14]–[18]. We implemented YOLO with our dataset and conducted training to build a model detector. Furthermore, we compared three CNN-based models in the face recognition process, namely Residual Network (ResNet), Inception, and Xception. We used these models as feature extraction and added some layers for classification. We did not remove the face area covered as did in [9] and extracted the features directly from the detected faces from the previous process. We use the best models and apply them to the face recognition system for combination masked and unmasked faces.

2. METHODS

2.1 CNN Architectures

CNN is a Deep Learning algorithm inspired by the human ability to recognize objects by building an abstract concept through a series of overlapping layers [19]. The four basic operations in CNN are convolution, activation function, pooling layer, and fully connected layer [20]. The purpose of the convolution layer is to extract image features using various kernels for feature detection and extraction. The activation function is applied to the convolution results to extract non-linear features. The pooling layer reduces the dimensions of the features, and the Fully Connected Layer aims to classify features into several label classes.

CNN's development was accelerated by introducing new structures, optimization techniques, and the availability of large-scale datasets. The CNN structure is currently getting deeper and more complex [21]. The architecture that produces the lowest error rate in image classification was proposed in the ImageNet Large-Scale Visual Recognition Competition (ILSVR) [22]. Some of the winners in this competition are AlexNet, GoogLeNet (Inception), VGGNet, and ResNet. These architectures can be used for other classifications as a pre-trained model. Besides being used as a classification, it can be used for independent feature extraction, feature extraction integrated with other models, and weight initiation for new model inputs.

2. 1.1 Residual Net (ResNet)

Increasing the depth of the network does not work simply by stacking the layers together. In Deep Learning, the deeper the model, the more difficult the network is to train. Repeated multiplication makes the gradient very small (close to zero), and this phenomenon causes the vanishing gradient problem. As a result, as the network goes deeper, its performance becomes saturated or degrades rapidly. ResNet supports training in this deep network using the concept of skip connection (shortcut connection) [23]. The signal feeds into a layer above it and is added to the layer's output located slightly higher up the stack. The skip connections solve the problem of vanishing gradients by allowing this shortcut path for the gradient to flow through. Figure 1 shows an illustration of the skip connection concept.

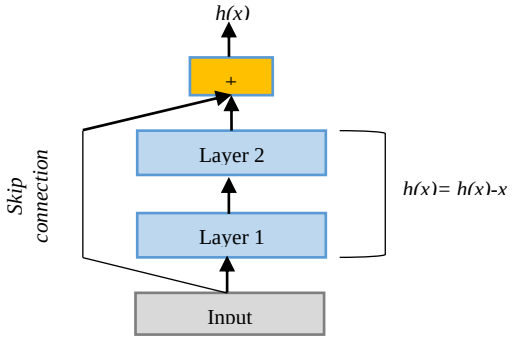


Figure 1 Skip connection on ResNet

2. 1.2 Inception

The idea behind Inception is how to make the network wider rather than deeper. Regular convolution operation extracts information from the previous layer using a kernel. Different kernel sizes extract different types of information, and Inception allows the network to select the most helpful information by using different kernel sizes simultaneously. It performs convolutional with three different filter sizes (1x1, 3x3, and 5x5) and max pooling. The outputs are concatenated and sent to the next layer. Extra 1x1 convolution was used before the 3x3 and 5x5 convolutions to make the computation cheaper. Figure 3 shows the Inception block architecture. The notation "Conv 3x3" means convolution using a 3x3 kernel. These four convolution layers can extract more complex patterns at various scales [24].

