Faster Region-based Convolutional Neural Network for Mask Face Detection

Indah Agustien Siradjuddin
Informatics Department
Faculty of Engineering
University of Trunojoyo Madura
Telang raya Street, Kamal, Madura
East Java Province, Indonesia
indah.siradjuddin@trunojoyo.ac.id

Reynaldi
Informatics Department
Faculty of Engineering
University of Trunojoyo Madura
Telang raya Street, Kamal, Madura
East Java Province, Indonesia

Arif Muntasa
Informatics Department
Faculty of Engineering
University of Trunojoyo Madura
Telang raya Street, Kamal, Madura
East Java Province, Indonesia

2021 5th International Conference on Informatics and Computational Sciences (ICICoS) | 978-1-6654-3807-0/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICICOS53627.2021.9651744

Abstract— **We present two-stage detection approach, Faster Region-based Convolutional Network, Faster R-CNN for masked face detection. In this face detection, we localize the face object in the image and classify the face object based on mask occlusion on the face. There are three classes: a face without any mask, the second is an incorrectly masked face, and the third is a face with correctly masking. The first stage of the detection is finding the candidate regions of the target object. This stage uses the Region Proposal Network (RPN). Then, the candidate regions are fed into the last pooling layer of the Faster R-CNN identified as the ROI Pooling layer. The model is trained using MAFA and AFLW datasets. The mean Average Precision of trained model for all classes is 0.73, with the highest accuracy is obtained by the face without mask class, and the lowest accuracy is the incorrectly masked face class.**

Keywords— Masked Face Detection;Faster R-CNN; Region Proposal Network; ROI Pooling

I. Introduction

For the last couple of years, the covid-19 pandemic has been spread almost all over the world. As a result, there are health protocols that we must obey to prevent the spread of the corona virus. Properly wearing a mask is one of the protocols, especially when we are in public spaces. Therefore, it would be advantageous if a system could automatically detect the person who does not put the mask on the face correctly or even wears any mask.

For that purpose, we built and trained the mask face detection model based on the Convolutional Neural Networks (CNN) architecture since CNN architecture has several advantages. For instance, feature learning ability [1]–[3]. With the feature learning ability from the convolutional layers, an image (a raw image) can be used as an input for the CNN architecture.

Mask face detection using faster R-CNN was proposed previously by Shilaja et al. in [4]. However, this research's detection model is based on transferred learning and showing detects faces with masks only. We use the faster R-CNN (Region-based Convolution Neural Network) architecture also to localize and classify mask faces into three classes, i.e., a face without a mask (a normal face), improper mask on face (incorrectly mask face), and properly wearing a mask (correctly mask face). The three main contributions of this article are, first, creating the required dataset (three classes) taken from MAFA (Masked Face) and AFLW (Annotated Face Landmarks in the Wild). Second, we used the dataset to build and train the Faster R-CNN architecture, including the RPN (Region Proposal Network). Third, our proposed model classifies the faces into three classes (a normal face, an incorrectly mask face, and a correctly mask).

II. Related Works

The proposed masked face detection model uses a two-stage approach that consists of two stages. The first is to find candidate regions for face location or recognized as the region proposal module. The second stage localizes and classifies each region proposal from the first stage with the CNN architecture. We apply Faster Region-based Convolutional Neural Network (R-CNN) [5], part of the R-CNN family, for the detection model.

The first R-CNN family for the detection model is R-CNN itself. For the first stage, i.e., the region proposal, R-CNN uses the selective search module. This module finds candidate regions from an image based on the image segmentation process, The second stage, CNN architecture will determine whether an object within a region is a target object [6] Since CNN architecture extracts feature maps from each candidate region, the R-CNN model requires a high computational cost. Therefore fast R-CNN is proposed in [7].

The Fast-RCNN model shares the feature maps from the convolutional layers. Therefore, feature extraction for each region from the selective search module is not required. In addition, the fast-RCNN model uses the ROI (Region of Interest) pooling layers instead of the pooling layer [7]. Thus, the sharing scheme reduces the computational cost. However, since the selective search is used as the proposal module, the fast R-CNN model still requires more time in the region proposal stage.

The Faster-RCNN introduces the RPN (Region Proposal Network) to reduce the time for finding candidate regions [5]. Therefore, we create our model to localize and classify masked faces based on this architecture. The remaining article is organized as follows, section 3 describes the Faster R-CNN architecture, including the RPN module and the ROI Pooling mechanism. In section 4, we present our result and discussion, and the final is the conclusion section.

