Performance Evaluation of Face Mask Detection for Real-Time Implementation on an RPi

Ivan George L. Tarun[1], Vidal Wyatt M. Lopez[2], Pamela Anne C. Serrano[3], Patricia Angela R. Abu[4], Rosula S.J. Reyes[5], Ma. Regina Justina E. Estuar[6]
Ateneo Laboratory for Intelligent Visual Environments-Dept. of Information Systems and Computer Science
Ateneo de Manila University, Quezon City, Philippines[1,2,3,4]
Dept. of Electronics-Computer and Communications Engineering, Ateneo de Manila University, Quezon City, Philippines[5]
Ateneo Center for Computing Competency and Research-Dept. of Information Systems and Computer Science
Ateneo de Manila University, Quezon City, Philippines[6]

***Abstract*—Mask-wearing remains to be one of the primary protective measures against COVID-19. To address the difficulty of manual compliance monitoring, face mask detection models considerate of both frontal and angled faces were developed. This study aimed to test the performance of the said models in classifying multi-face images and upon running on a Raspberry Pi device. The accuracies and inference speeds were measured and compared when inferencing images with one, two, and three faces and on the desktop and the Raspberry Pi. With an increasing number of faces in an image, the models' accuracies were observed to decline, while their speeds were not significantly affected. Moreover, the YOLOv5 Small model was regarded to be potentially the best model for use on lower resource platforms, as it experienced a 3.33% increase in accuracy and recorded the least inference time of two seconds per image among the models.**

***Keywords*—Face mask detection; multi-face detection; Raspberry Pi; embedded platform**

I. Introduction

Throughout history, infectious diseases have continuously emerged and evolved due to natural causes and human activities [1][2]. These diseases can affect a significant number of people and even become global health issues. A timely example would be the Coronavirus disease or COVID-19, a highly contagious disease caused by the Severe Acute Respiratory Syndrome Coronavirus 2 or SARS-CoV-2 [3]. It originated in Wuhan, China in December 2019 and was eventually declared by the World Health Organization (WHO) a pandemic from March 2020 to May 2023 [4][5]. In the first 50 days of its onset in China, there were more than 70,000 infected individuals and 1800 deaths recorded [6]. In the current Philippine context, the Department of Health (DOH) has reported a total of about 4.16 million cases of infection, 66,000 deaths, and 4.09 million recoveries in the Philippines as of June 2023 [7].

To combat the spread of COVID-19, several health protocols have been issued by authorities and institutions. Among the common ones are physical distancing, hand hygiene, surface disinfection, proper ventilation, and, especially, mask-wearing [8]. Regarding mask use, the WHO promotes the wearing of non-medical or medical masks in poorly ventilated or crowded indoor settings and public areas with insufficient physical distancing. Moreover, wearing strictly medical masks is recommended for vulnerable populations, potential or confirmed COVID-19 patients, and caretakers of COVID-19 patients [9]. The Centers for Disease Control and Prevention (CDC) recommends mask-wearing in public transportation vehicles and hubs. For people situated in COVID-19 hotspots, the CDC requires the use of face masks, especially for those who are highly susceptible to severe infection [10].

For guidelines against COVID-19 to take full effect, public compliance is essential. However, the extended pandemic duration has resulted in growing complacency and, consequently, disobedience to health protocols [11]. While stricter monitoring of compliance can be helpful, the need for physical distancing and a reduced workforce due to the pandemic makes it challenging. To address this issue, the Ateneo Laboratory for Intelligent Visual Environment (ALIVE) has developed face mask detection models that can classify medically-approved masks, non-medically-approved masks, and unmasked faces in consideration of both forward-looking and angled or side-view face images [12].

The said models were tested using only a desktop computer. In real-time monitoring of mask-wearing compliance, portable devices with relatively lower processing capabilities may be used for convenience. Moreover, only single-face images were considered in both model training and testing. In public settings, multiple faces may be captured by the mask detection models at a time.

With these, the study then aims to evaluate and compare the performance of the developed mask detection models on both desktop and the embedded platform Raspberry Pi. Multi-face mask detection will also be explored by testing the models on combined single-face images from the validation and test sets. It builds on [12] by making the following contributions:

- The real-time performance of robust mask detection models considerate of both frontal and angled faces are examined through deployment on a Raspberry Pi. This helps determine the viability of using the models for live monitoring on low-power systems.
- A comparative analysis of the models' mask detection capabilities in a desktop computer and a Raspberry Pi is performed. This results in insights into the strengths and limitations of the existing models based on the computational power of the hardware used.
- The models' performance in detecting multiple masked faces is observed. This helps identify the

suitability of using the models for simultaneous mask detection in crowded settings.

This paper contains the following sections: Section II provides a discussion on the object detection architectures used by the developed models, the robust detection models from [12] involved in this study, and related works on real-time and multi-face mask detection. Section III elaborates on the methods employed in this study, particularly multi-face inference and real-time implementation on a Raspberry Pi. Section IV presents a description of the results and their detailed analysis. Lastly, Section V summarizes the study's findings and offers recommendations on potential future developments.

II. Related Work

A. Object Detection Architectures

In developing the models involved in this study, the state-of-the-art object detection architectures CenterNet and YOLO, particularly YOLOv5, were used.