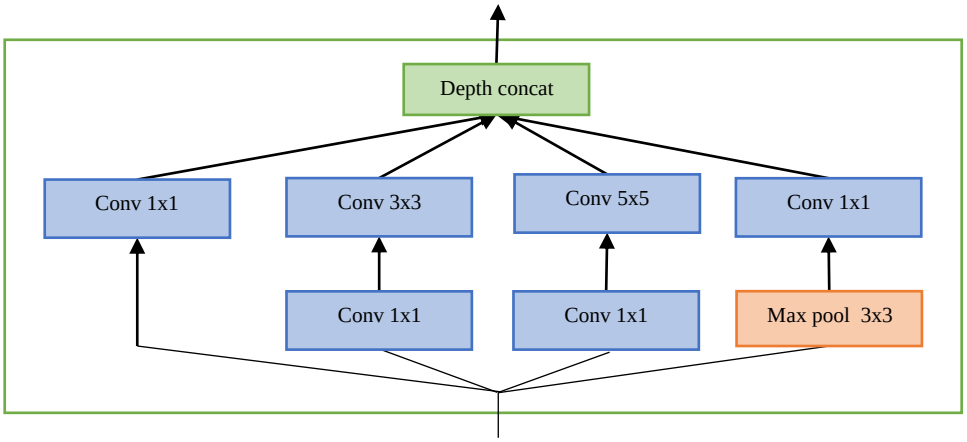


Figure 2 Architecture of Inception module

2. 1.3 Xception

Xception, which stands for "Extreme Inception," is a novel deep convolutional neural network architecture inspired by Inception, where depthwise separable convolutions replace Inception modules. Standard convolution performs the channel-wise and spatial-wise computation in one step, while depthwise separable convolutions split the process into two stages. A depthwise convolution applies a single convolutional filter per each input channel. Then a pointwise convolution is used to create a linear combination of the output of the depthwise convolution. Two differences between Xception and a depthwise separable convolution are the order of the operations and the presence or absence of a non-linearity. Depthwise separable convolutions perform first channel-wise spatial convolution and perform 1x1 convolution, whereas Xception performs the 1x1 convolution first. Depthwise separable convolutions are usually implemented without non-linearities, while a ReLU non-linearity follows Xception operations. Figure 3 shows Xception primary ide [25].

2. 2 YOLO

You Only Look Once (YOLO) is an object detection approach using a single neural network architecture to simultaneously predict bounding boxes and object classes from an entire image. This unified model has several advantages over traditional methods because of its speed and high precision. Unlike sliding windows and region proposal-based techniques, YOLO can see the whole image, thereby implicitly encoding contextual information about the class. This method divides the image into an S x S grid. If the center of an object falls into a grid cell, that grid cell is responsible for detecting the object. Each grid cell predicts as many B bounding boxes and a confidence score for each box. The bounding box consists of 5 predictions: *x, y, w, h*, and confidence. The coordinates (*x,y*) represent the center of the grid relative to the boundary of the grid cells, and the width and height are predicted relative to the entire image. Finally, the

IOU represents the confidence prediction between the prediction and truth boxes [26]. This study used YOLOV3, a hybrid approach between the network used in YOLOv2, Darknet-19, and the rest of the new model network. It has 53 convolution layers and is called Darknet-53. The YOLOV3 architecture is shown in Figure 4 [27].