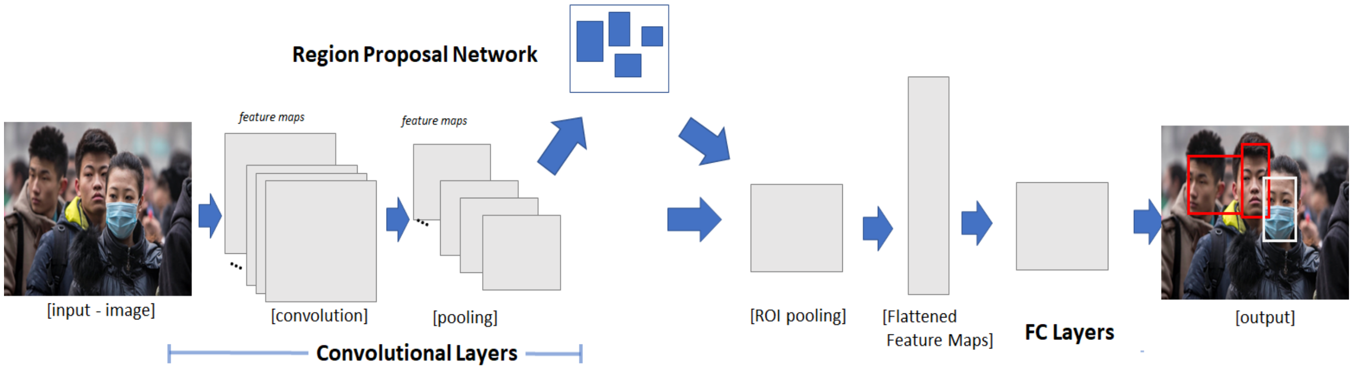


Fig. 1. Faster R-CNN architecture with RPN as the proposal module

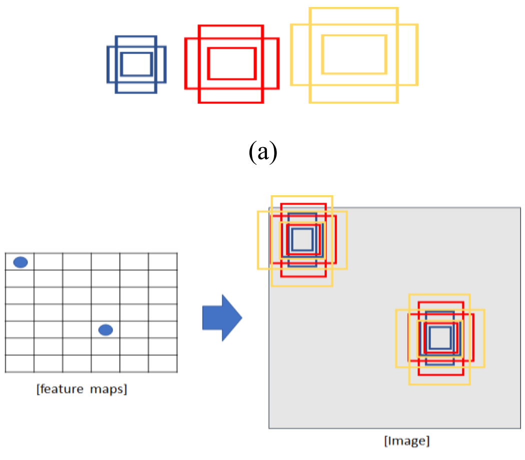
III. Faster Region-Based Convolutional Neural Network (Faster R-CNN)

A. *Faster R-CNN architecture*

As in R-CNN and Fast R-CNN, the Faster R-CNN consists of two stages: finding the proposal region, and the second is to detect (localize and classify) the target object. However, unlike the R-CNN and Fast R-CNN use the selective search module, Faster-RCNN uses a deep fully convolutional network to find the candidate region [5], i.e., the Region Proposal Network (RPN). The architecture of the Faster R-CNN is shown in Fig.1. The Faster R-CNN architecture has two highlights: the regional proposal module with RPN, ROI pooling layers, and classification layers. Meanwhile, feature maps from the convolutional layers (feature learning layers) are obtained using the weights from convolutional layers of VGG-16 [8].

B. *Region Proposal Network (RPN)*

We use trainable VGG-16 in the convolutional layers of Faster R-CNN. Feature maps from the last layer in the convolutional layers are used as an input to the RPN. Every point in feature maps plays as an anchor point for generating region proposals as seen in Fig. 2. We have nine bounding boxes based on three different ratios and three sizes. The ratios are $1:1,\ {}^{1}/_{\sqrt{2}}:{}^{2}/_{\sqrt{2}},\ {}^{2}/_{\sqrt{2}}:{}^{1}/_{\sqrt{2}}.$ Meanwhile the sizes are, 64, 128, and 256. Blue, red bounding boxes, and yellow bounding boxes from Fig.2 (a) are the bounding box from size 64, 128, and 256 respectively.



(a)

(b)

Fig. 2. Generating Candidate Proposals in RPN, (a) variant bounding boxes, (b) generate candidate proposals in an image from an anchor point

Two outputs from our trained RPN, i.e., output class and regression (the modified location of bounding box). To create target class label for the RPN, we calculate IOU (Intersection Over Union) of each bounding boxes with the ground truth. If the IOU greater than the threshold, then the target class is 1, otherwise is zero.