CenterNet [13] is characterized by anchorless object detection, which implements a more time- and resource-efficient algorithm in place of the standard Non-Maximum Suppression (NMS) technique. Under this approach, objects are modeled as a single point, corresponding to the center of the bounding box. Searching of center points is done through keypoint estimation, while determining other object properties including the size, location, and pose involves regression. In evaluating the relevance of bounding box predictions, the CenterNet architecture focuses on where their centers are located rather than how much they overlap with the object being detected. Compared to anchor-based detectors, fewer irrelevant detections are generated by CenterNet, resulting in faster inference and less power usage.

On the other hand, YOLO (You Only Look Once) is a state-of-the-art framework that generally operates by dividing an input image into grids, with object detection taking place in each grid. This approach boasts remarkable speed and efficient consumption of computational resources. YOLOv5 [14], one of the latest versions of YOLO, mostly differs from its predecessors in terms of the model backbone, neck, and head. In charge of image feature extraction, a combination of Cross Stage Partial Networks (CSPNet) [15] and Darknet termed CSPDarknet serves as the backbone of YOLOv5. With CSPNet, the issue of duplicate gradients common in large-scale backbones gets resolved, resulting in increased inference speed and reduced model size due to the decline in parameters and floating-point operations per second. Path Aggregation Network (PANet) [16] functions as the model neck, used to create feature pyramids for aggregating and passing features to the model head. It implements a novel feature pyramid structure with an improved bottom-up path that allows for better low-level feature propagation. Overall, PANet helps in locating objects more accurately. For generating predictions and bounding boxes, YOLOv5 retains the model head utilized by YOLOv4, which can perform multi-scale detection [17]. The activation and loss functions of YOLOv5 also set it apart from older YOLO models and contribute to its faster learning and enhanced performance. To deal with the vanishing gradient problem, it uses Leaky Rectified Linear Unit and Sigmoid activation functions. For the loss function, it employs the Binary Cross-Entropy with Logits Loss function.

B. Mask Detection Models and Dataset

As previously stated, this study makes use of the object detection models from [12] which include YOLOv5 Small, YOLOv5 Medium, CenterNet Resnet50 V1 FPN 512x512, and CenterNet HourGlass104 512x512. These models were trained for the image classification task on the relabeled Face Mask Label Dataset (FMLD) curated in [12]. The relabeled FMLD is made to train deep learning models in detecting three mask-wearing classifications, which are Medical Masks, Non-Medical Masks, and No Mask, in consideration of front and side view face images. With this, the six classes represented in the dataset are Front - Medical Mask, Front - Non-Medical Mask, Front - No Mask, Side - Medical Mask, Side - Non-Medical Mask, and Side - No Mask. In the relabeled FMLD, there are 50 images per class which sum up to a total of 300 images. In relation to this, the training of each object detection model involved 300 epochs. The classification accuracies of the models on the test set of the relabeled FMLD were then measured and compared. The models with the highest accuracy were found to be the CenterNet Resnet50 and CenterNet HourGlass104 models, both having an overall accuracy of 95%. The YOLOv5 Medium comes next with an overall accuracy of 93.33%, and the YOLOv5 Small model is the least accurate with 91.67%.

C. Similar Works

The study of ben Abdel Ouahab et al. [18] aimed to develop a mask detection model and evaluate its real-time performance on the Raspberry Pi. Fine-tuning was performed on the pre-trained MobileNetV2 model, constructing a new classification head with five layers: average pooling, flatten, dense with ReLU activation, dropout, and dense with softmax activation. The model was trained on a dataset with 1915 masked and 1918 unmasked face images. It achieved an accuracy of 99% upon testing on the validation set. For real-time implementation, two high-performing laptops, Raspberry Pi 3 and 4 devices, and Raspberry 4 devices with Intel Neural Compute Stick (NCS) 2 were used. The model obtained the highest average FPS value of 4.8 on the Raspberry Pi 4 (8 GB RAM) with NCS 2 among the different versions. It was observed that the performance of the model in terms of speed decreased significantly when deployed on low-power systems rather than on desktop devices.

Moreover, Mohandas et al. [19] focused on creating a real-time face mask detection system running on edge computing devices for access and egress control. For model training, transfer learning was employed on the SSD InceptionV2 model using a GPU-accelerated device. The dataset used for re-training consisted of images from the Real-World Masked Faces Dataset (RMFD), Labelled Faces in the Wild (LFW) dataset, and various web resources. Upon implementation on a Raspberry Pi 4 device, an average detection time of 1.13 ms per frame was obtained. The model also achieved perfect precision for both classes, 89% recall for masked faces, and 91% recall for unmasked faces on the Raspberry Pi 4, which is comparable to its performance on the GPU-enabled computer. While the system is meant to detect only one face at a time, restricting

the face to a certain Region of Interest, it was found to be effective for detecting multiple masked or unmasked faces as well.

Lastly, the study of Reza et al. [20] investigated the face mask detection performance of selected Convolutional Neural Network (CNN) models on mobile IoT devices. New classifiers were added to the MobileNetV2, InceptionV3, VGG16, and ResNet50 models and trained on top of their frozen layers. Model training involved a public dataset containing both single- and multiple-face images of masked and unmasked people. The models were trained four times, using varying ratios of the training data, and tested accordingly on the NVIDIA Jetson TX2 and Jetson Nano. VGG16 had the highest average accuracy across all ratios and for both devices, reaching a peak of 96.07% for 20% training data, but obtained the slowest inference speed. On the other hand, MobileNetV2 performed the best in terms of speed, having the least inference time of 25.10 ms for 5% and 10% training data, but lagged behind in testing accuracy. InceptionV3 achieved promising accuracy results upon being trained and tested on the smallest dataset ratio.