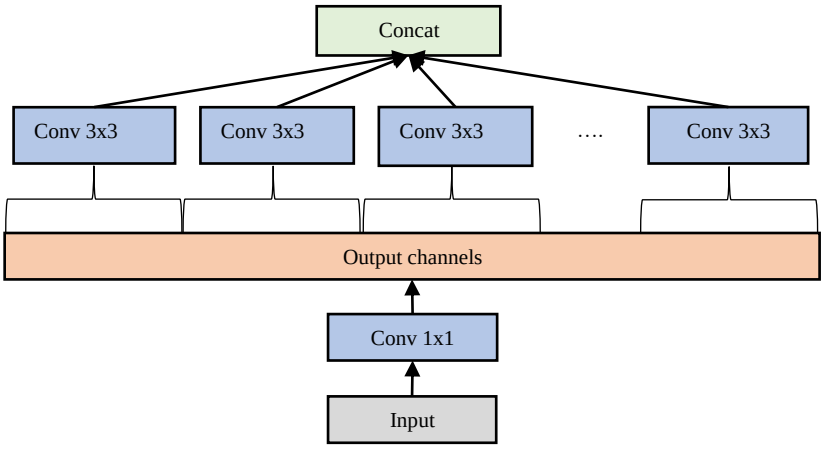


Figure 3 Xception module architecture

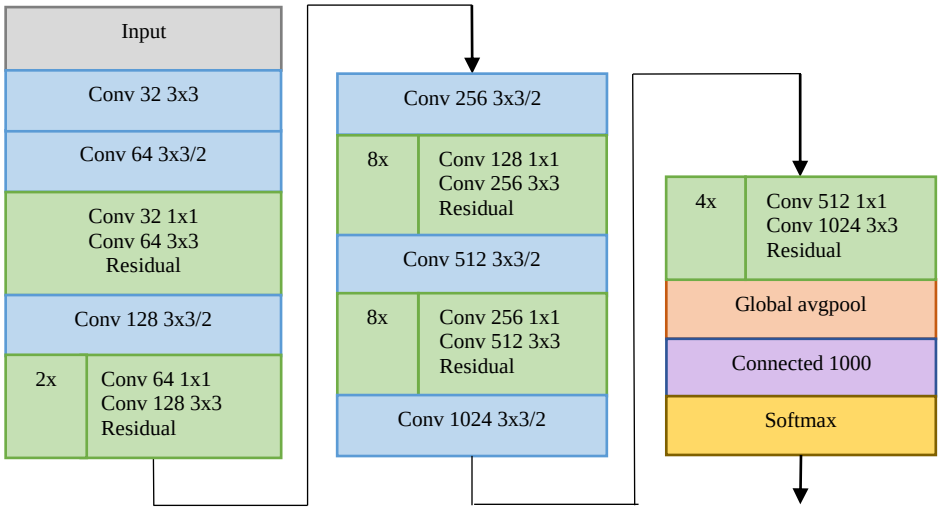


Figure 4 The YOLOV3 architecture

2. 3 Dataset and Pre Processing

We used the primary dataset taken in the office area using a webcam. We captured the image of masked and unmasked faces facing the camera with natural backgrounds and lighting. Everyone wears a different mask, reflecting the diversity in the real world. We took pictures of 28 people, and we have a dataset of 550 for mask detection and 1400 for face recognition. Each part is divided 80% and 20% respectively for training and validation.

We labeled images for mask detection training using LabelImg from Tzutalin, a graphic image annotation tool [28]. Image annotations are saved in the YOLO format. An array contains five elements: the object's label, *x* and *y* coordinate, length, and width of the bounding box. For face recognition, we manually labeled the images into 28 tags referring to the identity of the face. Figure 5 shows the dataset labeling using LabelImg.

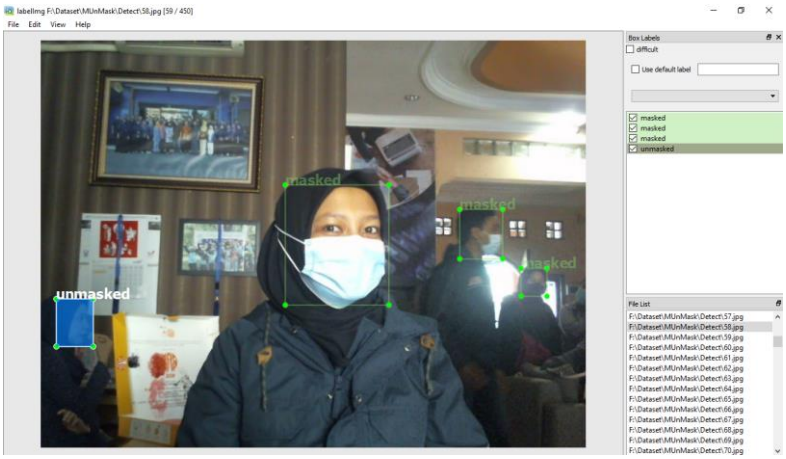


Figure 5 The dataset labeling using LabelImg for mask detection

2.4 Mask Detection and Face Recognition

This research divides into two stages, namely the mask detection stage and the face recognition stage. In the mask detection stage, we label the dataset with LabelImg. Then we divide the data into training sets and testing sets. Next, YOLOV3 is trained to get a mask detector. We then implemented the detector to the dataset for the face recognition stage. The detection results are in the form of a bounding box for the face area, predictions class, and confidence level. Areas of the detected face are labeled manually according to their identity. Next, we compared three models, namely ResNet152V2, InceptionV3, Xception, to extract faces, whether they were masked or not. The model's output is the prediction of the identity of the face. Next, we implement the model with the best performance for an unseen dataset. The expected result is that when the system sees an image, it will generate the output of the face, predict whether the face is masked or not, and predict face identity. Figure 6 shows the process of mask detection and face recognition.

