Object Detection Frameworks for Real-Time, Scale-Invariant Face Mask Detection

Louis Philippe B. Facun, Maria Jeseca C. Baculo, Marlon F. Libao, Ceazar M. Eisma, Christian B. Fredeluces, Rigzor A. Garlejo, and Raymart S. Idio

***Abstract*—As the world experiences a pandemic caused by the spread of COVID-19, following health protocols like the proper wearing of face masks is essential to help prevent the spread of the virus as stressed by the World Health Organization (WHO). In the Philippines, manual monitoring of these measures may not be as efficient when handling large crowds. The authors were able to generate a real-time, scale-invariant face mask recognition model that also handles occlusions. It was trained using images from the wild to simulate real-world conditions. Existing state-of-the-art object detection frameworks such as MobileNet, RetinaNet, and YOLOv4 were implemented to detect and localize the proper use of face masks in public. To evaluate the performance of the models, the mAP, inference time, and precision curve were used. The performance of the model has resulted to as high as .94 mAP with an inference time of 73 ms.**

***Index Terms*—Face mask detection, object detection, deep learning, occlusion, scale-invariance.**

I. Introduction

The COVID-19 pandemic had caused a devastating and life-changing global crisis since it negatively affected health, livelihood and food systems, and the world's economy [3]. Computer vision applications are transforming the world through modern technologies to help alleviate the toll of the global spread of this disease. From the health industry to manufacturing, it helps make everyday functions in the "new normal" better and easier.

The weekly epidemiological update reported that there are currently 53.7 million confirmed cases of coronavirus disease or COVID-19 with 1.3 million deaths globally. Additionally, the rate of new cases and deaths continued to rise with almost 4 million new cases and 60,000 new deaths were recorded. Until no vaccine is developed, following health protocols such as properly wearing face masks is essential to help prevent the transmission of the virus. To prevent the spread, wearing a face mask on a sick person was proven effective by blocking the respiratory droplets that carry the virus when that person talks, coughs, or sneezes. A healthy person wearing a face mask protects himself against catching the respiratory droplets of other people.

Government agencies have recommended the mandatory wearing of face masks by all residents in all public places. This preventive measure is being monitored manually by all Operation Officers. However, this approach may not be as efficient because monitoring such may be difficult, especially when handling large crowds.

Given the statistics brought about by the COVID-19 pandemic, various computer vision and deep learning techniques have been implemented to applications such as face mask detection to help facilitate the monitoring of preventive ways to contract the said virus.

The application of this paper will be of great benefit to society, considering that the COVID-19 pandemic is affecting the whole world today. The implementation of this model can make the public spaces safer since people can now strictly follow the face mask-wearing protocols with the help of its strict monitoring. This application can also assist the local government and private agencies and establishments in ensuring efficacy and facilitating the monitoring of protocols.

With these motivations in mind, the authors propose a technological solution that automatically detects the violations in the proper wearing of face masks in crowded public places monitored by surveillance cameras.

The authors gathered the instances of the dataset for training from publicly available online resources. The model aims to detect the face and the presence or absence of a face mask then make a prediction that falls on either of the three classes: Wearing a Mask, Not Wearing a Mask, and Improperly Wearing a Mask. The authors have trained face mask models using some of the state-of-the-art object detection techniques. Finally, the authors used mAP, inference time, and the Precision-Recall curve to evaluate the model's performance.

II. Related Works

The following are the different computer vision and deep learning techniques implemented to detect face masks from images [4], [7] and [10].

Reference [8] trained a face mask detection model using their NVIDIA Transfer Learning Toolkit (TLT) and deployed the model using NVIDIA DeepStream SDK. The dataset consisted of 27,000 images from the FDDB and WiderFace datasets for faces without masks and the MaFA and Kaggle Medical mask datasets for faces with masks, and used cherry-picked images consisting of 4,000 images from the same dataset. The experiments showed that using fewer datasets on images without masks results in higher accuracy. Using the datasets of 27,000 images, they achieved a mean Average Precision (mAP) of 78.98% and 86.12% mAP using the 4,000 images. The model was tested on a camera mounted

Manuscript received August 20, 2021; revised December 23, 2021. This work was supported in part by the Don Mariano Marcos Memorial State University ◦

The authors are with the Don Mariano Marcos Memorial State University, Agoo, \Philippines (e-mail: lpfacun@student.dmmmsu.edu.ph, mjbaculo@dmmmsu.edu.ph, mlibao@student.dmmmsu.edu.ph, ceisma@student.dmmmsu.edu.ph, cfredeluces@student.dmmmsu.edu.ph, rgarlejo@student.dmmmsu.edu.ph, ridio@student.dmmmsu.edu.ph).

doi: 10.18178/ijfcc.2022.11.1.582

5 feet away and not farther, and not tested from an overlooking camera perspective.

