Proceedings of the Sixth International Conference on Research in Intelligent and Computing pp. 71–76
DOI: 10.15439/2021R13
ISSN 2300-5963 ACSIS, Vol. 27

Densely Populated Regions Face Masks Localization and Classification Using Deep Learning Models

Anh Pham-Hoang-Nam
School of Computer Science and Engineering
International University
Ho Chi Minh City, Vietnam
phna0220@gmail.com

Linh Phung-Khanh
School of Computer Science and Engineering
International University
Ho Chi Minh City, Vietnam
phungkhanhlinh.iu@gmail.com

Vi Le-Thi-Tuong
School of Computer Science and Engineering
International University
Ho Chi Minh City, Vietnam
lttvi1822@gmail.com

Nga Ly-Tu
School of Computer Science and Engineering
International University
Ho Chi Minh City, Vietnam
ltnga@hcmiu.edu.vn

***Abstract*—Over the last year, the correct wearing of facial masks in public is still a relevant matter in the fight against the COVID-19 pandemic. A popular approach that helps regulate the situation by global researchers is building smart systems for face mask detection. Following such spirit, this paper will contribute to the literature in two main aspects:**
(1) We first propose a new face mask detector model using the state-of-the-art RetinaFace for face localization in populous regions and the ResNet50V1 classifier to group the faces under 3 categories: correctly-worn, incorrectly-worn and no-masks-worn. (2) In order to select the ResNet50V1 as the backbone for the final model, we also analyzed its performance in accordance with another 3 classifiers on a face mask dataset beforehand. Performance metrics from the test phase have shown that our detector achieved the best accuracy among all the works compared, with $94,59\%$ on one test dataset and a less satisfactory 69.6% on another due to certain characteristics of the set. The code is available at: https://github.com/barbatoz0220/Densely-populated-FMD.git

***Index Terms*—Dense Population Regions, Face Mask, Localization, Classification, Covid-19, Deep Learning, MobileNet, ResNet, AIZOO, Neuralet**

I. Introduction

Unexpected as it may seem, the discovery of the first COVID-19 case has tragically started a series of ongoing depressive episodes for many people across the world, while at the same time presenting a global reordering moment in many aspects [1]. The pandemic has been listed as an extreme global crisis when the 1.4 million infected cases in April 2020 [2] have risen to more than 250 million [3]. These statistics would have been worse had it not been for the intensive implementation and conformance to many suggested preventive measures [2], from which the subject of our research - the correct wearing of face masks in public - is withdrawn since it is the most recommended, widely applicable and highly effective in reducing transmission rate with or without the implementation of other intervention methods [4].

While there exists a vast body of other technologies that are being utilized to help relieve the situation, within the scope of our research, we will solely focus on the use of Deep Learning in the field of Face Recognition to detect correct face-masks-wearing in public. Following such premise, the technology has proven to remain a trending topic as the latest review of Wang and Deng has introduced and discussed thoroughly the past, present and future various concepts as well as researches [5]. Even more so, with respect to growing concerns regarding the COVID-19 global pandemic up to date, the technology has definitely gathered enough traction and gained significant interest when research teams around the world rushed to develop and propose Deep Learning models to help protect the health of public communities [6]. Whether the degree of time was back in the early stages around which the outbreak occurred or varied through the current year, many research papers were studied and published in dedication to the means of carefully monitoring the usage of facial masks in public place [7]. For example, a three-component model has been proposed by Loey et al. in their research to supervise the wearing of medical face masks in public [8]. By using YOLO-v2 with ResNet-50, the hybrid model was then exclusively trained on 2 datasets featuring medical face masks and achieved a higher average precision rate (81%) in comparison to one of its related works' model, which was trained on a different dataset with mixed types of face masks (76.1%). Another approach from Loey et al. is to focus on the detection of people who are not wearing face masks in public [9]. Using ResNet-50 combined with Support Vector Machine

This research is funded by International University, VNU-HCM under grant number SV2020-IT-03.