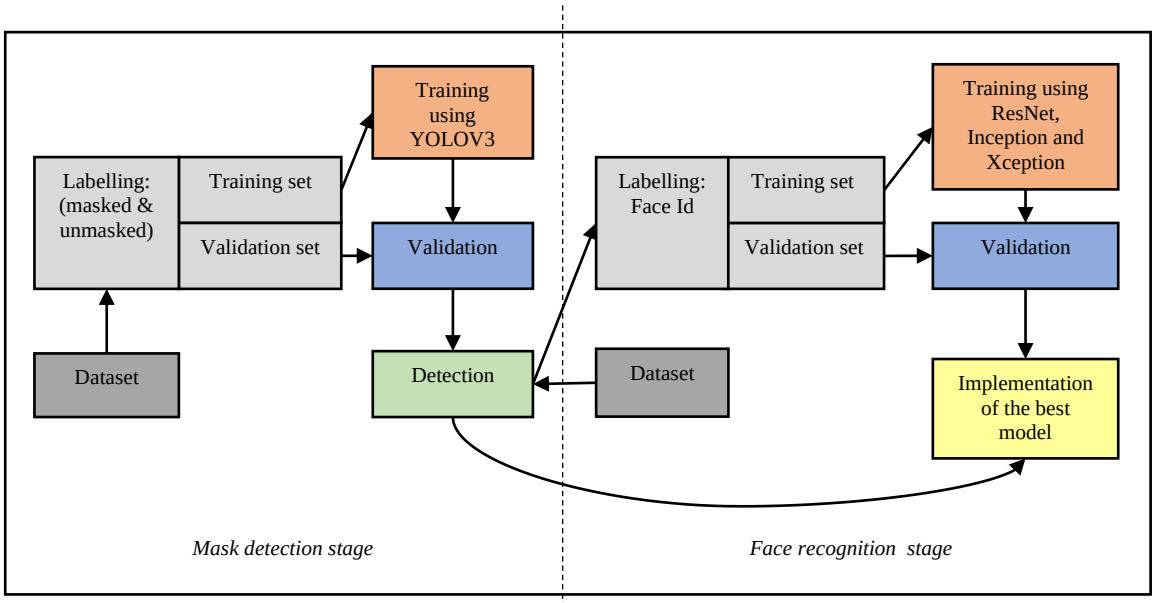


Figure 6 Mask detection and face recognition process

3. RESULTS AND DISCUSSION

3.1 Result of Mask Detection

We applied YOLOV3 on the provided dataset to localize and detect masked and unmasked faces. The training was carried out for two classes, batch=7000, filter=21, momentum=0.9, decay=0.0005, using darknet53.conv.74 pre-training weights. We implemented code with Google Colaboratory, CUDA (version: 11010), OpenCV (version: 3.2.0), and GPU: Tesla P100-PCIE-16GB. Table 1 shows the results of the mask detection training.

Table 1. Result of training and validation using YOLOV3

Dataset	MAP@0.50	Precision	Recall	F1-score	Average IoU
Training	0.8960	0.9903	0.9987	0.9945	0.8729
Validasi	0.8957	0.8802	0.8942	0.8950	0.7274

The detection results for training show the promised results with a Mean Average Precision (MAP) of 0.8960. There were 820 correct detections (320 unmasked and 500 masked), eight False Positives (2 unmasked and six masked), and one False Negative object. The final loss in the training process is 0.0672. The validation for the detection process resulted in a high MAP of 0.8957 but unsatisfactory Precision and Recall. In this process, 179 objects were correctly detected (68 unmasked and 101 masked), 23 False Positives (seven unmasked and 16 masked), and 20 False Negatives. This result may occur due to prediction errors on small objects, resulting in many detection errors, but the MAP validation is still relatively comparable with MAP in the training process.