Reference [5] created a deep learning-based model and architecture called "Facemasknet." They used a few datasets consisting of 35 images, 15 images with improperly worn masks, 10 images with masks, and 10 images without masks but had three classifications. The dataset is then preprocessed to 227x227x3 pixels. Despite having limited images, the model was tested where it produced a maximum accuracy of 98.7% and a minimum accuracy of 74.97%.

Reference [11] also utilized a deep learning method in their face mask identification model and had three classifications. They conducted image preprocessing of their datasets using image Super-Resolution (SR). The datasets used were: CelebA with 70,534 images and the CASIA Web-Face dataset with 493,750 images. To improve the accuracy, images with a size of no more than 150x150 are enhanced using SR. After preprocessing, the final number of datasets consisted of 3,835 images. They used a super-resolution crack network (SRCNet) to implement the Convolutional Neural Network (CNN) model. The experiment results achieved a 98.7% accuracy.

Despite the valuable performance of the aforementioned models, similar limitations can be observed. First, the datasets used mostly consisted of surgical masks only. These datasets may have an issue with detecting other types of masks since, in some places, cloth masks are more common than surgical masks. To cater to this need, datasets should include different types of masks. Also, the test set of these models only used frontal and close-up of images, instances with occlusions and are farther in the frame may be difficult to detect in this case. Thus, a scale-invariant model that can handle occluded samples may solve this problem.

III. Methodology

Fig. 1 illustrates the procedures and methodologies that were implemented in this study. This includes acquiring and preprocessing the data and the training and evaluation of the model.

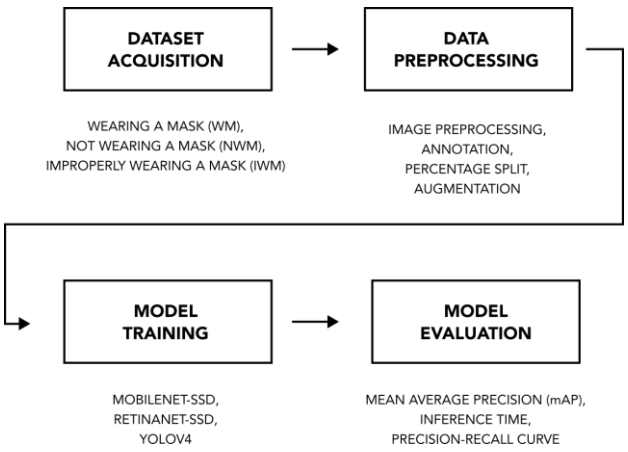

Fig. 1. Block diagram.

The authors used the following software tools to make for dataset preparation, model training, and evaluation: Google Colaboratory as the training and testing environment, Python 3 as the primary programming language; Tensorflow Object Detection API that is built on Tensorflow library, which is a framework that provides tools for quickly creating object detection models; Darknet, an open-source neural network framework, written in C, that is used in YOLOv4 framework, and a required dependency for both Tensorflow and Darknet framework, OpenCV (Open Source Computer Vision) library, an open-source computer vision library that is optimized for real-time computer vision. Other optional dependencies for training and inference include NVIDIA's CUDA and CuDNN for optimized deep learning calculations.

A. Data Gathering

The authors acquired datasets that are publicly available in the wild. The classes represented by the instances belong to the following labels: *Wearing a Mask (WM), Not Wearing a Mask (NWM), and Improperly Wearing a Mask (IWM)*.
To collect the instances for training, the authors manually collected publicly available images from search engines and consented public social media images of people. For the images taken from individuals, a consent form was signed by certain respondents.

The instances in the *WM* class contain people wearing plain surgical or cloth masks. On the other hand, *NWM* class instances contain faces of people that are not wearing a mask. These images consisted of different conditions, including people looking at the camera, group meetings, and pedestrians. For the *IWM* instances, the authors gathered images containing different ways people improperly wear a mask. These include the mouth is covered, but the nose is visible, the mask is hanging on the ear, and the mask is only covering the neck.