Fig. 1: Face Mask Detection Sequential Model

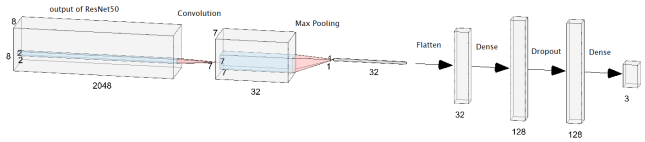

Fig. 2: Architecture of Our Classification Head Model

(SVM) and ensemble algorithm, their results have shown that the SVM classifier generally performed better than the others in comparison and also achieved higher accuracy scores using the same datasets. One interesting and realistic contribution, as expressed by Rudraraju et al. [10], was to consider the monitoring of entry and access control as another function to the face mask detection system. This was also one of the few works surveyed that also performed extensive research on the topic of correctness in mask wearing through the architecture of their application.

However, given how the wearing of different masks types as well as the validity of mask-wearing itself in crowded areas has not been broadly reviewed in the literature, we will attempt to fill in that gap with the proposal of a two-stage face mask detection model in this research paper. Our contributions can be summarized as follows:

- We will propose a face mask classification model that employs transfer learning by combining our head model with a backbone model.
- In order to opt for the most prominent backbone, from the architectures of MobileNetV1 [11], MobileNetV2 [12], ResNet50V1 [13] and ResNet50V2 [14], we respectively evaluate their versions of classification models on a custom dataset that we created using the two sources Kaggle-12K [15] and the MaskedFace-Net [16].
- Following the integration of our classification model and RetinaFace, the proposed model's capabilities will then be assessed against the AIZOO Face Mask Detector [17] and Neuralet Face Mask Detector [18] on the two datasets Face Mask Detection [19] and the MAsked FAce [20].

Following this introduction, Section II will briefly describe our methodology; then, the specifications of our setups behind the experiences will be detailed in Section III, whereas Section IV will elaborate further on how each steps in those experiments are conducted with their corresponding results to justify our proposed model abilities; and ultimately, our research paper will be concluded alongside a brief view at possible future improvements at the end of Section V.

II. Methodology

The face mask detector that we are proposing will be composed of two stages: the first stage is the identification of facial regions from an image or a frame and the second one is the classification of detected faces into pre-defined subcategories. The stages are demonstrated following the orders in Figure 1. The design of such sequential model brings 2 benefits to our detector.

Firstly, the state-of-the-art RetinaFace will be applied to identify the Region of Interest (RoI) or faces available in one image. Being a single-stage dense face localization model, it can achieve an average precision score of 91.286% [21] on the hard subset of the WIDER FACE dataset. In other words, the mentioned face detection model was trained given a dataset of crowded, diverse types of faces, including normal, masked and make-up faces. Such process is believed to be beneficial to our model in terms of increasing detection accuracy.

Secondly, during the second stage of classifying all the detected faces, transfer learning is applied using the four backbone models: MobileNetV1, MobileNetV2, ResNet50V1, ResNet50V2. The experiments to opt for the backbone models are described in Section IV. Our head model is simple and light-weighted, including a convolution layer, a max-pooling layer, and a series of fully connected layers and drop-out. Details about the head of the classification model can be found in Fig. 2. The probabilistic output is (p_1, p_2, p_3), where p_i is simply the corresponding probability that the classified image belongs to one of the three classes discussed later in Section IV-A. Such architecture allows us to freely opt for the most prominent backbone model without having to modify our entire model.

III. Implementation

A. Training details

The model is trained using the binary loss function and Adam optimizer with the initial learning rate at 10^{-4}, dropout rate at 0.5 and batch size of 32 on NVIDIA Tesla V100-SXM2 $(16GB)$ using Google Colab environment. Therefore, we will also upload all the datasets to accessible Google Drive folders on the same account for ease of execution and reusability. The training process terminates after 50 $epochs$.

B. Validating and testing details

1) Validating details: There are three purposes to the validating process. Evidently, the first is to evaluate the

TABLE I: Testing Dataset Information

Dataset	No. images	No. faces	No. class 1	No. class 2	No. class 3
Kaggle	853	4072	3232	717	123
MAFA	4935	10033	6354	996	-

where *No. images, No. faces, No. class* 1*, No. class* 2*, No. class* 3 are defined as the number of images, of faces in total and of faces belonging to each of the correctly-worn, no-mask-worn and incorrectly-worn class, respectively.

performance of our classification model. Additional to that, such process will discover how different base models, or the so-called backbones, might affect the efficiency of the classification model once applied. Last but not least, due to the various sources of our training and validation datasets, their heterogeneity, and imbalance should be carefully revised to see whether they are strongly involved in the creation of our model. The metrics mentioned in this subsection are determined following said reasons.