C. *ROI Pooling*

ROI (Region of Interest) Pooling layer firstly introduces on Fast-RCNN model. The pooling operation (maximum or average pooling) requires the candidate region's information location (the bounding box coordinate). The process in the ROI pooling layer is illustrated in Fig. 3.

The ROI coordinates from an image are transformed into feature maps coordinates using (1) and (2). We built our Faster R-CNN with n ROI with $2\ x\ 2$ ROI size. Since we have m feature maps, the ROI Pooling layers produce $m\ x\ n\ x\ 2\ x\ 2$ extracted features. These features are then flattened for the fully connected layers.

$$x' = x \times \frac{featureMap\_width}{image\_width} \qquad (1)$$

$$y' = y \times \frac{featureMap\_height}{image\_height} \qquad (2)$$

Flattened feature maps from ROI pooling are fed into Fully Connected layers (FC layers). This layers in the Faster R-CNN plays as the detector i.e., localize and classify the object within the region. Therefore, there are two kinds of output in this FC Layers, i.e., output class and regression. Since we classify three kinds of mask usage in a face, they are, normal face (face without any mask), incorrectly masked face (the mask is use in the proper way), and correctly masked face. Therefore, we have three neurons for the output class. Meanwhile in the regression, we have four locations of the bounding boxes.

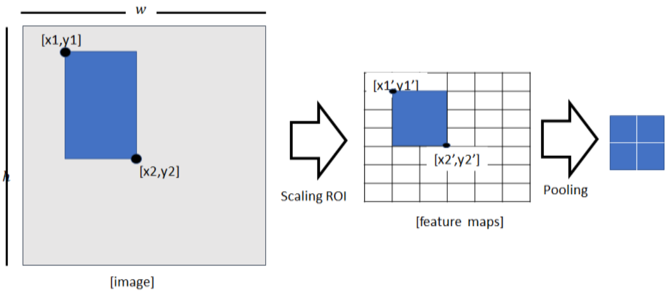


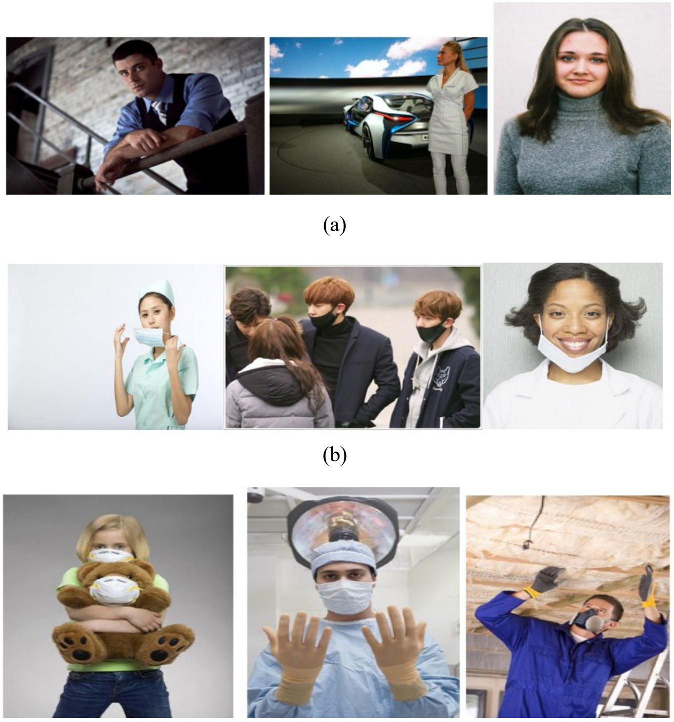
Fig. 3. ROI Pooling

IV. Result And Discussion

We trained our architecture using Masked Face (Mafa dataset) [9], and Annotated Facial Landmarks in the Wild (AFLW dataset) [10]. Mafa dataset provides faces with different mask occlusion degrees. The degree is ranged from 1 up to 3, and the lower degree values indicate the mask is incorrectly put on faces. Since we detect and classified masked faces into three classes (i.e., a face without mask or a normal face, incorrectly masked face, correct masked face), we use faces with occlusion degrees 1 and 2 from MAFA as an incorrectly masked face class. Meanwhile, face with occlusion degree 3 as a correctly masked face class. For a normal face (face without mask), we use the AFLW dataset for training. The images example of our three classes (taken from MAFA and AFLW) can be seen in Fig. 4.