In order to meet a minimum number of 1,000 annotated instances per class, the authors gathered a total of 1,365 images from various sources. The gathered images vary in size, image lighting, angles where faces are looking, and image quality to best represent the real-life conditions of images taken on a day-to-day basis. The authors ensured that the collected images are mixed with different scales and how occluded the facial details were.

The final number of instances that are annotated in *WM* class is 1,012 instances. The number of annotated instances for the *NWM* class is equivalent to 1,006 instances. For the *IWM* class, the final number of instances that are annotated is 1,007 instances.

B. Data Preprocessing

The authors applied image preprocessing techniques such as format conversion, metadata removal, rescaling, and cropping. This preprocessing was automatically done using a Python script programmed by the authors. Format conversion was applied to ensure format compatibility among the instances. Moreover, metadata removal was used to reduce the image size and to ensure disinformation. The resizing and cropping of instances were made to ensure the uniformity of the aspect ratio before training.

After collecting unlabeled images, the authors performed annotations by manually drawing bounding boxes on the images using an open-source image annotation tool called LabelImg, where annotations files are exported in YOLO format (.txt) and later converted to PASCAL VOC (.xml) format.

Data augmentation is a technique used to increase the size of the dataset by generating training instances by applying different image processing techniques. It is instrumental in

applying face mask detectors because it can generalize the dataset in various conditions. To perform data augmentation, the authors used the albumentations Python library created by [2]. This library supports augmenting datasets with bounding boxes because it also adjusts the bounding boxes whenever an augmentation technique that changes the position of objects is applied.

After rescaling the images, the authors annotated the images using the LabelImg tool. The images were carefully and consistently annotated so that the models would generalize well. For the face mask detection model, the bounding boxes were precisely drawn in the area of the face, where the face is the object of interest, excluding the hair. All annotation files were saved in YOLO format for YOLOv4 training and later converted to PASCAL VOC format for RetinaNet and MobileNet-SSD frameworks.

The authors split the dataset for training and testing. The dataset was split correctly into 80% dataset distribution for the training set and 20% instances for the test set. The splitting is based on the number of instances and not on the number of images. In addition, the splitting is balanced into an approximately 8:2 ratio per class. For example, in *NWM* class, images that contain 80% of instances were set as training data, and the remaining 20% were included for testing data. The authors chose to stick to the nearest possible value of the calculated values, which equates to 667 instances used in 300 images for the test set since there is no definite number of instances contained in the images used and the images used in the test set were randomly chosen.

Data augmentation was applied to the training data, in exclusion of the test set. The transformations done to the images were: horizontal flip, random rotation, blur, and gaussian noise. First, the horizontal flip is applied to the original image, and both original and flipped images are augmented with combinations of random rotation, random blur, and random addition of noise.

After augmentation, the number of images increased by 4-folds using the authors' techniques plus the factors added by the predefined augmentation techniques given in the frameworks.

C. Face Mask Detection Model Training

Object detection in computer vision locates instances of objects in an image or a video. Newer object detection frameworks utilize machine learning and deep learning networks to create more accurate predictions than before. Object detection works using two tasks: localization and classification, where localization tries to predict the object's location by segmenting the image or video, while classification identifies the object's correct label.

For the model training, the authors used three different object detection frameworks. All three chosen frameworks utilize a one-stage technique that eliminates the process of region proposal, making the object detection results in faster predictions. MobileNet-SSD, RetinaNet, and YOLOv4 are state-of-the-art one-stage object detection frameworks that are all open-source frameworks.

The SSD MobileNet V1 [6] implementation in Python was used to train the MobileNet model. The size of the training set used for this model was approximately 14,484 - 9,675 of which was the result of the authors' additional augmentation, and the remaining was from the random crop it does during training. The base learner, MobileNet V1, is a 28-layered deep learning architecture designed for real-time detections. For this architecture, the input size was set to 640x640. Fig. 2 illustrates the structure of the MobileNet framework used in this study.

