

# FACIAL MASK DETECTOR USING STATE-OF-THE-ART OBJECT DETECTION MODELS

*Ang Kelvin*

Faculty of Computing  
and Informatic, MMU

*Chang Kai Boon*

Faculty of Computing  
and Informatic, MMU

*Soe Zhao Hong*

Faculty of Computing  
and Informatic, MMU

*Pritesh Patel*

Faculty of Computing  
and Informatic, MMU

## ABSTRACT

The ongoing COVID-19 pandemic has caused a major problem in the world. Experts have suggested wearing a face mask to combat the spread of covid19. This motivates us to look into face mask detection technology to detect if a person is wearing a face mask. The dataset we are using is obtained from kaggle with 853 annotated images with mask, unmask and incorrect mask classes. In this research, we will study four object detection models: Faster R-CNN, SSD, YOLOv5 and EfficientDet. We trained the model with 100 epochs, varying amounts of batch size and different image resolution. From the result, we found that YOLOv5s managed to perform best in each scenario and the best condition for YOLOv5s model was image size of 320x320, batch size 16 which managed to achieve a mean average precision (mAP) of 87.5%.

## 1. INTRODUCTION

In recent years, the pandemic caused by Coronavirus has become a major issue worldwide. The main way for the virus to spread is through air. COVID-19 transmits when people breathe in air contaminated by droplets and small airborne particles containing the virus. The risk of breathing these in is highest when people are in close proximity, but they can be inhaled over longer distances, particularly indoors. Transmission can also occur if splashed or sprayed with contaminated fluids in the eyes, nose or mouth, and, rarely, via contaminated surfaces according to the World Health Organization. A main way to combat the virus is by wearing a face mask.

For this research, our motivation is to apply computer vision and visual image processing techniques to detect if a person is wearing a face mask, not wearing a face mask or wearing a face mask incorrectly. We find this problem an interesting way to contribute to help combat the spread of COVID-19 virus. With our research, we can develop models to help detect which civilian is wearing a face mask, not wearing a face mask and wearing it incorrectly. We can use this model to help enforce mask wearing mandates in a country.

We will be comparing four models which are YOLOv5, single shot detector (SSD), ResNet, Faster R-CNN and EfficientDet. We will try to find the best model that suits this problem the best and list out the pros and cons of each model.

## 2. OBJECT DETECTION MODELS AND CNN BACKBONE ARCHITECTURE

**Faster R-CNN** - Region-Based Convolutional Neural Networks family, as known as R-CNNs, are a group of object detection and semantic segmentation models which consists of R-CNN, Fast R-CNN, Faster R-CNN and Mask R-CNN. R-CNN was first introduced by (Girshick, R et al, 2013) [4] to use selective search for identifying bounding box object region candidates and the CNN features are extracted from each region independently for classification and detection. To make R-CNN faster, (Girshick, R, 2015) [5] proposed a jointly trained framework which increased the shared computation and named it Fast R-CNN. In (Ren, S. et al, 2015), the authors proposed a single unified network called Faster R-CNN for object detection. The model is composed of two modules. In the first module, a Region Proposal Network (RPN) was proposed to generate region proposals and in the second module, the Fast R-CNN used the proposed regions for object detection. Using the recently popular terminology with attention mechanism, the RPN tells the Fast R-CNN where to look for. With such architecture, Faster R-CNN achieved state-of-the-art performance on PASCAL VOC 2007, 2012 and MS COCO datasets.