III. Methodology

This study used two general sets of methods, which are Multi-Face Inference and Real-Time Implementation in Raspberry Pi. In the corresponding subsections, the procedures performed are discussed in greater detail.

A. Multi-Face Inference

To evaluate the multi-face mask detection performance of the models from [12], images with two or three faces must be included in the test set for this study. Since the mask detection models in [12] were trained and tested on the relabeled FMLD, the same dataset was used in building the test set. Single-face images from the validation and test sets in [12] were combined through the image editing tool Photopea to synthetically produce images that contain two or three faces. The images used per multi-face combination came from random classes, but the equal representation of all six classes among the test images with two or three faces was ensured as far as possible. Black pixels were used to fill the empty spaces left in the synthetic multi-face images which were caused by the varying dimensions of component images. Fig. 1 and 2 present sample test images with two and three faces, respectively.

Using the four mask detection models from [12], inferencing was done on the multi-face images created from the relabeled FMLD. The classification accuracy for each class and the overall classification accuracy were recorded. Moreover, the inference speed was examined to describe the relationship between the number of faces in an image and the speed at which the models classify the image, if any. First, baseline speeds for all four models were determined through the inferencing of single-face images from the validation and test sets used in [12] and the computation of average inference time per image. Then, similar steps were carried out to measure the speeds for inferencing two-face and three-face images using the four models. For the YOLOv5 models, the speeds were automatically printed out after the completion of inference.

Fig. 1. Sample image containing two faces.

Fig. 2. Sample image containing three faces.

Conversely, calculating the inference speed for the CenterNet models was done using the timeit package for Python. The desktop computer utilized for testing was equipped with an NVIDIA GTX 1080 Ti GPU.

For further verification of the mask detection models' classification accuracies on the images with two or three faces, the single-face images that comprise the synthetic multi-face images went through individual inferencing. This process served as a way of confirming that any difference in classification accuracy between an image with a single face and one with multiple faces can be primarily attributed to the number of faces in an image instead of other possible factors. In individually inferencing the faces present in the test images containing two or three faces, the classification accuracy for each class and the overall classification accuracy were recorded.

B. Real-Time Implementation in Raspberry Pi

To test the capabilities of the mask detection models from [12] on lower resource machines, they were exported and run on an embedded system, and their performance in terms of accuracy and speed was measured. This process is aimed at assessing the feasibility of deploying the models on portable or mobile platforms for increased accessibility and convenience of use.

Initially, it was planned to export all four models from [12] to a Raspberry Pi 4 Model B with 4GB of RAM for testing procedures. Unfortunately, there were compatibility issues with the Tensorflow 2 Object Detection API, which facilitated the use of the CenterNet models. Transferring and running the said models on the Raspberry Pi turned out to be challenging due to the format in which they were exported by the Tensorflow 2 Object Detection API. While it is possible to run the CenterNet models on the Raspberry Pi, it was not done successfully for this study. In the end, only the YOLOv5 Small and Medium models were run and tested on the Raspberry Pi. The CenterNet models were excluded since it was taking a significant amount of time to resolve the problems.

In testing the YOLOv5 models on the Raspberry Pi, the classification accuracies were recorded, so that the performance in terms of accuracy when running on a desktop computer and an embedded system may be compared. This was performed using the same methods from the testing phase in [12] and Section III A as well. Inferencing with the YOLOv5 models running on the Raspberry Pi device was performed on the combined validation and test set images from [12]. The class with the highest predicted confidence score was determined to be the predicted class for each image. To compute the accuracy per class in percentage value, the number of correct predictions was divided by the total number of images per class, and the resulting quotient was then multiplied by 100. Python was used for inferencing and calculating the respective classification accuracies of each mask detection model. In particular, the PyTorch package was instrumental in inferencing with the YOLOv5 models.

Aside from their classification accuracies, the inference speeds of the YOLOv5 models upon testing on the Raspberry Pi were also recorded. This process served as a way of determining if the processing speeds of the YOLOv5 models on a low-end computing device fall within an acceptable range. This was performed using similar procedures from Section III A. Measuring of speeds took place while inferencing images with one, two, and three faces from the combination of the validation and test sets used in [12].

IV. Results and Discussion

A. Multi-Face Inference

A total of 20 synthetic multi-face images were produced from the dataset preparation stage of Section III-A, having ten images with two faces each and another ten images containing three faces each. Table I details the distribution of the six classes found in the relabeled FMLD from [12] among the two-face and three-face images.

Fig. 3, 4, 5, and 6 present the multi-face classification accuracies measured from the YOLOv5 Small, YOLOv5 Medium, CenterNet Resnet50, and CenterNet HourGlass104 models, respectively. Each figure consists of three bar graphs that show the corresponding mask detection model's accuracies in classifying images with one, two, and three faces. The accuracies for each class present in the relabeled FMLD are specified, as well as the overall accuracy per model. The labels for classification accuracies come in the form of percentages and fractions showing the number of correct predictions over the total of possible predictions. The dotted line found in each figure represents the overall accuracy per model, which is just another way of displaying the values from the bar for overall accuracy. It helps create a better visualization of the changes in accuracy while detecting images with varying numbers of faces. The values for the Single Face Accuracy section per model were taken from the results obtained in [12]. From the figures, it can be observed that the overall accuracy for all models tends to decrease as the number of faces present in an image increases. The YOLOv5 models perform with similar levels of accuracy when classifying two-face images, only suffering from reduced overall accuracies when classifying three-face images.