We first define the accuracy of the classification model on the validation set after 50 epochs as follows:

$$\psi_{classify}(y, \hat{y}) = \frac{1}{m} \sum_{i=0}^{m-1} 1(\hat{y}_i = y_i), \tag{1}$$

where m is the size of the validation set and y, $\hat{y}$ is the set of classified and ground-truth label, accordingly.

To illustrate the model stability, the variance σ^2 of the classification model for both training and validation sets is defined in (2).

$$\sigma^2 = \frac{1}{n} \sum_{i=1}^{n} (\psi_i - \mu)^2, \tag{2}$$

where ψ is the accuracy obtained after each epoch i and n is the total number of epochs.

Precision will be used to estimate the classification abilities of our model to not falsely label negative samples. The calculation of recall will also be adopted to estimate the sensitivity in finding all positive samples. The general formula of precision $P(y, \hat{y})$ and recall $R(y, \hat{y})$ are given in (3) and (4).

$$P(y, \hat{y}) = \frac{TP}{TP + FP} \tag{3}$$

$$R(y, \hat{y}) = \frac{TP}{TP + FN}, \tag{4}$$

where TP, TN, FP, FN are respectively true positives, true negative, false positive, false negative number of classified cases.

With the imbalance of the dataset under consideration, a computation is made for the two types of precision and recall: macro and weighted. While macro metrics ignore the imbalance of the dataset, weighted metrics take into account how it can alter the final results. The formula for the macro and weighted precision and recall are given in (5), (6), (7) and (8) respectively.

Let L be the set of available classes, y and $\hat{y}$ are defined as in (1), we have:

$$P_{macro}(y, \hat{y}) = \frac{1}{|L|} \sum_{l \in L} P(y_l, \hat{y}_l) \tag{5}$$

$$P_{weighted}(y, \hat{y}) = \frac{1}{\sum_{l \in L} |\hat{y}_l|} \sum_{l \in L} |\hat{y}_l| \, P(y_l, \hat{y}_l) \tag{6}$$

$$R_{macro}(y, \hat{y}) = \frac{1}{|L|} \sum_{l \in L} R(y_l, \hat{y}_l) \tag{7}$$

$$R_{weighted}(y, \hat{y}) = \frac{1}{\sum_{l \in L} |\hat{y}_l|} \sum_{l \in L} |\hat{y}_l| \, R(y_l, \hat{y}_l) \tag{8}$$

2) Testing details: In order to reasonably evaluate the model with the test set, three metrics, namely average confidence, accuracy of face detection model, and ordinary accuracy are defined in (9), (11), (10).

Let D be the set of all faces detected by our model, then the average confidence β, known as its ability to correctly classify a given face, can be described as:

$$\beta = \overline{1 - \alpha} = \frac{1}{|D|} \sum_{d \in D} \psi_d, \tag{9}$$

where α is the significance level and ψ is the accuracy or the confidence of each detected face.

Assume that T represents the set of all ground-truth faces, we define Ψ_{RoI} as the accuracy of face detection model computed by the formula:

$$\Psi_{RoI} = \frac{|D|}{|T|} \tag{10}$$

The last metric to be used in the testing process is the final accuracy, representing the ability of the model to correctly classify a given face.

Let $A \subset D$, where A is the set of correctly localized and classified faces, the final accuracy Ψ_{final} is defined as

$$\Psi_{final} = \frac{|A|}{|D|} \tag{11}$$

The evaluation method is given in Algorithm 1, in which the threshold is set at $\theta_{lower} = 0.5$. This parameter is used in Line 7 under the form of pseudocode.