Next, we apply the detector to the face recognition dataset. Figure 7 shows the samples of the detection results. Although YOLOV3 is a superior model in detecting small objects than other one-stage approaches [29], the result indicates that False Positive and False Negative is pretty high (Figure 7a and 7b). However, for the big objects (more than 100 pixels), the model resulted in promised results, even in relatively low light (Figure 7c). From 1400 images contain big size faces, only five were False Positive and one False Negative. At this process, 2631 objects were detected. The detection was performed in an average of 17 ms per image. In general, the YOLO object detection architecture is reliable for this mask detection task. The model promises a high inference speed and good performance even though we trained it on a relatively small data set. A more accurate face mask detection could be improved by adding a labeled dataset by spending more on human annotation [30].

There is a case that also appears in the detection result. An area detects two overlapping bounding boxes, and each shows a different class. We can overcome it by looking at the IoU between the two Bounding Boxes, determining the threshold of whether the bounding box does refer to the same objects or not. We can also choose the decision by reviewing the probability level of the detected class. In Figure 7d, there are three bounding boxes seen. The first is a masked class with a probability of 100 percent, and we have no problem with that area. The second and the third bounding boxes overlap with high IoU. Each shows a masked face with a probability of 92% (green box) and an unmasked face with a probability of 40% (purple box). In this case, since we have high IoU and one box has a significant probability, we can eliminate the box with a small probability.

Figure 7 Result of masked and unmasked face detection using YOLO

3.2 Result of Face Recognition

We apply the detector to the face recognition dataset to get masked and unmasked faces. We then labeled this dataset as a material for building a face recognition model. We combine masked and unmasked faces in a single training, so the model must fit one label in two face conditions (masked or unmasked). Next, we conduct training for face recognition using the three pre-trained models. Each model was used as a feature extractor. We add a flatten layer and two dense layers for classification. We use the Softmax function for the output layer, Adam for the optimizer, and the Categorical Cross Entropy as the loss function. The training run for 50 epochs with batch size = 32 and learning rate = 0.001. We tried each model 10 times. We only identify faces above 100 pixels and then resize the resulting image to 128x138. We used Xception, variations of ResNet152V2 and InceptionV3, which are best in their class in top-1 and top-5 accuracy on the ImageNet. We did not change the parameters for each model from Keras Applications so that each model uses the default settings. Figure 8 shows the sample results of face recognition training, and Table 2 shows the comparison results for three models.

Table 2 shows that all models can perform classification well; each can achieve 100% accuracy in the best model. ResNet152V2 is the model with the lowest average loss on validation by 0.0093. Unfortunately, ResNet152V2 is a model with extensive parameters that require considerable computation time. Therefore, especially for real-time applications, we can consider the Xception model with a loss value and accuracy that competes with ResNet152V2 but has fewer parameters.

We also review the loss and accuracy charts for the three models. We get the fact that in all models, there was instability in the loss and accuracy values. Therefore we reduce the learning rate to 0.0001. This change turned out to have a significant impact where all models could reduce their loss values stably. This change also makes ResNet152V2 and Xception has an average accuracy of 100% in training and validation, although the loss value in epoch tends

to be smaller by 0.08063 for ResNet152V2 and 0.09157 for Xception. Figures 9 show the best loss values and accuracy graphs in the three-value model with a two-value learning rate.