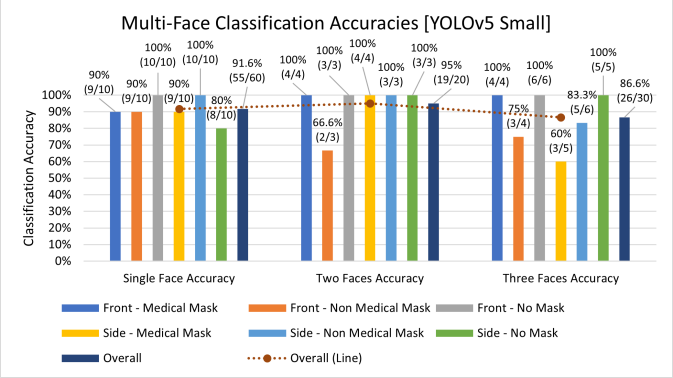

Fig. 3. Multi-face classification accuracies of the YOLOv5 small model.

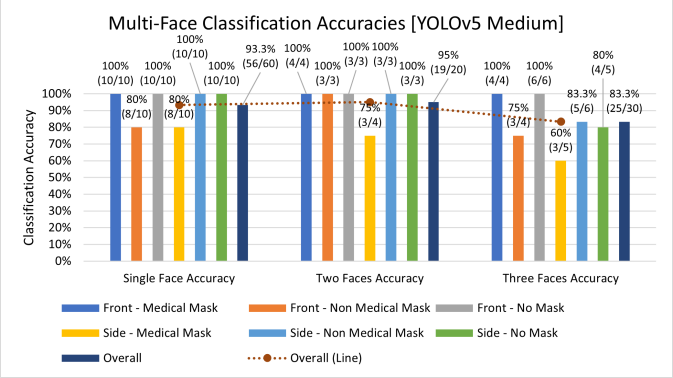

Fig. 4. Multi-face classification accuracies of the YOLOv5 medium model.

Going from detecting single-face images to detecting three-face images, the YOLOv5 Small and YOLOv5 Medium models incur an estimated 5% and 10% loss in overall accuracy, respectively. On the other hand, the CenterNet models are more prone to losses in overall accuracies upon classifying images with an increasing number of faces. Going from detecting single-face images to detecting two-face images, both

TABLE I. DISTRIBUTION OF THE SIX CLASSES OF THE RELABELED FMLD FOR THE IMAGES CONTAINING TWO OR THREE FACES

Class	Number of Faces	
	Images with Two Faces	Images with Three Faces
Front - Medical Mask	4	4
Front - Non-Medical Mask	3	4
Front - No Mask	3	6
Side - Medical Mask	4	5
Side - Non-Medical Mask	3	6
Side - No Mask	3	5
Total	20	30

Note: Ten images containing two faces each equals 20 faces in total.
Similarly, ten images containing three faces each equals 30 faces in total.

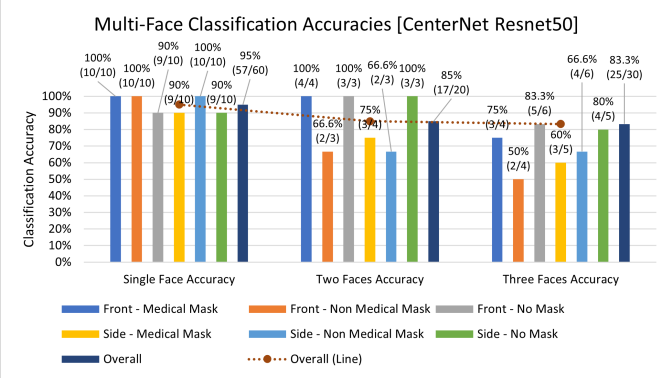

Fig. 5. Multi-face classification accuracies of the CenterNet Resnet50 model.

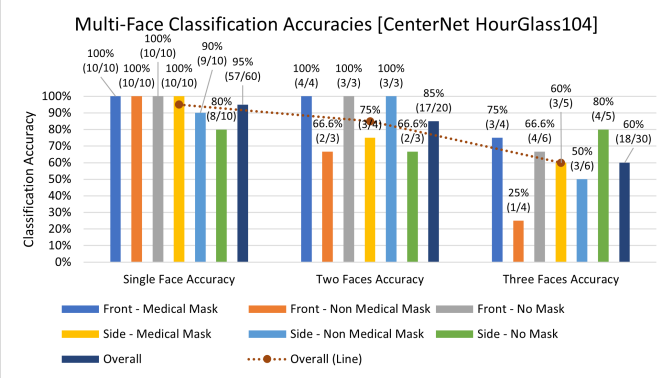

Fig. 6. Multi-face classification accuracies of the CenterNet HourGlass104.

CenterNet models suffer from an approximated 10% loss in overall accuracy. Moving further to detecting images with three faces, the CenterNet Resnet50 model somehow obtains the same level of accuracy compared to its performance on two-face image detection. Conversely, the CenterNet HourGlass104 model incurs an additional 25% decline in overall accuracy, accumulating an estimated total loss of 35% in overall accuracy when classifying three-face images compared to when classifying single-face images. Among all mask detection models, the CenterNet HourGlass104 model experiences the greatest decline in overall accuracy upon classifying images with multiple faces. This can be possibly attributed to the susceptibility of CenterNet models to overfitting as discussed in the model training procedure in [12]. Overfitting makes it difficult for the said models to perform well on inputs that differ from those that they were trained on, which are single-face images.