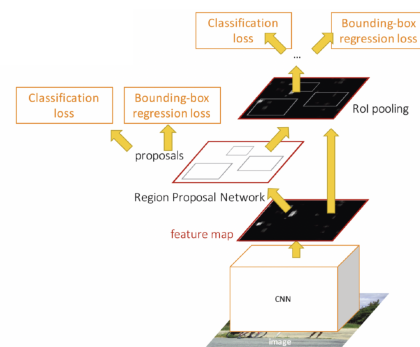


Figure 1: Architecture of Faster RCNN

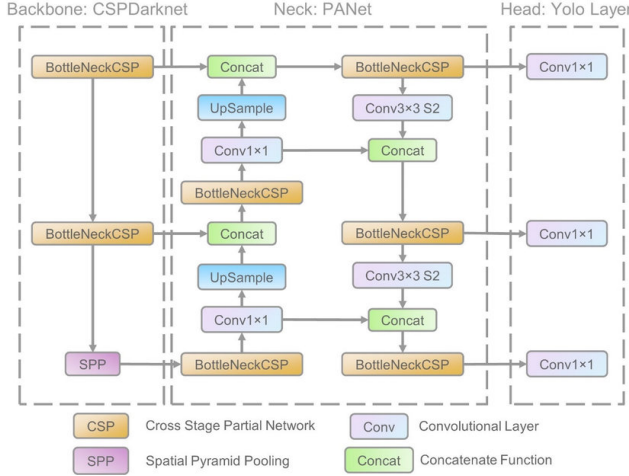


Figure 2: Architecture of YOLOv5

**YOLOv5-** The YOLO family of models consists of 3 main architectural blocks which are Backbone, Neck, and Head. The YOLOv5 Backbone will employ CSPDarknet as the backbone for feature extraction from images consisting of cross-stage partial networks. Then, the YOLOv5 Neck uses PANet to generate a feature pyramid network to perform aggregation on the features and pass it to the Head for prediction. Lastly, YOLOv5 Head is the layer that generates the predictions from the anchor boxes for object detection. Figure 2 shows the overall architecture of YOLOv5.

YOLOv5 has numerous assortments of pre-trained models as shown in figure 3 such as YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x. The difference between these models is the trade-off between the estimate of the demonstration and induction time. The lightweight model from YOLOv5s is just 14MB but not very precise. On the other side of the range, we have YOLOv5x whose estimate is 168MB but is the foremost exact form of its family. Although the YOLOv5s and YOLOv5m do not have a better performance compared to YOLOv5x, we only use the YOLOv5s and YOLOv5m for our proposed work because of the limitation on hardware to train other models of YOLOv5.





			
Small	Medium	Large	XLarge
YOLOv5s	YOLOv5m	YOLOv5l	YOLOv5x
14 MB <sub>FP16</sub>	41 MB <sub>FP16</sub>	90 MB <sub>FP16</sub>	168 MB <sub>FP16</sub>
2.0 ms <sub>V100</sub>	2.7 ms <sub>V100</sub>	3.8 ms <sub>V100</sub>	6.1 ms <sub>V100</sub>
37.2 mAP <sub>COCO</sub>	44.5 mAP <sub>COCO</sub>	48.2 mAP <sub>COCO</sub>	50.4 mAP <sub>COCO</sub>

Figure 3: Types of YOLOv5 and their details

**EfficientDet-** EfficientDet is built upon Efficientnet backbone and uses Bidirectional feature pyramid network (BiFPN). Efficientnet backbone uniformly scales all dimensions using a compound coefficient. The compound scaling method is justified by the intuition that if the input image is bigger, then the network needs more layers to increase the receptive field and more channels to capture more fine-grained patterns on the bigger image. Efficientdet has 3 section a backbone, feature network and box network. The backbone extracts the feature form the image, the feature network takes the features as input and output a fused feature that represent salient characteristics of the image and lastly the box network predicts the location of each object. BiFPN is a new feature network idea that incorporates the multilevel feature fusion idea from feature pyramid network PANet and NAS-FPN that allows flow in both top down and bottom up while using efficient connections. The overall architecture of EfficientNet is depicted in figure 4.

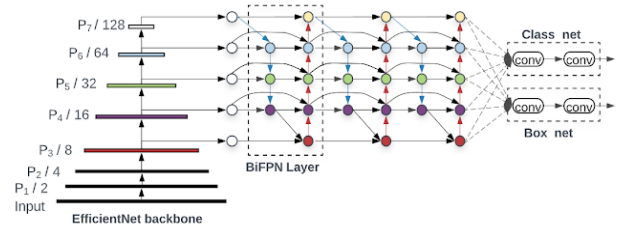


Figure 4: Architecture of EfficientNet

**SSD-** Single Shot MultiBox Detector, as known as SSD, a method for identifying objects in image which were to begin with presented by (Liu, W. et al, 2015) [16] discretizes the yield space of bounding boxes into a set of default boxes over distinctive viewpoint proportions and scales per highlight outline area. The network will produce the scores for the presence of each object category in each default box and make adjustments on the box to have better matching of the object shape during the prediction process. Other than that, the arrangement combines with the expectations to handle objects of diverse sizes. SSD kills proposition era and consequent pixel of include resampling stages and typifies all calculations in an organized manner which makes SSD simple to prepare. With such engineering, SSD accomplished state-of-the-art execution on PASCAL VOC, COCO, and ILSVRC datasets. Figure 5 shows the overall architecture of the SSD.

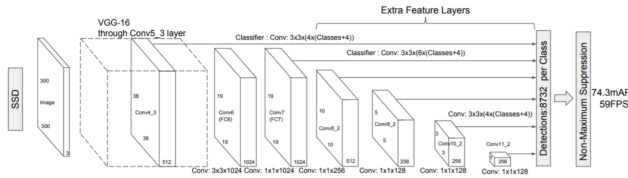


Figure 5: Architecture of SSD.