In our experiment, the total images from the MAFA dataset are 25876 and from AFLW are 14587 images. However, the number of target faces (normal, incorrectly masked face, and correct masked face) in each image are variated. Most of the images have one face object, and few have more than one face object. Thus, each class's total face object from those images is 17019 for normal face, 3842 for incorrectly masked face, and 25610 for a correct masked face. We tested our detection model with 2713 images with 3000 face objects where there are 1000 face objects in each class.

The example of detected faces in the testing phase can be seen in Fig. 5 and Fig 6. Fig. 5 shows there is only one face object in each image. Moreover, Fig 6 shows there is more than one face object in each image. Meanwhile, the example for false detection from our model can be seen in Fig. 7. In Fig. 7, some objects are missed detection, and some objects are falsely classified. Precision-Recall Curve of our face mask detection model Fig. 8 (Face without mask), Fig. 9 (incorrect masked face), and Fig.10 (correct masked face).



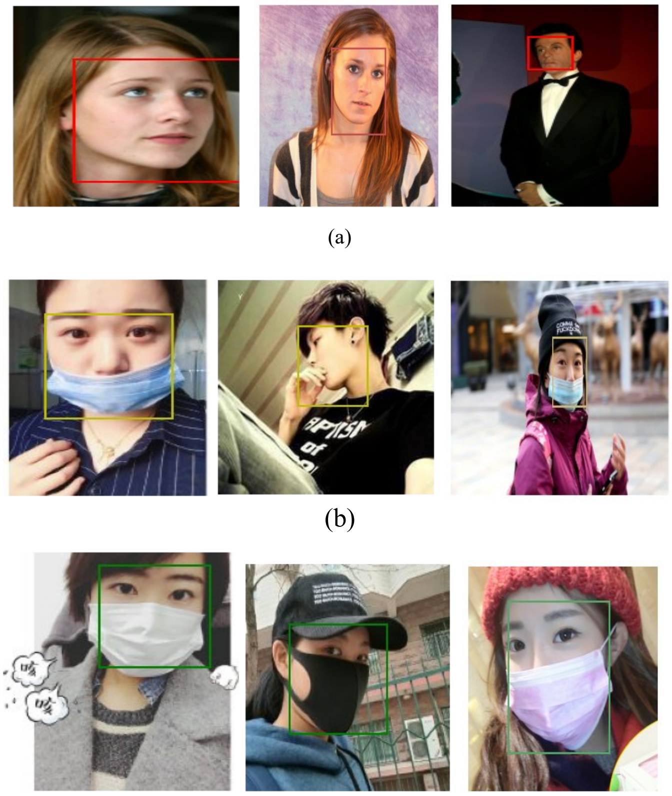
(a)

(b)

(c)

Fig. 4. Face Detection and Classification Dataset. (a) images for face without a mask (normal face), (b) images for Incorect Masked Face (improper way of putting on mask), and (c) images for Correct Mask Face (properly mask wearing).

The figures show that the incorrect masked face Precision-Recall has the lowest decreasing curve, and the table shows incorrect masked face has the lowest mAP. The lowest performance of the incorrect masked face class since there is imbalanced data in training. For incorrect masked face class, the dataset only provides 8% of all data. Therefore, the number of images for this class is not sufficient for the training phase.

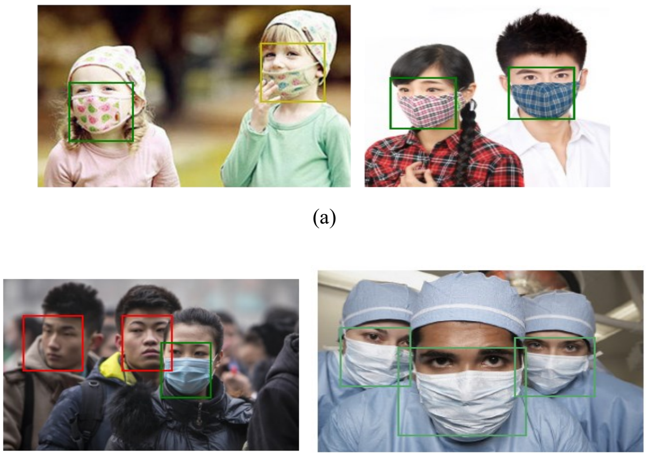


(a)

(b)

(c)