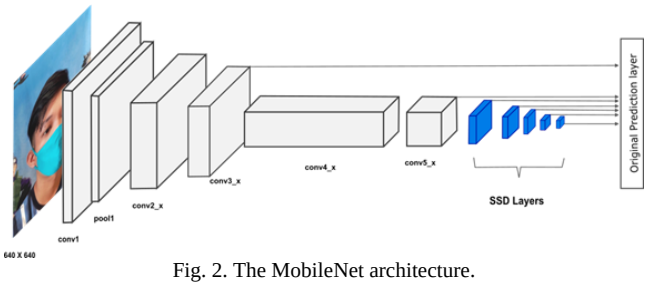

Fig. 2. The MobileNet architecture.

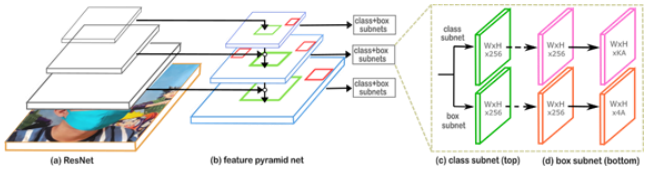

Fig. 3. The RetinaNet architecture.

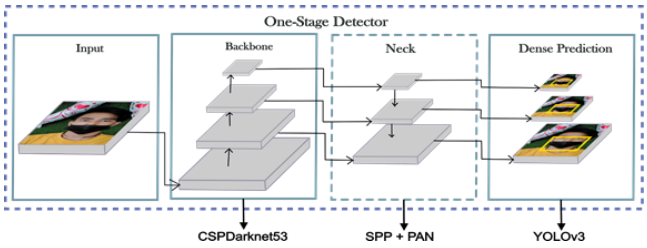

Fig. 4. The YOLOv4 architecture.

RetinaNet [9] also employed similar parameters and augmentations done in the MobileNet model. The training set used for this model was also approximately 14,484 - 9,675, which resulted from the authors' additional augmentation. The remainder was from the random crop it performs during training. ResNet-50, the base learner for RetinaNet, is a 50-layer deep convolutional neural network. ResNet's fundamental breakthrough was that it enabled authors to train extremely deep neural networks. The input size for this architecture was also set to 640x640. The RetinaNet architecture is presented in Figure 3.

YOLOv4 [1] is also a one-stage detector and uses CSPDarknet53 as its base learner. The four additional augmentation techniques were also applied to the instances along with the resulting images generated by the CutMix and Mosaic algorithm integrated with the Bag-of-Freebies in YOLOv4. The resulting training set size was 144,000 images. The framework used in this study is a 161-layer deep convolutional neural network, and the input size for this architecture was also set to 640x640. In YOLOv4 Darknet, training is done in max batches rather than epochs. The documentation recommends a value of 6,000 based on the number of classes, which is 3, and multiplied by 2,000, which equals 6,000. Fig. 4 shows the YOLOv4 architecture.

D. Evaluation

To evaluate the performance of face mask detection models using the MobileNet-SSD, RetinaNet-SSD, and YOLOv4 frameworks, the authors used the PASCAL VOC 2021 metric. This metric evaluates the model's performance using the Mean Average Precision (mAP) in the IoU threshold of 0.5. The mAP is the standard basis for measuring the models' prediction performance and the inference time for measuring

the model's speed. By measuring the mAP, the authors have the best view of seeing how well the model performed. Other evaluation metrics such as Intersection over Union (IoU), Precision, and Recall should be first calculated to get the Average Precision (AP) per class. In this study, if the measured mAP is greater than or equal to 0.80, the models will be considered acceptable. This value served as the threshold that the models must achieve. Otherwise, optimization is required.

The authors used the confusion matrix to evaluate the performance of traditional classification models. This matrix is where the accuracy of correctly classified instances predicted by the model was derived. For object detection, correctly classified instances must pass the acceptable IoU threshold of 0.50.

The precision metrics are defined as the number of correct predictions in all predictions made, while the recall metric is defined as how well the predictions correctly predicted all the true positives.

The PR curve evaluates the overlap between the classes being predicted. A perfect test, in technical terms, is characterized by 100% precision and 100% recall. Below is an example of a perfect test. The upper-right corner of the graph indicates the 100% value of the PR, meaning the closer the lines are to this corner, the better the test is.