**ResNet-** Deep Residual Learning, as known as ResNet, a framework to ease the training of networks that are substantially deeper than used previously which were to begin with presented by (Kaiming He et al.) [9] The author unequivocally reformulate the layers as learning residual functions with reference to the layer inputs, rather than learning unreferenced functions. The author also provides comprehensive empirical evidence showing that these networks are simpler to optimize, and can pick up accuracy from significantly increased depth. On the ImageNet dataset the author assesses residual nets with a profundity of up to 152 layers—8×deeper than VGG nets but still having lower complexity. An outfit of these remaining nets accomplishes 3.57% error on the ImageNet test set. This result has the best performance put on the ILSVRC 2015 classification task. The author displays an investigation on CIFAR-10 with 100 and 1000 layers. The depth of representations is of central importance for numerous visual recognition tasks.

**MobileNet-** Howard et al., 2017 [8] proposed a new convolutional neural network catered for mobile vision applications which is known as MobileNet. MobileNet is implemented based on depth wise separable convolutions to create low-latency, lightweight deep neural networks for mobile and embedded devices. The depth wise separable convolutions will factor a normal convolution into 2 convolutions: depthwise convolution and a one times one pointwise convolution. A pointwise convolution will combine both the output from depth wise convolution. All of the input channels have a single depthwise convolution applied on it. The purpose of factoring this layer is to reduce the size and the computation resources required by this network. Howard et al., 2018[31] published an improved version of MobileNet, which they called MobileNetV2. The difference between MobileNet and MobileNetV2 is that MobileNetV2 has linear bottleneck layers inserted into convolutional layers to prevent activation functions from destroying too much information.

### 3. PREVIOUS WORK

**Yolov5-** In a research by Vinay Sharma., (2020)[25], performed face mask detection using yolo v5. The researcher used a labeled image with size of 224x224. The metrics used to evaluate the face mask detection model is precision, recall and mean average precision(MAP). In this research, the author compares two different yolo v5 models, yolo v5s and yolo v5x. The author found that both models had similar results in terms of precision, recall and MAP. However, yolo v5s had better performance in terms of speed. One problem faced by the proposed model is that it is unable to detect faces that are not directly facing the camera. One improvement the author suggests is to increase the amount of image data as input to include faces facing different angles.

Guan hao Yang et al., (2020)[6] proposed a system for people entering a mall. A picture will be taken of the person and an algorithm runs to enhance the image, segment the facial mask image and recognize if the person is wearing a mask or not. Based on whether the person is wearing a mask or not, the mall gate will open. The author claims that image enhancement is an important step because in real life, images are often affected by noise. So the main purpose of enchantment is to remove noise and extract useful information. This research used a dataset with 7959 images with class mask and no mask. This research adopts the method of combining GIOU loss and Center Loss to identify if a person is wearing a mask. Center loss reduces the distance of each datapoint from its class center and GIOU loss maximizes the overlap area of the ground truth and predicted boundary box. One flaw this research faces is that if a person is covering the face only partially and the nose is still exposed, the model predicts it as wearing a mask. Another limitation is that if the face is covered by hand and not an actual mask, the model predicts wearing a mask. Other than these limitations, the model has an accuracy of 97.9%

Jiarat et al., (2021) [10] compared the yolo v5 model for face mask detection with different numbers of epoch 20, 50, 100, 300 and 500. This research uses 853 image data with 3 labels, “With\_Mask”, “Without\_Mask” and “Incorrect\_Mask”. The data set was split into 682 training and 85 testing. It was found that models trained using 500 epochs had more desirable results than 20, 50 and 100 epochs. However, in terms of mean average precision, 300 epochs performed better than 500 epochs. From this research it show that increasing epoch does not necessarily improve the model

**Faster R-CNN-** In (Singh, S et al, 2021) [28], the authors implemented face mask detection using two state-of-the-art object detection models which are Faster R-CNN and YOLOv3 by applying transfer learning techniques. Due to

the unavailability of a large dataset, the authors have made a custom dataset of around 7.5k images with two classes : with mask and without mask. The dataset is composed of MAFA WIDER FACE and can be assessed publicly. For training the YOLOv3 model, the authors used the weights of a pre-trained model provided by the original author (Redmon, J. et al, 2018) [24]. The authors froze all layers except the last 3 layers and trained the model with 70 epochs. For training the Faster R-CNN model, the authors used ResNet-101-FPN architecture as backbone and pre-trained weights provided by Facebook's model zoo and trained all the layers with 50 epochs. For both models, the threshold used for IoU and score are 0.5 and 0.4 respectively. As a result, the YOLOv3 model achieved average precision of 55 with 0.045s inference time while Faster R-CNN model achieved average precision of 62 with 0.15s inference time. The authors concluded that it's a speed/accuracy tradeoff on which model to be used.