Algorithm 1 Evaluation using IoU

```
1: for image = 1, 2 ..., N do
2:     detectedFaces ← Detect faces in that image and classify them
3:     totalDetectedFaces+ = len(detectedFaces)
4:     totalConfidence+ = ∑ detectedFacesConfidence
5:     for realFace = 1, 2, ... do
6:         for detectedFace = 1, 2, ... do
7:             if θ_lower < IoU(realFaceBoundingBox,
               detectedFaceBoundingBox) then
8:                 if realFaceClass == detectedFaceClass then
9:                     trueVal+ = 1
10:                    Break the outer for loop since the correct detected
                       face has been found
11:                end if
12:            end if
13:        end for
14:    end for
15: end for
16: Calculate β, Ψ_RoI, Ψ_final in (9), (10), (11) respectively.
```

IV. Experiment and Result

Given the 4 datasets used in our research, namely Kaggle-12K [15], MaskedFace-Net [16], Kaggle Face Mask [19] and MAsked FAces (MAFA) [20], we have decided to select them accordingly for the tasks and to attain even distribution among the 3 classes of mask-wearing.

A. Dataset

1) Training and validation dataset: The training dataset for the classification stage is a human face dataset composed of all the images collected from two sources, namely, Kaggle-12K [15] and MaskedFace-Net [16]. These images are then classified into 3 classes, as seen in Table I. Class 1 represents the correctly-worn-mask faces, Class 2 being the incorrectly-worn-mask faces, and Class 3 for the no-mask-worn faces. The masks used in this dataset are not necessarily medical masks. The class correctly-worn-mask faces is designated to faces that are fully covered with fabric or medical masks. On the contrary, faces are classified into incorrectly-worn-mask when the people in question are wearing their masks in such ways that vital parts like noses or mouths are left uncovered. There are 13338 images in total, in which 3594 of them are in the incorrectly-worn-mask class, 4816 are for the correctly-worn class, and 4928 for the no-mask faces. The distribution of images for each class are shown in Fig. 3. To appropriately train and validate the classification models, we have divided this dataset into 2 subsets, in which 90% of the images are used as training set and the other 10% are used as validation set. All images contain only 1 face.

2) Testing dataset: Our model will be tested and evaluated on the two datasets Kaggle Face Mask [19] and MAsked FAces (MAFA) [20] with their details listed in Table I. Kindly note that each image in these sets will contain multiple faces for the sake of testing the entire model, and that the annotation file in MAFA has declared the "invalid" class to hold various cases that do not follow our definition of incorrectly-worn-mask class (for example, the faces in question were occluded, blurred, or partially shown in their corresponding frames, etc.); ergo, they have been excluded from our consideration and left untouched.

B. Comparison of our classification model backbone

In this subsection, we will validate the four classification backbone models, namely MobileNetV1, MobileNetV2, ResNet50V1 and ResNet50V2, and filter out the model with the best performance possible.

1) MobileNets: For practical purposes, it is ideal that our face mask detection model can be embedded in cameras or smaller digital devices. Therefore, small yet powerful architectures as MobileNets are considered here in our works. MobileNetV1, proposed by Google in 2017, is a rare model which combines depth wise separable convolutions and 11

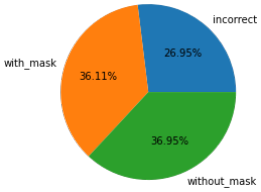

Fig. 3: Training dataset distribution

standard convolutions for its input channels filtering and linear computation. MobileNetV1 only has 88 layers in total and the size of $16MB$ [11]. MobileNetV2, proposed by Google a year after the introduction of MobileNetV1, is an updated version of its first version with highlights in the two new features: linear bottlenecks between the layers, and shortcut connections between the bottlenecks [12]. It was reported to be lighter and $30-40\%$ faster on a Google Pixel phone than MobileNetV1 while having an increase in ImageNet Top 1 accuracy [12, 22].

With MobileNets as our backbones to support the head model described earlier, we are able to provide the information about the accuracy and loss of the model during 50 epochs in Table II. In general, the implementation of both MobileNets as backbones is synonymous with the increase in accuracy and decrease in loss. We can acknowledge that the validation accuracy of the model using MobileNetV2 as the backbone is less stable than the one with MobileNetV1. Withstanding some minor fluctuations, the loss of MobileNetV2-supported model on the validation set tends to increase in the later epochs, whereas its training loss remains the same. For further evaluation of our usage of MobileNets as backbone, the mentioned metrics in Section III are calculated in Table II.