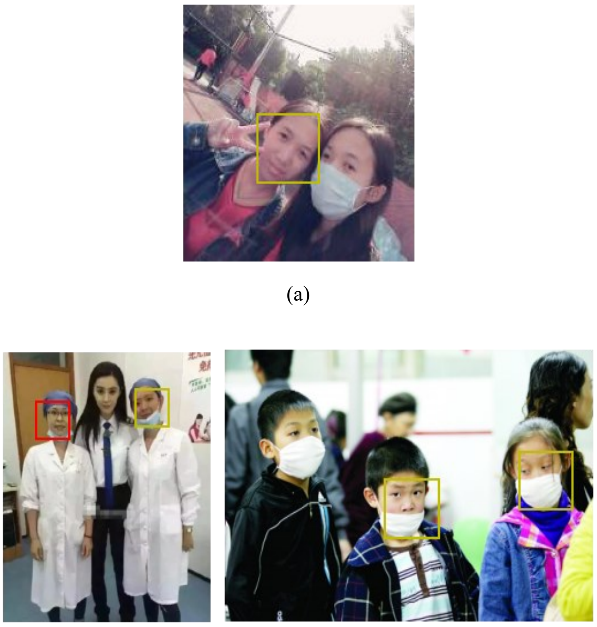
Fig. 5. Testing Result with one face object, red rectangle indicates no mask face (normal face), yellow rectangle indicate incorect masked face, and the green one is the correct masked face. (a) images with one normal face, (b) images with one incorectly mask face, and (c) images with one correctly mask face.



(a)

(b)

Fig. 6. Testing Result with more than one face object in the images, red rectangle indicates no mask face (normal face), yellow rectangle indicate incorect masked face, and the green one is the correct masked face. (a) images with two faces, and (b) image with three faces



(a)

(b)

Fig. 7. Testing result for false detection, red rectangle indicates no mask face (normal face), yellow rectangle indicate incorect masked face, and the green one is the correct masked face. (a) image with two faces, and (b) images with three faces

We compared our proposed face detection model based on Faster R-CNN architecture with Fast R-CNN architecture for the experiments. Table 1 shows the mAP (Mean Average Precision) of both models. As seen in the table, our proposed model achieved higher performance accuracy. The lower accuracy of Fast R-CNN is mainly because of the regional proposal stage. In the Faster R-CNN, the regional proposal stage uses the deep network. Hence, the classification layer receives only specific information, i.e., the location of the target object.

Meanwhile, in the Fast R-CNN, the selective search is used for the regional proposal stage. This algorithm is based on image segmentation. Therefore, the classification layer receives general information on many locations of the candidate regions. These candidate regions result from the segmentation process, and it can be the location of other objects (not always the target objects).

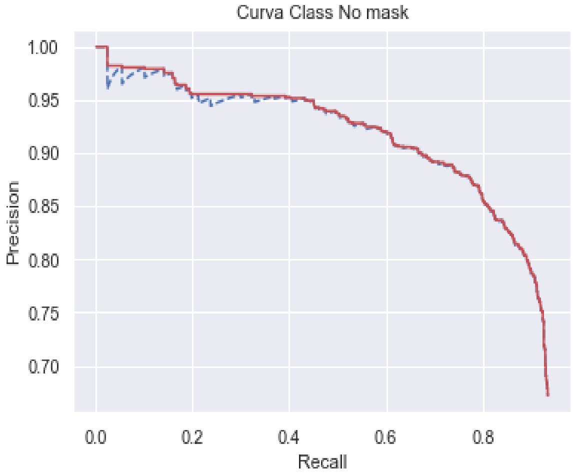


Fig. 8. Precision Recall Curve for Normal Face (Face without mask)