Fig. 7 presents the different inference speeds of the models on images with one, two, and three faces. From the graph, it can be observed that the fastest mask detection model, having the least inference time in milliseconds, is the YOLOv5 Small model, followed by the YOLOv5 Medium, then the CenterNet Resnet50, and finally the CenterNet HourGlass104. The arrangement of the models in increasing inference speed corresponds to their arrangement in increasing network size, making the former quite expected. Being the smallest of the four models, the YOLOv5 Small model is relatively computationally lightweight and thus faster to execute. Conversely, the CenterNet HourGlass104 model is computationally intensive and thus slower to run, since it has the largest network size among the models. In testing multi-face detection, the models' inference speeds incur an initial decrease upon detecting two-face images compared to when detecting single-face images. However, there is negligible change in the speeds of the models when going from detecting two-face images to detecting three-face images. Based on these results, classifying images with even more faces may not have significant effects on the models' inference speeds. It is also important to note that the speeds specified in Fig. 7 were recorded during the first round of inference on the test images. Upon repeated inferencing in several rounds, the speeds measured turned out to be faster, but such speeds were no longer considered.

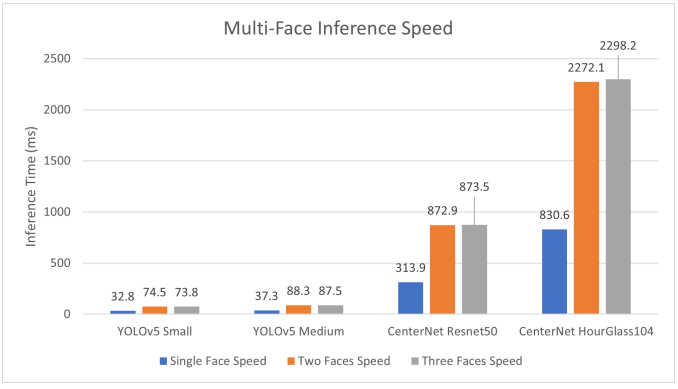

Fig. 7. Multi-face inference speeds of all of the four models of this study.

Furthermore, the multi-face inference speeds of the YOLOv5 models, which were also presented in Fig. 7, were shown in more detail in Fig. 8. From the graphs, it can be

seen that the inference times were dominated by the inferencing process itself, while the pre-process and Non-Maximum Suppression (NMS) stages took up only small portions. The YOLOv5 models automatically generated the speed breakdown after the inferencing process. Since the CenterNet models have no similar capability, their inference speed breakdown did not get included anymore.

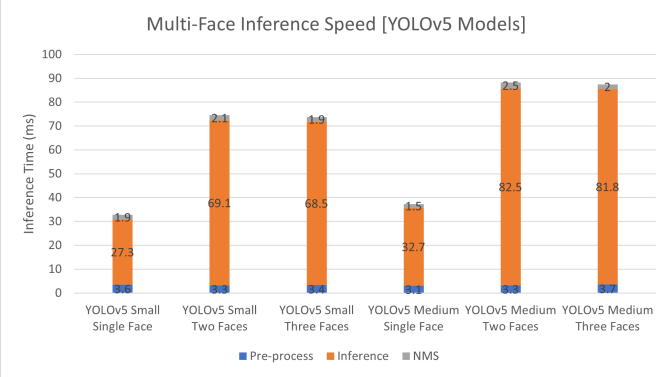

Fig. 8. Breakdown of the multi-face inference speeds of the YOLOv5 models.

Fig. 9, 10, 11, and 12 show the results for the validation of the multi-face classification accuracies of the YOLOv5 Small, YOLOv5 Medium, CenterNet Resnet50, and CenterNet HourGlass104 models, respectively. Each figure is divided into two cluster groups, the first group is for results on two-face images while the second one is for results on three-face images. The first cluster in each group presents the classification accuracies of the corresponding model when individually inferencing the faces that comprise the synthetic multi-face images. On the other hand, the second cluster in each group presents the model's accuracies upon inferencing the merged images with two or three faces themselves. The results verify that the differences in accuracy when classifying images with one face and those with multiple faces are caused by the changes in the number of faces contained in them. For most of the mask detection models, their classification accuracies were higher when single-face images were inferenced individually and not as components of a synthetic multi-face image. However, the results obtained from the YOLOv5 Small model deviate from the general observation, as its classification accuracies were found to be higher when inferencing images with multiple faces compared to those with only one face. These further support the earlier findings about YOLOv5 models being able to classify two-face images with similar levels of accuracy and only obtaining reduced overall accuracies upon classifying three-face images. The discussion on CenterNet models being more prone to declines in accuracy due to an increase in the number of detected faces also gets further confirmed.

B. Real-Time Implementation in Raspberry Pi

Table II shows the classification accuracies of the YOLOv5 models when tested on the Raspberry Pi. These accuracies are for images with single faces only, as the combined validation and test set images from [12] are used without making modifications. The table presents a direct comparison between the model's classification accuracy on the desktop computer and its accuracy upon running on the Raspberry Pi. Each table cell corresponds to a particular mask detection model and image class, containing the ratio of the number of correctly predicted images from the class by the model to the total number of images in the class and the equivalent accuracy in percentage form.