Despite the better performance of MobileNetV2 compared to its ancestor in ImageNet [12, 22], on our dataset, particularly the validation set, MobileNetV1-supported model results in a higher accuracy of 0.9981 after 50 epochs with the minimum accuracy being 0.9940 and the maximum 1.0. In contrast, the accuracy of MobileNetV2-based model ranges from 0.9910 to 0.9985 with the final accuracy at 0.9970. The classification model based on MobileNetV1 is also more stable than the one on MobileNetV2. It should be noted that, in spite of being 1.4 times less in training accuracy variance, the MobileNetV2-supported model scores more in validation accuracy than the MobileNetV1-based. All the precision and recall metrics of the classification model with MobileNetV1 are slightly higher than those of that using MobileNetV2. Additionally, within the margins of our experiments, the MobileNetV1-supported model tends to classify in a shorter duration of 1.139 seconds in comparison with 1.445 seconds of MobileNetV2. Lastly, it is observed from the weighted and macro precision and recall metrics that the imbalance of our dataset is trivial, and thus the validation process strictly follows the performance of the models. Conclusively, MobileNetV1-supported classification model is able to achieve higher results on our dataset and within a shorter period of run time as opposed to the one employing MobileNetV2 as its backbone.

2) ResNets: Being one of the most groundbreaking architecture developed in 2015 by Microsoft and won 1st place in the ILSVRC classification competition with top-5 error rate of 3.57%, ResNet uses residual blocks, which applied the idea of skip connections, to overcome the problem of vanishing gradients while the depth of the convolution network increases [13]. A few months after the birth of ResNet, the second version of it was proposed also by Microsoft. ResNetV2 improved the residual unit, which facilitates the training process and improves generalization [14].

TABLE II: Evaluation on Our Classification Model Using MobileNets

Backbone	σ^2_{train}	σ^2_{val}	$\psi_{classify}$	P_{macro}	$P_{weighted}$	R_{macro}	$R_{weighted}$	Run Time(s)
MobileNetV1	$1.507*10^{-4}$	$1.759*10^{-6}$	0.9981	0.9982	0.9981	0.9980	0.9981	1.139
MobileNetV2	$1.053*10^{-4}$	$1.896*10^{-6}$	0.9970	0.9972	0.9970	0.9967	0.9970	1.445

TABLE III: Evaluation on Our Classification Model Using ResNets

Backbone	σ^2_{train}	σ^2_{val}	$\psi_{classify}$	P_{macro}	$P_{weighted}$	R_{macro}	$R_{weighted}$	Run Time (s)
ResNet50V1	$4.895 * 10^{-5}$	$4.183 * 10^{-7}$	0.9982	0.9984	0.9983	0.9981	0.9982	2.467
ResNet50V2	$2.037 * 10^{-4}$	$4.003 * 10^{-6}$	0.9985	0.9986	0.9985	0.9981	0.9985	2.274

Information concerning the loss and accuracy of ResNet50V1-based and ResNet50V2-based models within 50 epochs is shown in Table. III. It is visible that the accuracy of the training process for these two models rises sharply and then stabilizes. Inversely, on the validation set, both models suffer from minor fluctuations. There are also similarities in their loss functions as the training loss values of both swiftly decrease in the beginning then remain constant until some later epochs where they are to rise one more time. Their evaluation metrics can be found in Table III. By comparison, the scoring results obtained from ResNet50V2-supported model are 0.0002 unit higher than those from other model roughly. While the run time of the model based on ResNet50V1 is longer than that on ResNet50V2, the variance in both train and validation sets' accuracy scores of ResNet50V1 based are approximately 4 to 10 times smaller than of ResNet50V2.

Considering all results, the model based on ResNet50V2 backbone scores the highest in accuracy, precision and recall (both weighted and unweighted). Meanwhile, the ResNet50V1-based model emerged with the lowest variance for both the train and validation sets but at the same time the slowest classifier. Nevertheless, the duration of run time of ResNets are about 2 times longer than that of MobileNets. For the last assessment, the final model, including both localization and classification stages, is tested on the Kaggle dataset. There, we were able to localize 3399 faces over 4072 faces, 3215 faces of which are accurately classified by the model using ResNet50V1. The other three, namely ResNet50V2, MobileNetV1 and MobileNetV2, respectively made 2992, 2992, 2991 classifications. Based on the small variance on the previous dataset and the high accuracy on this dataset, the ResNet50V1-supported model was the reasonably ideal backbone for the classification model that will be used in the following subsection.