IV. Experimental Results

This section presents the experiments' results to train face mask detection models, which can generalize the test data well. First, an inference was executed for all images in the testing set using their libraries for running predictions based on the saved weights of the trained models. Afterwards, the predictions made by each model were extracted in the form of text files and used for evaluating the performance of the models. The prediction data generated by the models consist of the confidence score, the label, and the bounding box coordinates. Using the values of the ground truth and the prediction, the mAP was computed. In this study, the metric followed was the metric used in the PASCAL VOC 2012 challenge, where the default minimum IoU threshold set for evaluation was 0.5.

The following confusion matrices illustrate the performance of the models based on the predictions made in comparison with the ground truth. Confusion matrices show the correctly classified instances for each of the classes. There were also occurrences when the models were not able to detect instances that were labeled as ground truth. To handle this, the authors added another row signified by the *none* class.

According to the confusion matrix in Fig. 5, the models were able to surpass the 0.80 mAP threshold set to indicate model acceptability. This threshold implies that all the models were able to generalize well in the three classes. YOLOv4 had the most correctly classified instances of the three models. The model was able to perform best in recognizing *NWM* class. This is intuitive because this is the easiest class that can be predicted due to the lack of a mask on the facial area.

On the other hand, very minimal incorrect predictions were made in the different classes given. However, the majority of the false positives belong to the *none* class. These false positives happen when the model generates no bounding boxes or predictions in correspondence to the ground truth. Upon further inspection, these images are characterized by objects in more complex angles, such as a side view of a face and a face looking down, where training data only had a few samples of these angles. In addition, small objects made it more difficult for the model to extract features from these images. As reflected in the figure, YOLOv4 achieved the highest number of correctly classified instances among these models.

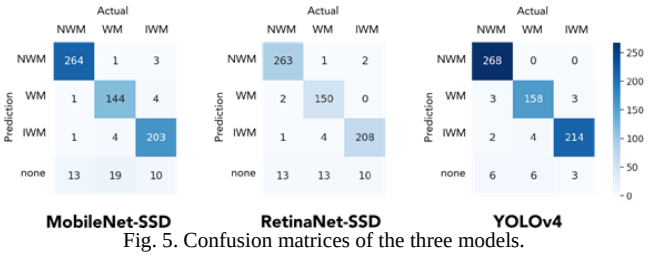

Fig. 5. Confusion matrices of the three models.

Based on the values in the confusion matrix, the authors were able to compute the precision of the models for each of the classes in the dataset. The performance is shown in Table 1. The results manifest that YOLOv4 surpassed the average precision of the classes in both MobileNet and RetinaNet. It can also be observed that the YOLOv4 model was able to predict better the *NWM* and *IWM* classes having 0.95 AP for both, while predictions made for *WM* class come in a close second with 0.92 AP.

TABLE I: Comparison of Average Precision of the Models per Class

Architecture	Average Precision		
	NWM	WM	IWM
SSD MobileNet V1	0.91	0.77	0.85
SSD ResNet 50 V1 (RetinaNet)	0.89	0.76	0.84
CSPDarknet53 (YOLOv4)	0.95	0.92	0.95

Table II shows the summary of the performance of the models in terms of the inference time and mAP. The performance of the models was also quantified using the mAP with the IoU threshold set to 0.50. The results show that the mAP for YOLOv4 (0.94) with base learner CSPDarknet53 significantly outperformed the MobileNet and RetinaNet models with the mAP of 0.84 and 0.83 respectively. The predefined augmentation techniques included in the Bag-of-Freebies in YOLOv4 and the deeper base learner may have contributed to the said model's better performance than the other two.

TABLE II: Comparison of Performance of the Models

Architecture	Inference Time	mAP
SSD MobileNet V1	442 ms	0.77
SSD ResNet 50 V1 (RetinaNet)	702 ms	0.76
CSPDarknet53 (YOLOv4)	73 ms	0.92

In terms of speed, the models were evaluated for how long the predictions were made based on the images in the test set. The average inference time was computed for all the test images. Results show that YOLOv4 was 6 times faster (73 ms) than MobileNet and almost 10 times as fast as the model trained using RetinaNet.

The PR curve was used to evaluate the performance of the model across a variety of thresholds. In this study, the confidence score generated by the classifier along with the values of precision and recall were used. This metric aims to reach the highest point of the upper right corner of the graph,

which would signify how well a classifier was trained based on the precision and recall values.