In (Adhikarla, E et al, 2021) [1], the authors proposed a new webcam based real world face mask detection dataset of 1016 GB images collected across different regions of the US and made the dataset publicly accessible. The authors then re-implemented eight state-of-the-art object detection models to demonstrate their effectiveness against the real-world webcam images. From the experiments conducted, the YOLOv5 model achieved the highest mAP of 35.1% followed by Faster R-CNN of 28.1%.

In order to study the challenges for real time monitoring of people wearing masks or not, (Roy, B et al, 2020) [23] have implemented some popular object detection algorithms such as YOLOv3, SSD and Faster R-CNN and evaluated them on Moxa3K dataset. The dataset contains around 3k images with two classes: mask and no-mask. For training the YOLO v3 model, the authors used the default configurations in Darknet and the pre-trained weights provided by the original author (Redmon, J. et al, 2018) [10]. On the other hand, SSD was trained with Mobilenet v2 backbone while Faster R-CNN was trained with Inception v2 backbone using Tensorflow Object Detection API with the pre-trained weights on MS-COCO dataset. From the experiments conducted, the YOLOv3 model achieved the highest mAP of 66.8%, followed by Faster R-CNN of 60.5% and SSD of 46.5% .

**SSD-** In research by Shashi Yadav, 2020 [29], a deep learning model based on safe social distancing and face mask detection in public areas for Covid-19 safety had been proposed. In this proposed system, the system uses a transfer learning approach to performance optimization with a deep learning algorithm and computer vision to monitor people with masks or no mask in public spaces with a live camera automatically. The deep learning algorithm that is

used by the system is the Single Shot object Detection (SDD) using MobileNetV2. The reason the author use MobileNetV2 is because it provides a huge cost advantage compared to the normal 2D CNN model. The SDD also trained on a large collection of images such as ImageNet and PascalVOC. Lastly, the proposed system can track the mask from the people in public places in an automated manner efficiently and successfully in real-time.

**EfficientDet-** In a research by Abdullayev & Lim .,(2021) [2], Resnet, EfficientDet, Yolo v3 and Yolo v4 model was applied to street scene detection which means detection of cars and pedestrians. The data used in this research was manually collected from the big cities in Korea. The data was manually annotated to indicate car, person, face and license plate. EfficientDet-D1, a version of EfficientDet on Efficientnet-B1 as backbone was trained using SGD optimizer with 640x630 input image size and 300 epochs. They found that EfficientDet and ResNet was relatively faster than the Yolo model. The research found that yolo v4 had the highest accuracy for each category. However, all models tested had an accuracy of 83% and higher.

Neda Fatima et al ., (2021) [21] proposed using efficientdet and faster rcnn model to create a smart border system. They use the model to detect if a person is either a soldier or an intruder. The model was trained using 512x512 sized images and took 3 hours to train for efficientdet and 2 hours for faster rcnn using NVIDIA GeForce GT 710. They found that faster rcnn performed much better than efficientdet. One flaw this research is that if a soldier is wearing a mask it will detect as an intruder.

**ResNet-** A deep learning model research based on YOLO-v2 with ResNet-50 for medical face mask detection has been proposed by Mohamed Loey et al.,2020 [14]. In this work, the author merged the two datasets: Medical Masks Dataset (MMD) and Face Mask Dataset (FMD) by removing bad quality images and redundancy. The author introduced a detector model which includes three main components: the first component is the number of anchor boxes, the second component is the data augmentation, and the final main component is the detector. YOLOv2 with ResNet-50 is used in the detector for the feature extraction and detection during the training, validation, testing phase. Estimating the number of anchor boxes is very important to produce a high-performance detector and the estimate process will use Mean IoU as illustrated in the equation distance metric. Besides, data augmentation is also used to increase the diversity of datasets for the performance of detectors. The achieved result from the proposed model shows that the adam optimizer achieved the highest average precision percentage which is 81%. Last, the proposed work achieves a better performance than the related work.

According to Bishwas Mandal et al [19] a masked face recognition using Resnet-50 has been proposed by using the “Real-world masked face recognition datasets” (RMFRD). The author uses ResNet-50 architecture because it has the best time and memory performance when compared to VGGNet19 and DenseNet121 based on a few literature reviews. Also, the author uses supervised domain adaptation to the resulting model by training and testing the model based on two different scenarios. The author trained the model only on faces without masks and tested the performance on faces with masks. Then, the author trained the model on faces without masks and a portion of the face with masks and tested the model on the other portion of the face with masks. The precision, recall, and F1-scores are used as the evaluation metrics and the result shows that detection on the unmasked face is better than detecting on the masked face with 0.8933, 0.8970, and 0.897 in Precision, Recall, and F1-Score where detect on the masked face only have 0.4613, 0.4719, and 0.4473.