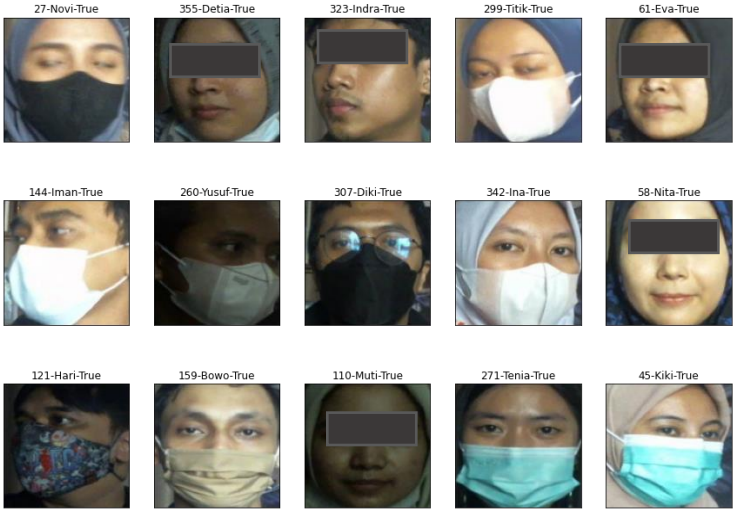
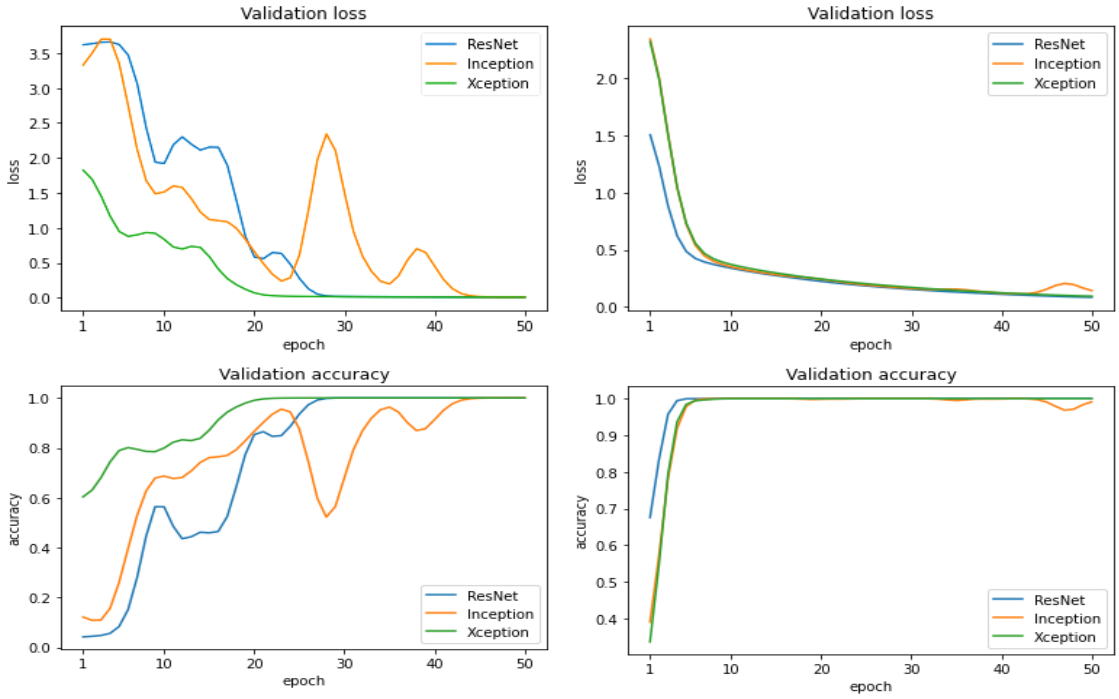


Figure 8 The results of face recognition training

Table 2 Comparison of three models in the recognition of masked and unmasked faces

Model	**Number of Parameters**	**Avg time/ step (*ms*)**		**Training**		**Validation**	
				Loss	**Accuracy**	**Loss**	**Accuracy**
ResNet152V2	60,430,684	199	Avg	**0,0113**	**0,9990**	**0,0093**	**0,9995**
			Best	0,0103	0,9990	0,0072	1,0000
InceptionV3	22,328,956	88	Avg	0,0335	0,9929	0,1897	0,9674
			Best	0,0224	0,9980	0,0089	1,0000
Xception	22,960,516	142	Avg	0,0219	0,9960	0,0410	0,9917
			Best	0,0107	0,9990	0,0063	1,0000



(a) $LR = 0.001$ (b) $LR = 0.0001$

Figures 9 The loss values and accuracy

The fundamental advantage of YOLO is that there is no dependence on a pre-existing face classifier. Therefore, especially for limited environments, YOLO can be used for mask detection and recognition of the identity behind the masked face. So we need to assign the identity of the face directly in the YOLO labeling process. With this strategy, detectors and identifiers are combined into a single architecture. We only need to do a single training for localization, mask detection, and face recognition behind masks.