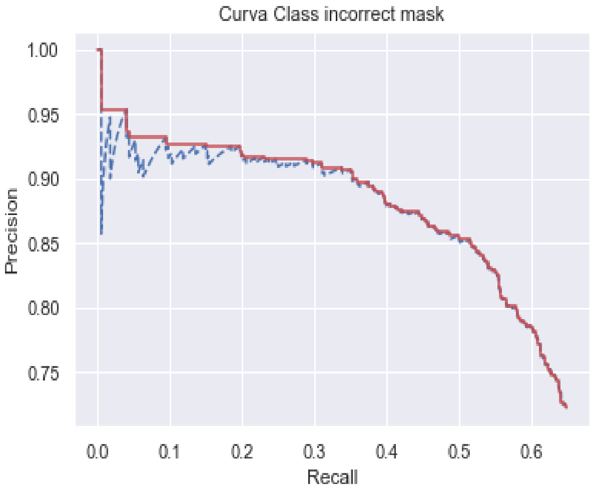


Fig. 9. Precision Recall Curve for Incorect Masked Face

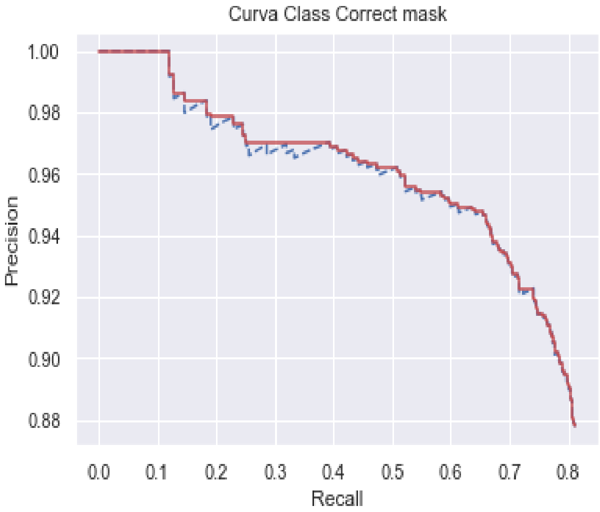


Fig. 10. Precision Recall Curve for Correct Masked Face

TABLE I. AVERAGE PRECISION IN EACH CLASS

Class	**Average Precision - Faster R-CNN**	**Average Precision - Fast R-CNN**
Normal Face / Without Mask	0.86	0.52
Correct Masked face	0.78	0.47
Incorrectly Masked face	0.57	0.17
mAP	**0.73**	**0.39**

V. CONCLUSION

Faster R-CNN is implemented to localize and classify masked face in this article. We find the location of the face and determine whether the detected face is using the proper mask. Three classes in the proposed models, they are normal face, correct face, and incorrectly masked face. The highest average precision is 0.86 from detecting normal face, meanwhile the lowest precision is 0.57 from incorrectly masked face. This result is obtained since the dataset for training from this class is difficult to get, therefore there is an imbalanced data in the training process. For future research, augmenting and resampling data will be a consideration.

Acknowledgment

The authors would like to thank to the University of Trunojoyo Madura, Institute of Research and Community Service who has supported this research under Research Grant Number: 3068/UN46.4.1/PT.01.03/2021.

References

[1] S. Hijazi, R. Kumar, and C. Rowen, *Using Convolutional Neural Networks for Image Recognition*. 2015. Accessed: Jun. 12, 2017. [Online]. Available: http://www.multimediadocs.com/assets/cadence_emea/documents/using_convolutional_neural_networks_for_image_recognition.pdf

[2] H. Khalajzadeh, M. Mansouri, and M. Teshnehlab, "Face recognition using convolutional neural network and simple logistic classifier," in *Soft Computing in Industrial Applications*, Springer, 2014, pp. 197–207. Accessed: Jun. 12, 2017. [Online]. Available: http://link.springer.com/chapter/10.1007/978-3-319-00930-8_18

[3] I. A. Siradjuddin, *Kecerdasan Komputasional dan Aplikasinya dengan Menggunakan Python | Graha Ilmu.id*. Teknosains, 2018. Accessed: Nov. 30, 2019. [Online]. Available: http://grahailmu.id/product/kecerdasan-komputasional-dan-aplikasinya-dengan-menggunakan-python/

[4] S. H N, L. H N, P. H N, and B. Uma, "Detection and Localization of Mask Occluded Faces by transfer learning using Faster RCNN," *SSRN Journal*, 2021, doi: 10.2139/ssrn.3835214.

[5] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *arXiv:1506.01497 [cs]*, Jun. 2015, [Online]. Available: http://arxiv.org/abs/1506.01497

[6] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, OH, USA, Jun. 2014, pp. 580–587. doi: 10.1109/CVPR.2014.81.

[7] R. Girshick, "Fast R-CNN," *arXiv:1504.08083 [cs]*, Apr. 2015, [Online]. Available: http://arxiv.org/abs/1504.08083

[8] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. [Online]. Available: http://arxiv.org/abs/1409.1556

[9] "Masked Face Data." https://www.kaggle.com/rahulmangalampalli/mafa-data

[10] "AFLWData." https://www.tugraz.at/institute/icg/research/teambischof/lrs/downloads/aflw/