**MobileNet-** Sanjaya & Rakhmawan, 2020 [30] introduced a face mask detection method implemented using MobileNetV2. The experiment is conducted on two datasets. The first dataset, Real-World Masked Face dataset (RMFD), can be found on the github of the dataset’s author. It contains 5,000 masked faces of 525 people and 90,000 normal faces which sums up to a total of 95000 images. However the dataset is not annotated. The second dataset is a dataset composed of images taken from public places of 25 cities in Indonesia. The author did not provide the source for the second dataset. The authors claimed that their model has an accuracy of 96.85%. However, we did not see any object detector used in their framework. They are evaluating their model using F1-score.

Almghraby & Okasha Elnady, 2021 [3], proposed a face mask detection model in real-time using MobileNetV2. The experiment is conducted on two datasets. The datasets are compiled together and contain 1800 images. The compiled dataset contains 2 classes: with mask and without mask. Both of the dataset can be found on Kaggle. However, the authors did not provide a link to both datasets. The dataset is then segmented into train, test and validation sets to prevent overfitting. They are using F1 scores to evaluate their model’s performance. Their model has a 99% training accuracy and 98% validation accuracy. They did not specify on which object detection models are used together with MobileNetV2.

#### 4. APPROACH

Many people have devised ingenious methods for detecting face masks. However, previous work by the researchers mostly focused on using Convolutional Neural Network

(CNN) to classify them. Thus, we think that it is feasible to use object detection models for detecting multiple people in an image and generating a bounding box with confidence score depending on whether they are wearing a mask, not wearing a mask or wearing a mask incorrectly. As some state-of-the-art object detection models have shown very satisfactory results on detecting objects, we believe that they will have very promising results on face mask detection too. With that motivation in mind, we decided to conduct an experiments with different settings on several object detection models such as Yolov5, Faster R-CNN, Single Shot Multibox Detector (SSD) and EfficientDet to evaluate, compare and find out which object detection models give the most promising result. We are using transfer learning techniques in building our object detection models. One of the advantages of transfer learning is that it allows us to re-use the weights of filters that learnt from large datasets such as Microsoft COCO and thus reduce the training time and increase the performance of the model instead of training it from scratch. An overview of our project framework can be shown in Figure 6.

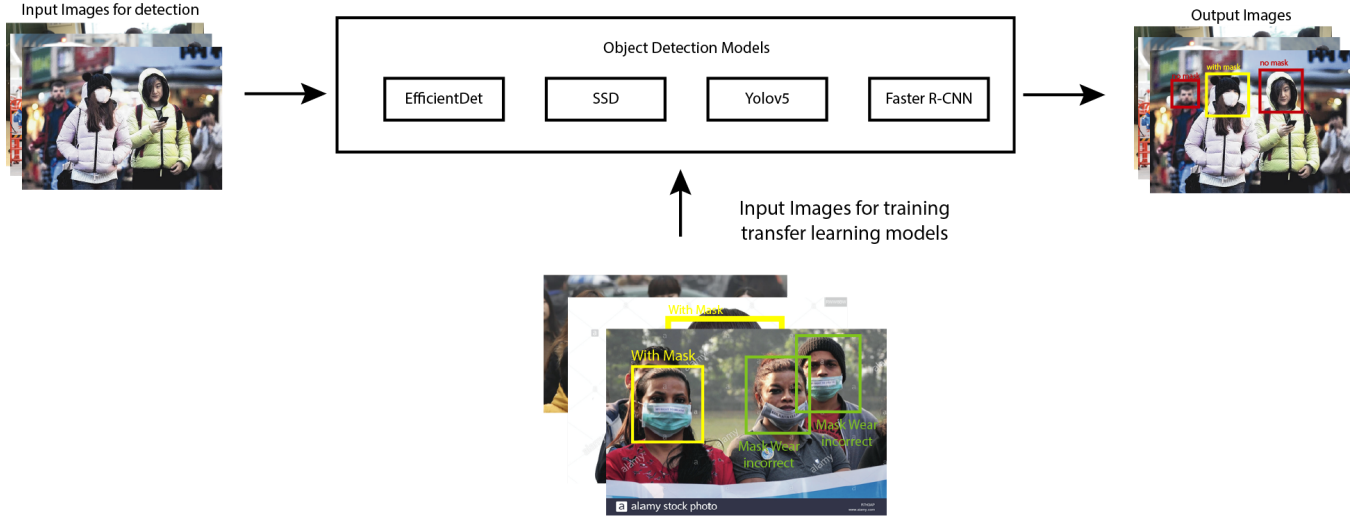
#### 5. EXPERIMENT

The dataset used to train on the models are from Andrew Maranhão which is publicly available on kaggle (insert footnote for link). The dataset only contains 853 annotated images in PASCAL VOC XML format. The dataset has three classes which are with mask, without mask and mask worn incorrectly. Although the dataset has the three classes that we need, the dataset is too small. To address this issue, we augment the data by mirroring and rotating all of the images.