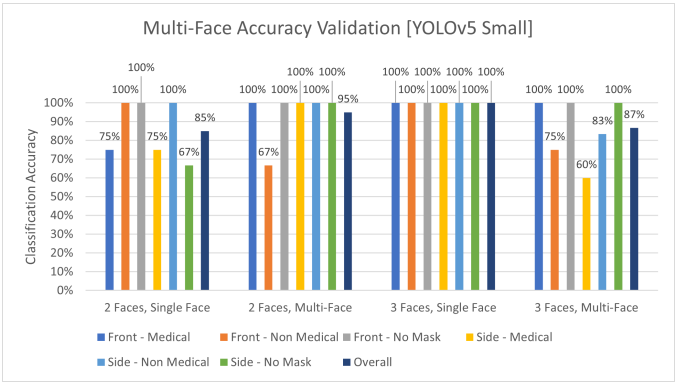

Fig. 9. Validation of the Multi-Face classification accuracies of the YOLOv5 small model.

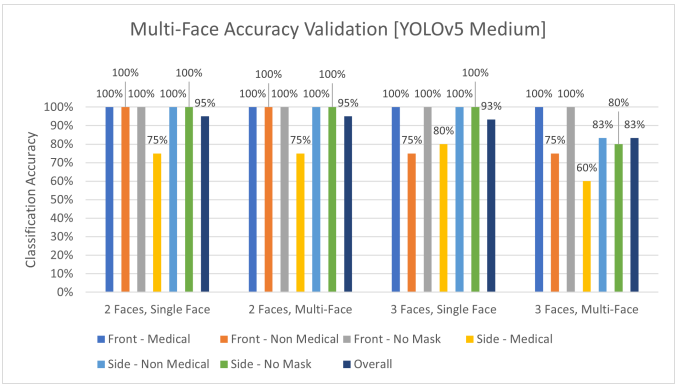

Fig. 10. Validation of the Multi-Face classification accuracies of the YOLOv5 medium model.

From these results, the differences in classification accuracies of the YOLOv5 Small and Medium models when tested on

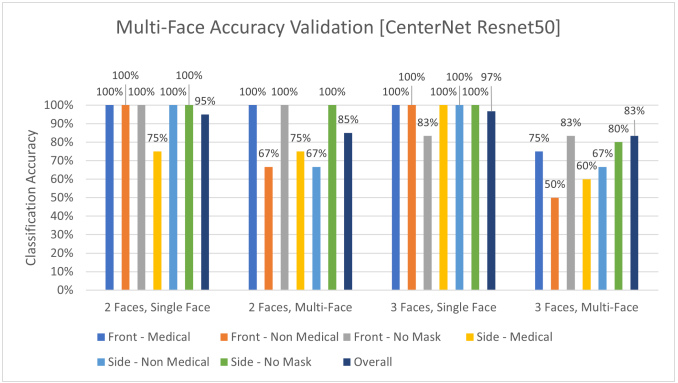

Fig. 11. Validation of the Multi-Face classification accuracies of the CenterNet Resnet50 model.

TABLE II. COMPARISON OF CLASSIFICATION ACCURACIES ON DESKTOP VERSUS ON RASPBERRY PI

Model	Front - Medical Mask	Front - Non Medical Mask	Front - No Mask	Side - Medical Mask	Side - Non Medical Mask	Side - No Mask	Overall
YOLOv5 Small (Desktop)	9/10 Accuracy = 90%	**9/10 Accuracy = 90%**	**10/10 Accuracy = 100%**	9/10 Accuracy = 90%	**10/10 Accuracy = 100%**	8/10 Accuracy = 80%	55/60 Accuracy = 91.67%
YOLOv5 Small (Raspberry Pi)	9/10 Accuracy = 90%	**9/10 Accuracy = 90%**	**10/10 Accuracy = 100%**	**10/10 Accuracy = 100%**	**10/10 Accuracy = 100%**	9/10 Accuracy = 90%	**57/60 Accuracy = 95%**
YOLOv5 Medium (Desktop)	**10/10 Accuracy = 100%**	8/10 Accuracy = 80%	**10/10 Accuracy = 100%**	8/10 Accuracy = 80%	**10/10 Accuracy = 100%**	**10/10 Accuracy = 100%**	56/60 Accuracy = 93.33%
YOLOv5 Medium (Raspberry Pi)	9/10 Accuracy = 90%	7/10 Accuracy = 70%	**10/10 Accuracy = 100%**	8/10 Accuracy = 80%	9/10 Accuracy = 90%	8/10 Accuracy = 80%	51/60 Accuracy = 85%

Bold = Highest Value in Column

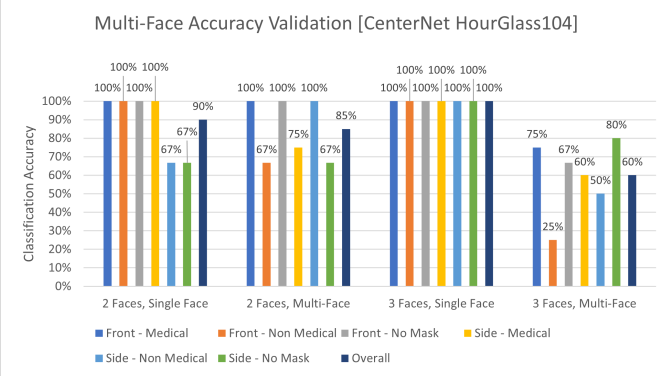

Fig. 12. Validation of the Multi-Face classification accuracies of the CenterNet HourGlass104 model.

the desktop computer and the Raspberry Pi can be examined. The YOLOv5 Small model obtained a greater classification accuracy on the Raspberry Pi than on the desktop computer, being able to correctly predict two additional images belonging to the Side - Medical Mask and Side - No Mask classes. The opposite was true for the YOLOv5 Medium model, as it achieved a lower classification accuracy when running on the Raspberry Pi than on the desktop computer. There were fewer correctly predicted images under the Front - Medical Mask, Front - Non-Medical Mask, Side - Non-Medical Mask, and Side - No Mask classes. Overall, there was a minimal difference of 3.33% for the classification accuracies of the YOLOv5 Small model and a larger discrepancy of 8.33% for the accuracies of the YOLOv5 Medium model.