(a) (b)

(c) (d)

Figure 10 Implementation of detection and recognition masked and unmasked face

4. CONCLUSIONS

We designed a system that can implement mask detection and face recognition for natural environment datasets. We implemented YOLOV3 for mask detection, and in general, it can localize bounding boxes and their classes with good performance. Adjustment parameters may be required to increase the Precision and Recall in this process. We performed a comparison of face identification models, and we chose Xception as an high-performance and fast model. This excellence is required for real-time applications. Face recognition results deliver the promised results and identify faces, even those behind the mask. However, our research is still limited to detecting faces with a size of 100 pixels and above. In future research, mask detection and face recognition can be integrated, especially in a limited environment; thus, we only need one Deep Learning architecture for all tasks.

ACKNOWLEDGEMENTS

Authors would like to thanks Lembaga Penelitian dan Pengabdian Masyarakat (LPPM) Universitas Amikom Purwokerto that have provided financial support to this research.

REFERENCES

[1] E. Noyes, J. P. Davis, N. Petrov, K. L. H. Gray, and K. L. Ritchie, "The effect of face masks and sunglasses on identity and expression recognition with super-recognizers and typical observers," *R. Soc. Open Sci.*, vol. 8, no. 3, pp. 1–18, 2021. https://doi.org/10.1098/rsos.201169

[2] E. Freud, A. Stajduhar, R. S. Rosenbaum, G. Avidan, and T. Ganel, "The COVID-19 pandemic masks the way people perceive faces," *Sci. Rep.*, vol. 10, no. 22344, pp. 1–8, 2020. https://doi.org/10.1038/s41598-020-78986-9

[3] M. Marini, A. Ansani, F. Paglieri, F. Caruana, and M. Viola, "The impact of facemasks on emotion recognition, trust attribution and re-identification," *Sci. Rep.*, vol. 11, no. 5577, pp. 1–14, 2021. https://doi.org/10.1038/s41598-021-84806-5

[4] C. Gupta and Nasib Singh Gill, "Coronamask: A Face Mask Detector for Real-Time Data," *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 9, no. 4, pp. 5624–5630, 2020. https://doi.org/10.30534/ijatcse/2020/212942020

[5] M. Sujaritha, S. Kabilan, M. Manikandan, and S. N. Kisore, "Real Time Face Mask Identification Using Deep Learning," *J. Phys. Conf. Ser.*, vol. 1916, no. 012077, pp. 1–10, 2021. https://doi.org/10.1088/1742-6596/1916/1/012077

[6] S. Ge, J. Li, Q. Ye, and Z. Luo, "Detecting Masked Faces in The Wild with LLE-CNNs," *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 426–434, 2017. https://doi.org/10.1109/CVPR.2017.53

[7] 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*, vol. 167, 2020. https://doi.org/10.1155/2021/4529107

[8] R. Katari, Sreekar Kaza, B. RamyaSree, V. Divyavani, and M. AbubakarJ, "A Comparative Analysis of Variant Deep Learning Models for COVID-19 Protective Face Mask Detection," *Turkish J. Comput. Math. Educ.*, vol. 12, no. 6, pp. 2841–2848, 2021. https://doi.org/10.17762/turcomat.v12i6.5791

[9] W. Hariri, "Efficient masked face recognition method during the COVID-19 pandemic," *Signal, Image Video Process.*, 2021. https://doi.org/10.1007/s11760-021-02050-w

[10] A. S. Joshi, S. S. Joshi, G. Kanahasabai, R. Kapil, and S. Gupta, "Deep Learning Framework to Detect Face Masks from Video Footage," in *Proceedings - 2020 12th International Conference on Computational Intelligence and Communication Networks, CICN 2020*, 2020, pp. 435–440. https://doi.org/10.1109/CICN49253.2020.9242625

[11] 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, no. 103848, 2021. https://doi.org/10.1016/j.jbi.2021.103848

[12] M. S. Ejaz and M. R. Islam, "Masked face recognition using convolutional neural network," in *2019 International Conference on Sustainable Technologies for Industry 4.0, STI 2019*, 2019, no. December 2019. https://doi.org/10.1109/STI47673.2019.9068044