By applying data augmentation techniques, we have a new dataset with 3412 images. After obtaining sufficient images, we split the dataset into train, validation and test set with the ratio of 8.5 : 1 : 0.5 (Train:2900, Validation:341, Test: 171). We then proceed to train the object detection models with the same settings for all of the models for a fair comparison and result. Models such as Faster R-CNN, SSD and EfficientDet are trained using Tensorflow Object Detection API provided by Tensorflow while the Yolov5 model is trained using the model provided by Ultralytics on their official github pages respectively. For each model, we also trained it with different CNN backbones to investigate the effectiveness of each backbone on the model’s overall performance.

We trained the Faster R-CNN model with 3 different backbones which are ResNet50, ResNet101 and ResNet152. Besides, we trained the SSD model with 2 different backbones which are MobileNet V2 and ResNet50 while for the Yolov5 model, Yolov5s and Yolov5m are trained.





Lastly, we trained the EfficientDet model with EfficientDet D0 which has the input size of 512x512 pixels. All the models are trained on Windows 10 machine and Nvidia RTX 3060 GPU with 100 epochs and default learning rate respectively.

Due to the limitations of our computing resources, we could not reproduce the same settings for all the models. Models like EfficientDet, SSD, and RCNN require a lot of computation resources and we could not train them using high settings. Hence, we can only train them with an image size of 320 x 320. Faster RCNN are even computation intensive, we can only train them using a batch size of 4 and images of size 320 x 320. Meanwhile, the training image of EfficientDet must be divisible by 128 so we could not use 320x320 as it is not divisible by 128. We will use 512x512 instead. With these limitations, the lowest setting for this project is batch size of 4 and image size of 320 x 320. The highest setting for this project is batch size of 32 and image size of 640x640 which can only be trained on Yolov5 models.

In assessing our model's performance, we decided to use mean Average Precision (mAP) for evaluation purposes. The mAP score is calculated by taking the mean of Average Precision (AP) of each class. Note that, we will first compare all of the models using the lowest setting to find the best performing model with limitations so that it is fair for every model. We will then find out which model has the highest mAP ignoring the training settings. An overview of models settings is shown in table 1.

## 6. RESULT AND ANALYSIS

As mentioned before, for a fair comparison, we will only compare models with the same image size and batch size.

Table 1 : Overview of object detection models settings. The symbol “✓” means the model is trainable while “-” means the model is not trainable due to insufficient memory issues.

Models	Image size			Batch size			
	320x320	640x640	512x512	4	8	16	32
SSD	✓	-	-	✓	✓	✓	✓
Yolov5	✓	✓	-	✓	✓	✓	✓
Efficient Det	-	-	✓	✓	✓	✓	-
Faster RCNN	✓	-	-	✓	-	-	-

However for EfficientDet D0, the image dimension must be divisible by 128, hence we could not set it to 320x320. We will still compare EfficientDet D0 with models with image size of 320x320 as it is already the lowest setting for EfficientDet. For a fair comparison, yolov5 models trained with image size of 640x640 would not be compared as the other models could not be trained on the same image size due to hardware limitation.

Table 2 shows the performance of the object detection models on the validation set. As depicted in table 2, Yolov5s has the highest overall mAP@0.5 and overall mAP@.5:.95 in the validation set which are 0.84 and 0.57, respectively. It also has a 0.671 mAP@.5:.95 for mask and 0.503 AP@.5:.95 for without mask which is the highest among the models trained with the same setting, image size of 320x320 and batch size of 4. On the other hand, for models trained

with batch size of 8, Yolov5s also has the highest overall  $mAP@0.5$ , overall  $mAP@.5:.95$ ,  $mAP@.5:.95$  for mask, without mask and mask worn incorrectly which are 0.869, 0.593, 0.679, 0.518 and 0.581, respectively. Yolov5s also has the highest  $mAP$  for all of the categories among models that are trained with batch size of 16 and 32.

Table 3 shows the performance of the object detection models on the test set. As shown in table 3, for models trained with batch size of 4, Yolov5s has the highest overall  $mAP@0.5$  and over  $mAP@.5:.95$  on the test set which are 0.74 and 0.471, respectively. It has also the highest  $mAP@.5:.05$  for mask and  $mAP@.5:.05$  on the test set for without mask which are 0.625 and 0.477, respectively. However, it has slightly lower  $mAP@.5:.05$  for incorrect masks compared with Faster RCNN with Resnet101 as backbone. While for models trained with batch size of 8, Yolov5s also has the highest overall  $mAP@0.5$  and overall  $mAP@.5:.95$  on test sets which are 0.869 and 0.593 respectively. Lastly, Yolov5s also has the highest overall  $mAP@0.5$  and overall  $mAP@.5:.95$  which is 0.756 and 0.48, respectively, among the models trained with images size of 320x320 and batch size of 32.