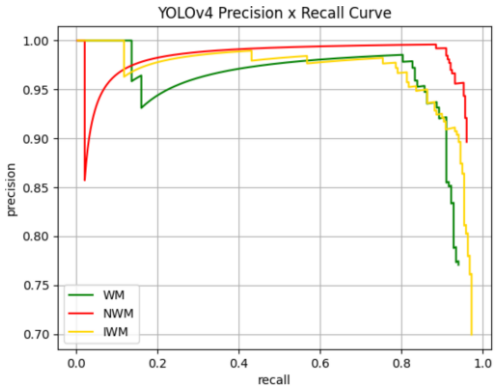

Fig. 6. Precision-recall curve for the YOLOv4 classifier.

The PR curve shown in Fig. 6 is consistent with the trends associated with the confusion matrices and other performance indicators. It can be noticed that the *NWM* class (red line) of the YOLOv4 classifier reached the highest precision values among the classes. This signifies the integrity of the experimental results conducted in this study that all models have consistent generalizations based on multiple different performance metrics.

Fig. 7. Sample predictions by the YOLOv4 model.

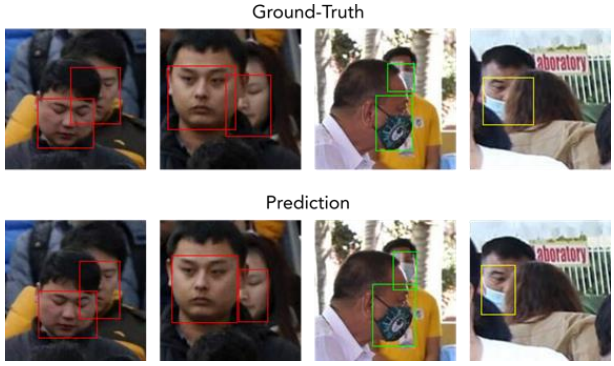

Fig. 8. Sample predictions by the YOLOv4 model on occluded instances.

Fig. 7 shows the ground truth along with the predicted bounding boxes and labels of some sample images from the test set. For this illustration, the detections of YOLOv4 are presented since this model achieved the highest performance among the three models. It can be observed that YOLOv4 was able to predict the different classes accurately and the bounding boxes were able to overlap most of the ground truth boxes. It was also able to detect the classes at different scales and angles. This implies that the model trained using YOLOv4 was scale-invariant and was able to cater to multiple occluded instances.

Fig. 8 shows the correct predictions in occluded faces. The results manifest that the model was able to locate and classify faces with missing details accurately. As seen in the images, the bounding boxes differ in size from the bounding boxes of the ground truth. Despite labelling the images with a larger box area, the model was able to limit the region to the most useful pixels in the image.

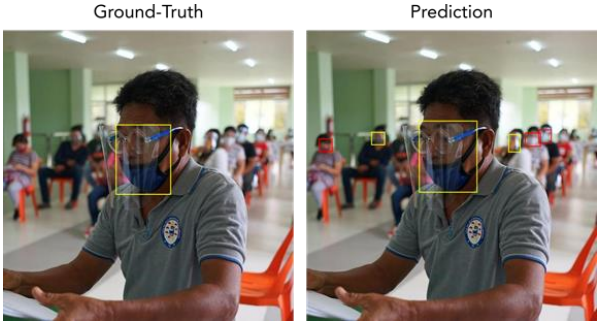

Fig. 9. Sample prediction by the YOLOv4 model on unlabeled data.

Another observation can be made using Figure 9. In this image, the faces/heads in the background were detected. However, these objects were not annotated as ground truth since the human recognition aspect of the authors could not classify on what class the instances belong because the images were blurred. Despite this, YOLOv4 managed to localize the faces/heads, but predictions achieved low confidence scores of 0.28, 0.15, 0.13, 0.56, and 0.12.

V. Conclusion and Future Works

In this study, the authors trained a real-time face mask detection model using object detection frameworks to handle scale invariance and occlusions. They acquired dataset and preprocessed it, trained a face mask detection model, and evaluated its performance. It was found that the YOLOv4 model was able to easily pass the 0.80 mAP threshold for model acceptability. The computed average inference time also supports that the model can be used in real-time. Moreover, based on the PR curve, YOLOv4 was able to outperform the other two models and generalize in detecting the three classes. Thus, the model was proven to be scale-invariant and handles occlusions well. These results imply that the model may be used in public places using surveillance cameras to monitor the wearing of face masks in public.