Varying performance metric values for the same deep learning models when run on the Raspberry Pi and on other platforms have also been recorded in [21][22][23]. Unfortunately, the said studies could not offer an explanation behind the discrepancy in accuracies or confidence scores upon testing on different devices, including the Raspberry Pi. Similarly, this study failed to come up with reasons for the difference in the mask detection models' classification accuracies on the desktop computer and on the Raspberry Pi.

Moreover, Fig. 13 presents the inference speeds of the YOLOv5 models upon testing on the Raspberry Pi. In general, the inferencing of YOLOv5 models took a longer time on the Raspberry Pi than on the desktop computer. The inference times of the models on the Raspberry Pi were divided by their inference times on the desktop computer to obtain the speedup values. For the YOLOv5 Small model, it performed 68.75, 29.09, and 27.99 times faster on the desktop computer than on the Raspberry Pi in inferencing images containing one, two, and three faces, respectively. On the other hand, the YOLOv5 Medium model obtained speedup values of 137.28, 57.83, and 59.08 upon inferencing single-face, two-face, and three-face images, respectively. Both models experienced the highest speedups with single-face images, which aligns with how the YOLOv5 Small and Medium models had lower inference times on the desktop computer when inferencing images with one face than those with two or three faces. Meanwhile, the YOLOv5 Small and Medium models had inference times of similar levels for single-face, two-face, and three-face images on the Raspberry Pi.

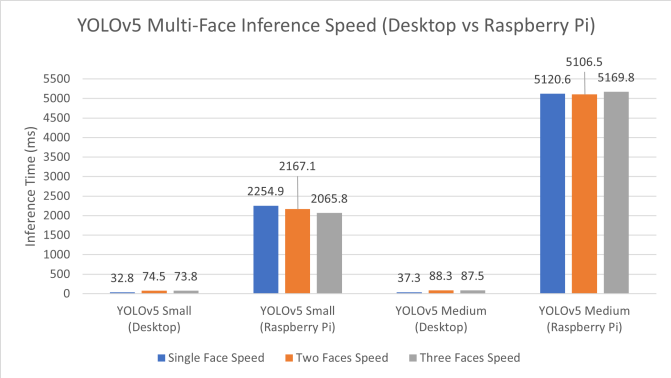

Fig. 13. Multi-face inference speeds of the YOLOv5 models when ran on the desktop machine versus when ran on the Raspberry Pi.

Overall, the YOLOv5 Small model turns out to be the most ideal for deployment on low-end computing devices. There was only a small difference in the classification accuracies of the YOLOv5 Small model when running it on the Raspberry Pi and on the desktop computer. As seen in Table II, the YOLOv5 Small model obtained the highest overall classification accuracy when tested on the Raspberry Pi. Furthermore, less discrepancy in the said model's inference speeds can be observed upon inferencing on the desktop computer then on the Raspberry Pi. The YOLOv5 Small model achieved an inference time of about two seconds per image on the Raspberry Pi, still falling within the acceptable range. Conversely, the YOLOv5 Medium model obtained an inference time of about five seconds per image, already considered to be somehow too slow. It also recorded a greater difference in inference speeds compared to the YOLOv5 Small model when inferencing on the desktop computer and on the Raspberry Pi.

V. Conclusion

Generally, the mask detection models were found to be capable of multi-face detection, although the accuracies were observed to decline in the presence of more faces in an image. In terms of inference speed, the YOLOv5 models performed faster than the CenterNet models due to their smaller network sizes. Moreover, it was observed that the number of faces in an image did not significantly affect the models' inference speeds.

For the implementation on the Raspberry Pi 4 Model B embedded platform, only the YOLOv5 Small and Medium models were used. Going from inferencing on the desktop computer to the Raspberry Pi, the YOLOv5 Small model experienced a 3.33% increase in classification accuracy, while the YOLOv5 Medium model suffered from an 8.33% decrease in accuracy. In terms of inference speed, the YOLOv5 models generally exhibited a slower mask detection performance on the Raspberry Pi than on the desktop computer, with the YOLOv5 Small model having an inference time of about two seconds per image and the YOLOv5 Medium model obtaining an inference time of about five seconds per image. Furthermore, both models recorded the highest speedup values when inferencing single-face images, just as they achieved lower inference times with single-face images on the desktop computer compared to multi-face images. Considering both the classification accuracy and inference speed, the YOLOv5 Small model was regarded to be potentially the best model for use on lower resource platforms.

Future work may involve the use and evaluation of other object detection models for the robust face mask detection task. Developed models can also be tested on several kinds of embedded systems, such as the Jetson Nano. It might also be worth looking into the addition of more mask-wearing categories, including incorrectly masked faces, among others.