## 7. DISCUSSION AND FUTURE WORK

Yolov5 perform very well for both the validation set and test set. They are very efficient too as they do not need a lot of computing resources to train. Faster RCNN requires a lot of computing resources although we train them using the lowest setting: 320x320 image size and batch size of 4. Hence, their architecture is not that efficient compared to other object detection models such as Yolov5 and SSD. From table 2 and 3, we can also see that Yolov5s is always superior than Yolov5m. This is probably due to the number of epochs used in training. Since the size of Yolov5m is three times larger than Yolov5s, it requires more epoch to train. We think that Faster RCNN with Resnet 101 has the potential to be a model specifically trained to detect people who wear masks incorrectly as with such a low training setting, they have relatively high  $mAP@.5:.95$  for masks worn incorrectly in both testing set and validation set compared to Yolov5.

Since Yolov5 is so versatile and efficient, we have decided to push this project further by training Yolov5 with higher settings. The results of validation set and testing set are shown in table 4 and table 5, respectively. As shown in table 4, Yolov5s trained with image size of 640 x 640 and batch size of 32, achieved an overall  $mAP@0.5$  of 0.918 and overall  $mAP@.5:.95$  of 0.701 which is the highest among all

the models tested on the validation set. On the other hand, Yolov5s trained with image size of 640 x 640 and batch size of 32 also showed amazing results in the test set. It can achieve an overall  $mAP@0.5$  of 0.868 and an overall  $mAP@.5:.95$  of 0.597 which is the best result among all of the models tested on the test set. Figure 7 shows the model detecting the presence of a facial mask on a person.



Figure 7: Example images of the best model detecting the presence of a facial mask on a person.

However, the best model is not perfect. We have tested the model several times and we found out that our model has a limitation. The model will identify the person wearing a mask if there is an object blocking the nose and mouth of that person. Figure 8 shows an example image of our model detecting someone with a mask although the person is just covering his nose with an object. This is due to the fact that our dataset used to train the model is not vast enough to cover scenarios like that.

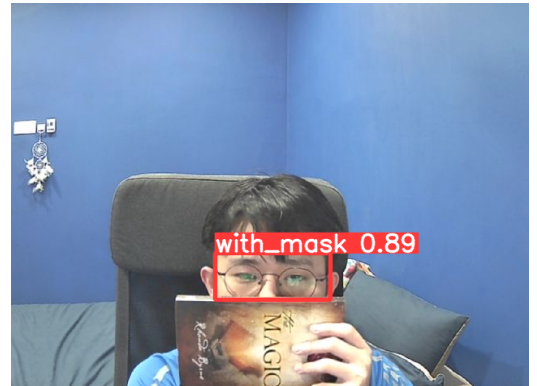


Figure 8: Our model identifies someone blocking their nose and mouth as wearing a mask.

Table 2 : Performance of object detection models in validation set

Model	Image Size	Batch size	Evaluation Metrics (with val set) at 100 epochs				
			mAP@0.5	mAP@.5:.95	mAP@.5:.95 mask	mAP@.5:.95 without mask	mAP@.5:.95 incorrect mask
Faster RCNN Resnet50	320x320	4	0.74	0.533	0.614	0.418	0.566
Faster RCNN Resnet101			0.754	0.561	0.636	0.448	0.599
Faster RCNN Resnet152			0.731	0.54	0.628	0.4522	0.538
Yolov5s			0.84	0.57	0.671	0.503	0.537
Yolov5m			0.814	0.527	0.64	0.438	0.504
SSD MobileNet			0.712	0.461	0.594	0.397	0.391
SSD ResNet50			0.533	0.327	0.488	0.258	0.235
EfficientDet D0	512x512	8	0.656	0.404	0.584	0.3641	0.2625
Yolov5s	320x320		0.869	0.593	0.679	0.518	0.581
Yolov5m			0.807	0.538	0.649	0.445	0.52
SSD MobileNet			0.756	0.503	0.6175	0.4198	0.4725
SSD ResNet50			0.593	0.377	0.526	0.292	0.312
EfficientDet D0	512x512	0.745	0.477	0.569	0.392	0.47	
Yolov5s	320x320	16	0.875	0.6	0.683	0.523	0.593
Yolov5m			0.818	0.552	0.655	0.451	0.549
SSD MobileNet			0.664	0.422	0.55	0.339	0.375
SSD ResNet50			0.699	0.462	0.597	0.382	0.406
EfficientDet D0	512x512	0.792	0.507	0.608	0.438	0.476	
Yolov5s	320x320	32	0.863	0.592	0.683	0.523	0.57
Yolov5m			0.815	0.547	0.654	0.462	0.525
SSD MobileNet			0.674	0.44	0.577	0.352	0.4018
SSD ResNet50			0.687	0.444	0.573	0.391	0.367