For future studies, it is recommended to gather more images such as face masks hanging on the ear, masks on the neck, and people with moustaches. This may improve the prediction from the models' problem confusing *NWM* class to *IWM* class. Further experiments on data preprocessing techniques should be tested. Bounding boxes in annotation could be drawn including the whole head. This may improve the localization of small objects as the current annotation only considers the face and the mask. In addition, more augmentations could be applied to the dataset as YOLOv4 proved that additional train-time augmentations contributed to higher mAP, and higher values on augmentations such as

blur and noise should be tried.

Finally, for future system developments, face shield recognition, social distancing violation detection, and image-based sensors to detect coughing or sneezing may be integrated with this model to develop a complete new-normal surveillance system to enforce social responsibility and public safety in common places.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

LP conducted the experiments, MB served as technical consultant, ML, CE, CF, RG and RI collected the dataset and served as documentation committee.

Acknowledgment

We thank our Heavenly Father providing us strength to finish this worthwhile undertaking.

References

[1] A. Bochkovskiy, C. Y. Wang, and H. Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," 2020.
[2] A. Buslaev, V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, and A. A. Kalinin, "Albumentations: Fast and flexible image augmentations," *Information*, vol. 11, no. 2, 2020.
[3] K. Chriscaden. (2020). Impact of COVID-19 on people's livelihoods, their health and our food systems. [Online]. Available: https://www.who.int/news/item/13-10-2020-impact-of-covid-19-on-people's-livelihoods-their -health-and-our-food-systems
[4] S. Degadwala, D. Vyas, U. Chakraborty, A. R. Dider, and H. Biswas, "Yolo-v4 deep learning model for medical face mask detection," in *Proc. 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS)*, pp. 209-213.
[5] M. Inamdar and N. Mehendale, "Real-time face mask identification using facemasknet deep learning network," 2020.
[6] J. Jiang, H. Xu, S. Zhang, and Y. Fang, "Object detection algorithm based on multiheaded attention," *Applied Sciences*, vol. 9, no. 9, 2019.
[7] M. Jiang, X. Fan, and H. Yan, "Retinamask: A face mask detector," 2020.
[8] A. Kularni, A. Vishwanath, and C. Shah, "Implementing a real-time, AI-based, face mask detector application for COVID-19," 2021.
[9] T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE International Conference on Computer Vision*, pp. 2980-2988, 2017.
[10] P. Nagrath, R. Jain, A. Madan, R. Arora, P. Kataria, and J. Hemanth, "SSDMNV2: A real time DNN-based face mask detection system using single shot multibox detector and MobileNetV2," 2021.
[11] B. Qin and D. Li, "Identifying facemask-wearing condition using image super-resolution with classification network to prevent COVID-19," *Sensors*, vol. 20, no. 18, 2020.

Louis Philippe B. Facun was born at Sison, Pangasinan on September 22, 1996. He received his bachelor's degree of computer science from Don Mariano Marcos Memorial State University in Agoo, La Union, Philippines. He is currently taking master's degree of computer science from the same university.

Maria Jeseca C. Baculo was born at Baguio City, Philippines on December 26, 1987. She is currently a PhDCS candidate at the De La Salle University – Manila and specializes in image processing and computer vision.

She is a faculty member of the Don Mariano Marcos Memorial State University and her research interests include machine learning, computer vision and deep learning.

Marlon F. Libao was born at San Fabian, Pangasinan on September 17, 1996. He received his bachelor's degree of computer science from Don Mariano Marcos Memorial State University in Agoo, La Union, Philippines.

Ceazar M. Eisma was born at Agoo, La Union on September 2, 1996. He received his bachelor's degree of computer science from Don Mariano Marcos Memorial State University in Agoo, La Union, Philippines.

Christian B. Fredeluces was born at Santo Tomas, La Union on August 21, 1997. He received his bachelor's degree of computer science from Don Mariano Marcos Memorial State University in Agoo, La Union, Philippines.

Rigzor A. Garlejo was born at Aringay, La Union on February 24, 1995. He received his bachelor's degree of computer science from Don Mariano Marcos Memorial State University in Agoo, La Union, Philippines.

Raymart S. Idio was born at Mangaldan, Pangasinan on August 8, 1996. He received his bachelor's degree of computer science from Don Mariano Marcos Memorial State University in Agoo, La Union, Philippines.