C. Result

We now compare our proposed model with another two models, specifically, AIZOO and Neuralet face mask detectors (AIZOO FMD and Neuralet FMD for short, respectively). AIZOO FMD developed by AIZOOTech, is a light-weighted, single-stage detector with only 1.01 million parameters and 24 layers for location and classification. AIZOO supports all popular deep learning frameworks model and inference code. In this comparison, we use only the Tensorflow version [17]. Neuralet FMD is supported by Neuralet company and is sponsored by Lanthorn Solutions. It is a two-stage detector

TABLE IV: Summary of Models

Model	Dataset	β	Ψ_{RoI}	Ψ_{final}
Proposed	Kaggle	0.9960	0.8347	0.9459
	MAFA	0.9950	0.9580	0.6960
AIZOO FMD	Kaggle	0.8671	0.5454	0.8249
	MAFA	0.9325	0.7938	0.6453
Neuralet FMD	Kaggle	0.9723	0.4050	0.3220
	MAFA	0.9383	0.1886	0.0587

that can both propose region of interests and provide the needed classification on those regions. Its first stage, known as the detector stage, uses one of two models for the $\times 86$ configuration, openpifpaf model (model to estimate human pose estimation) and tinyface model. In this subsection, we run the model using openpifpaf for detector and OFM Classifier for the classifier of Neuralet FMD [18].

As shown in Fig. 4 and Fig. 5, the results for accuracy and confidence of our proposed model are higher than those of AIZOO and Neuralet on both datasets. By using RetinaFace for face localization, which was able to detect 900 faces out of reportedly 1151 people, our proposed model managed to recognize more faces per photo and returned higher score of Ψ_{RoI} than the other mentioned models on both test datasets.

With ResNet50V1, the accuracy of our model to appropriately detect the faces across more than 800 images of the Kaggle dataset is at 0.9459 and is relatively higher than the compared models. On the other hand, for the MAFA dataset, perhaps as a consequence of the mentioned omission of incorrectly-worn masks previously mentioned in section III, the accuracy we managed to achieve was an unsatisfactory value of 0.696. Still, this result is the highest one achieved among the three models, despite the fact that ours was trained to label the same faces under the three aforementioned categories, while the conventionally used annotation in the dataset contained only two labels that both AIZOO and Neuralet could follow. The proposed model also gave competent results on the scores, or the so-called confidence, of precisely classifying a given region of interest.

By achieving good results in Ψ_{RoI} and Ψ_{final}, our model has therefore showcased its ability to (1) convincingly detect and classify faces in dense population regions as well as (2) to separate the detected faces into 3 classes, which enabled a more appropriate classification of the incorrectly worn mask faces. Please refer to the resulted illustrations for both cases that were featured on our GitHub repository using this link: https://github.com/barbatoz0220/Densely-populated-FMD.git

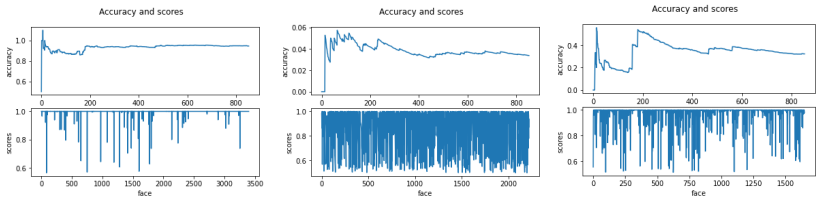

Fig. 4: Comparison of Accuracy and Score (Confidence) Among the Proposed Model - AIZOO - Neuralet on Kaggle Dataset

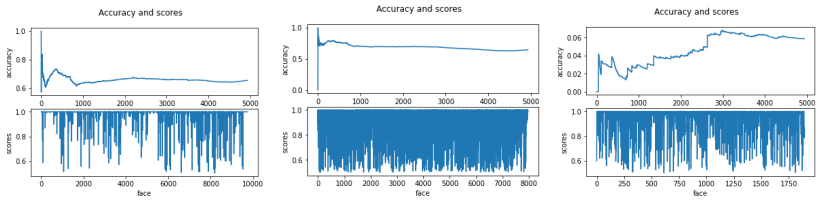

Fig. 5: Comparison of Accuracy and Score (Confidence) Among the Proposed Model - AIZOO - Neuralet on MAFA Dataset