References

[1] Petersen, E., Petrosillo, N., Koopmans, M., Beeching, N., Di Caro, A., Gkrania-Klotsas, E., Kantele, A., Kohlmann, R., Koopmans, M., Lim, P.-L., Markotic, A., López-Vélez, R .,Poirel, L., Rossen, J., Stienstra, Y., and Storgaard, M. Emerging infections—an increasingly important topic: review by the emerging infections task force. *Clinical Microbiology and Infection 24*, 4 (2018), 369–375.

[2] Sarmah, P., Dan, M., Adapa, D., AND TK, S. A review on common pathogenic microorganisms and their impact on human health. *Electronic Journal of Biology 14* (April 2018).

[3] Çelik, I., Saatçi, E., and Eyüboğlu, A. Emerging and reemerging respiratory viral infections up to covid-19. *Turkish journal of medical sciences 50* (April 2020).

[4] WHO. Statement on the fifteenth meeting of the IHR (2005) Emergency Committee on the COVID-19 pandemic, May 2023. Retrieved June 25, 2023 from https://www.who.int/news/item/05-05-2023-statement-on-the-fifteenth-meeting-of-the-international-health-regulations-(2005)-emergency-committee-regarding-the-coronavirus-disease-(covid-19)-pandemic.

[5] WHO. WHO Director-General's opening remarks at the media briefing on COVID-19 - 11 March 2020. *World Health Organization* (March 2020).

[6] Shereen, M. A., Khan, S., Kazmi, A., Bashir, N., and Siddique, R. Covid-19 infection: Emergence, transmission, and characteristics of human coronaviruses. *Journal of Advanced Research 24* (2020), 91–98.

[7] DOH. COVID-19 Tracker. Report, Department of Health (Philippines), September 2022.

[8] WHO. Overview of public health and social measures in the context of COVID-19: interim guidance, 18 May 2020. Technical documents, World Health Organization, 2020.

[9] WHO. Coronavirus disease (COVID-19): Masks. *World Health Organization* (January 2022).

[10] CDC. Use and Care of Masks. *Centers for Disease Control and Prevention* (September 2022).

[11] Choudhary, O. P., Priyanka, Singh, I., and Rodriguez-Morales, A. J. Second wave of covid-19 in india: Dissection of the causes and lessons learnt. *Travel Medicine and Infectious Disease 43* (2021), 102126.

[12] Tarun, I., Lopez, V., Abu, P., Estuar, M. Robust Face Mask Detection with Combined Frontal and Angled Viewed Faces. In *Proceedings of the 24th International Conference on Enterprise Information Systems (ICEIS 2022) 1* (2022), pp. 462-470.

[13] Zhou, X., Wang, D., and Krähenbühl, P. Objects as points. *ArXiv abs/1904.07850* (2019).

[14] Jocher, G., Chaurasia, A., Stoken, A., Borovec, J., NanoCode012, Kwon, Y., TaoXie, Fang, J., imyhxy, Michael, K., Lorna, V, A., Montes, D., Nadar, J., Laughing, tkianai, yxNONG, Skalski, P., Wang, Z., Hogan, A., Fati, C., Mammana, L., AlexWang1900, Patel, D., Yiwei, D., You, F., Hajek, J., Diaconu, L., and Minh, M. T. ultralytics/yolov5: v6.1 - TensorRT, TensorFlow Edge TPU and OpenVINO Export and Inference, Feb. 2022.

[15] Wang, C.-Y., Mark Liao, H.-Y., Wu, Y.-H., Chen, P.-Y., Hsieh, J.-W., AND Yeh, I.-H. Cspnet: A new backbone that can enhance learning capability of cnn. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (2020), pp. 1571–1580.

[16] Liu, S., Qi, L., Qin, H., Shi, J., and Jia, J. Path aggregation network for instance segmentation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2018), pp. 8759–8768.

[17] Xu, R., Lin, H., Lu, K., Cao, L., and Liu, Y. A forest fire detection system based on ensemble learning. *Forests 12* (02 2021), 217.

[18] ben Abdel Ouahab, I., Elaachak, L., Bouhorma, M., and Alluhaidan, Y. A. Real-time Facemask Detector using Deep Learning and Raspberry Pi. In *2021 International Conference on Digital Age & Technological Advances for Sustainable Development (ICDATA)* (2021), pp. 23-30.

[19] Mohandas, R., Bhattacharya, M., Penica, M., Van Camp, K., and Hayes, M.J. On the use of Deep Learning Enabled Face Mask Detection For Access/Egress Control Using TensorFlow Lite Based Edge Deployment on a Raspberry Pi. In *2021 32nd Irish Signals and Systems Conference (ISSC)* (2021), pp. 1-6.

[20] Reza, S. R., Dong, X., and Qian, L. Robust Face Mask Detection using Deep Learning on IoT Devices. In *2021 IEEE International Conference on Communications Workshops (ICC Workshops)* (2021), pp. 1-6.

[21] Feng, H., Mu, G., Zhong, S., Zhang, P., and Yuan, T. Benchmark analysis of yolo performance on edge intelligence devices. *Cryptography 6*, 2 (2022).

[22] Sabri, Z. S., and Li, Z. Low-cost intelligent surveillance system based on fast cnn. *PeerJ Computer Science 7* (Feb 2021), e402.

[23] Süzen, A. A., Duman, B., and Şen, B. Benchmark analysis of jetson tx2, jetson nano and raspberry pi using deep-cnn. In *2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)* (2020), pp. 1–5.