[13] J. S. Talahua, J. Buele, P. Calvopina, and J. Varela-Aldas, "Facial recognition system for people with and without face mask in times of the covid-19 pandemic," *Sustain.*, vol. 13, no. 12, pp. 1–19, 2021. https://doi.org/10.3390/su13126900

[14] Y. Said, "Pynq-YOLO-Net: An embedded quantized convolutional neural network for face mask detection in COVID-19 pandemic era," *Int. J. Adv. Comput. Sci. Appl.*, vol. 11, no. 9, pp. 100–106, 2020. https://doi.org/10.14569/IJACSA.2020.0110912

[15] S. Abbasi, H. Abdi, and A. Ahmadi, "A Face-Mask Detection Approach based on YOLO Applied for a New Collected Dataset," in *26th International Computer Conference, Computer Society of Iran, CSICC 2021*, 2021, pp. 1–6. https://doi.org/10.1109/CSICC52343.2021.9420599

[16] B. Roy, S. Nandy, D. Ghosh, D. Dutta, P. Biswas, and T. Das, "MOXA: A Deep

Learning Based Unmanned Approach For Real-Time Monitoring of People Wearing Medical Masks," *Trans. Indian Natl. Acad. Eng.*, vol. 5, no. 3, pp. 509–518, 2020. https://doi.org/10.1007/s41403-020-00157-z

[17] A. Kumar, A. Kalia, A. Sharma, and M. Kaushal, "A hybrid tiny YOLO v4-SPP module based improved face mask detection vision system," *Journal of Ambient Intelligence and Humanized Computing*. 2021. https://doi.org/10.1007/s12652-021-03541-x

[18] J. Yu and W. Zhang, "Face Mask Wearing Detection Algorithm Based on Improved YOLO-v4," *Sensors*, vol. 21, no. 3263, 2021. https://doi.org/10.3390/ s21093263

[19] N. M. Aszemi and P. D. D. Dominic, "Hyperparameter Optimization in Convolutional Neural Network using Genetic Algorithms," *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no. 6, pp. 269–278, 2019. https://doi.org/10.14569/ijacsa.2019.0100638

[20] S. Loussaief and A. Abdelkrim, "Convolutional Neural Network Hyper-Parameters Optimization based on Genetic Algorithms," *Int. J. Adv. Comput. Sci. Appl.*, vol. 9, no. 10, pp. 252–266, 2018. https://doi.org/10.14569/IJACSA.2018.091031

[21] H. Choi, "CNN Output Optimization for More Balanced Classification," *Int. J. Fuzzy Log. Intell. Syst.*, vol. 17, no. 2, pp. 98–106, 2017. https://doi.org/10.5391/IJFIS.2017.17.2.98

[22] Aurélien Géron, "Deep Computer Vision Using Convolutional Neural Network," in *Hands-On Machine Learning with Scikit-Learn, Keras & tensorFlow*, 2nd ed., Sebastopol: O'Reilly Media, Inc., 2019, pp. 445–496.

[23] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. December, pp. 770–778, 2016. https://doi.org/10.1109/CVPR.2016.90

[24] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016, vol. December, pp. 2818–2826. https://doi.org/10.1109/CVPR.2016.308

[25] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017, vol. 2017-Janua, pp. 1800–1807. https://doi.org/10.1109/CVPR.2017.195

[26] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 779–788, 2016. https://doi.org/10.1021/je00029a022

[27] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," *arXiv:1804.02767v1*, pp. 1–6, 2018. https://pjreddie.com/media/files/papers/YOLOv3.pdf

[28] Tzutalin, "LabelImg (Git code)." 2015.

[29] N. Nguyen, T. Do, T. D. Ngo, and D. Le, "An Evaluation of Deep Learning Methods for Small Object Detection," *Hindawi*, 2020. https://doi.org/10.1155/2020/3189691

[30] Y. Ding, Z. Li, and D. Yastremsky, "Real-time Face Mask Detection in Video Data," *arXiv:2105.01816*, 2021. http://arxiv.org/abs/2105.01816