V. Conclusion

Given the lack of research on mask detection for crowded regions, as well as on the classification of face masks with respect to variety and validity, we have proposed a new face mask detector model for detection in densely populated regions and validation of masks wearing following how they are worn: correctly, incorrectly and without. Our proposed model correctly localized 83.47% faces and classified 94.59% of the confined set. While there are still certain limitations to some of the class variance, the performance metrics have justified our effectiveness in the combination of ResNet50V1 and RetinaFace. It is certainly possible for our model to be better optimized and utilized in the foreseeable future. In terms of data, we firmly believe that extensive attempts to improve the imbalance in current sets and, perhaps, to modify our own set from renowned sources will allow us to achieve more optimistic results. Further researches to exploit more capable architectures would definitely be considered and integration with tools, such as OpenCV, will ensure more opportunities for public usage through real-world application.

References

[1] Warwick McKibbin and Roshen Fernando. The economic impact of covid-19. *Economics in the Time of COVID-19*, 45, 2020.

[2] World Health Organization et al. World health organization coronavirus disease 2019 (covid-19) situation report, 2020.

[3] Worldometers. Coronavirus updates (live) - covid-19 coronavirus pandemic. Available at https://www.worldometers.info/coronavirus/.

[4] Steffen E Eikenberry, Marina Mancuso, Enahoro Iboi, Tin Phan, Keenan Eikenberry, Yang Kuang, Eric Kostelich, and Abba B Gumel. To mask or not to mask: Modeling the potential for face mask use by the general public to curtail the covid-19 pandemic. *Infectious Disease Modelling*, 5:293–308, 2020.

[5] Mei Wang and Weihong Deng. Deep face recognition: A survey. *Neurocomputing*, 429:215–244, 2021.

[6] Connor Shorten, Taghi M Khoshgoftaar, and Borko Furht. Deep learning applications for covid-19. *Journal of big Data*, 8(1):1–54, 2021.

[7] Elliot Mbunge, Sakhile Simelane, Stephen G Fashoto, Boluwaji Akinnuwesi, and Andile S Metfula. Application of deep learning and machine learning models to detect covid-19 face masks-a review. *Sustainable Operations and Computers*, 2021.

[8] Mohamed Loey, Gunasekaran Manogaran, Mohamed Hamed N Taha, and Nour Eldeen M Khalifa. Fighting against covid-19: A novel deep learning model based on yolo-v2 with resnet-50 for medical face mask detection. *Sustainable Cities and Society*, page 102600, 2020.

[9] Mohamed Loey, Gunasekaran Manogaran, Mohamed Hamed N Taha, and Nour Eldeen 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*, 167:108288, 2021.

[10] Srinivasa Raju Rudraraju, Nagender Kumar Suryadevara, and Atul Negi. Face mask detection at the fog computing gateway. In *2020 15th Conference on Computer Science and Information Systems (FedCSIS)*, pages 521–524. IEEE, 2020.

[11] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[12] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016.

[15] Ashish Jangra. Face mask 12k images dataset - 12k images divided in training testing and validation directories, May 2020. Available at https://www.kaggle.com/ashishjangra27/face-mask-12k-images-dataset.

[16] Adnane Cabani, Karim Hammoudi, Halim Benhabiles, and Mahmoud Melkemi. Maskedface-net – a dataset of correctly/incorrectly masked face images in the context of covid-19. *Smart Health*, 19:100144, 2021. Available at https://github.com/cabani/MaskedFace-Net.

[17] Detect faces and determine whether people are wearing mask. Available at https://github.com/AIZOOTech/FaceMaskDetection.

[18] Face mask detection at the edge. Available at https://neuralet.com/face-mask-detection-at-the-edge/#showcase-section-4.

[19] Larxel. Face mask detection, May 2020. Available at https://www.kaggle.com/andrewmvd/face-mask-detection.

[20] Rahul. Masked face data, May 2020. Available at https://www.kaggle.com/rahulmangalampalli/mafa-data.

[21] Jiankang Deng, Jia Guo, Evangelos Ververas, Irene Kotsia, and Stefanos Zafeiriou. Retinaface: Single-shot multi-level face localisation in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5203–5212, 2020.

[22] Howard Sandler. Mobilenetv2: The next generation of on-device computer vision networks. Available at https://ai.googleblog.com/2018/04/mobilenetv2-next-generation-of-on.html.