Table 3 : Performance of object detection models in test set

Model	Image Size	Batch size	Evaluation Metrics (with test set) at 100 epochs				
			mAP@0.5	mAP@.5:.95	mAP@.5:.95 mask	mAP@.5:.95 without mask	mAP@.5:.95 incorrect mask
Faster RCNN Resnet50	320x320	4	0.612	0.395	0.53	0.4	0.252
Faster RCNN Resnet101			0.642	0.438	0.559	0.429	0.326
Faster RCNN Resnet152			0.599	0.412	0.55	0.4166	0.268
Yolov5s			0.554	0.326	0.505	0.313	0.159
Yolov5m			0.74	0.471	0.626	0.477	0.311
SSD MobileNet			0.669	0.403	0.595	0.394	0.221
SSD ResNet50			0.578	0.357	0.514	0.344	0.211
EfficientDet D0	512x512		0.439	0.248	0.428	0.218	0.09
Yolov5s	320x320	8	0.731	0.463	0.637	0.476	0.277
Yolov5m			0.696	0.421	0.597	0.403	0.264
SSD MobileNet			0.584	0.368	0.5278	0.3582	0.2173
SSD ResNet50			0.47	0.27	0.4498	0.2619	0.0096
EfficientDet D0	512x512		0.591	0.365	0.492	0.368	0.234
Yolov5s	320x320	16	0.772	0.49	0.645	0.478	0.343
Yolov5m			0.7	0.422	0.599	0.424	0.243
SSD MobileNet			0.54	0.313	0.459	0.296	0.183
SSD ResNet50			0.542	0.342	0.516	0.335	0.174
EfficientDet D0	512x512		0.631	0.394	0.525	0.403	0.253
Yolov5s	320x320	32	0.756	0.48	0.642	0.488	0.308
Yolov5m			0.681	0.418	0.607	0.411	0.237
SSD MobileNet			0.56	0.337	0.492	0.331	0.185
SSD ResNet50			0.552	0.341	0.504	0.332	0.186

Table 4: Results on validation set of Yolov5m and Yolov5s trained on higher setting

Model	Image Size	Batch size	Evaluation Metrics (with val set) at 100 epochs				
			mAP@0.5	mAP@.5:.95	mAP@.5:.95 mask	mAP@.5:.95 without mask	mAP@.5:.95 incorrect mask
Yolov5s	640x640	4	0.909	0.679	0.744	0.63	0.662
Yolov5m	640x640	4	0.883	0.631	0.717	0.606	0.569
Yolov5s	640x640	8	0.921	0.69	0.747	0.626	0.696
Yolov5m			0.875	0.632	0.714	0.574	0.608
Yolov5s	640x640	16	0.922	0.689	0.75	0.63	0.686
Yolov5m			0.829	0.542	0.69	0.575	0.362
Yolov5s	640x640	32	<b>0.918</b>	<b>0.701</b>	<b>0.755</b>	<b>0.641</b>	<b>0.707</b>
Yolov5m			0.89	0.642	0.719	0.571	0.636

Table 5: Results on test set of Yolov5m and Yolov5s trained on higher setting

Model	Image Size	Batch size	Evaluation Metrics (with test set) at 100 epochs				
			mAP@0.5	mAP@.5:.95	mAP@.5:.95 mask	mAP@.5:.95 without mask	mAP@.5:.95 incorrect mask
Yolov5s	640x640	4	0.831	0.58	0.716	0.391	<b>0.632</b>
Yolov5m	640x640	4	0.836	0.552	0.688	0.409	0.558
Yolov5s	640x640	8	0.861	0.596	0.72	<b>0.637</b>	0.432
Yolov5m			0.809	0.54	0.683	0.558	0.38
Yolov5s	640x640	16	0.829	0.542	0.69	0.575	0.362
Yolov5m			0.848	0.59	<b>0.726</b>	0.627	0.417
Yolov5s	640x640	32	<b>0.868</b>	<b>0.597</b>	0.724	0.633	0.435
Yolov5m			0.827	0.546	0.693	0.578	0.367

For our future work, we plan to train those models that require more computation power using a stronger computer so that each of the models can compete under high settings. We would also increase the number of epochs used to train the models to allow the model such as Yolov5m to converge more. Lastly, we would like to collect a larger dataset which has a lot more scenarios to overcome the limitations that we stated above.

## 8. CONCLUSION

This paper built several state-of-the-art object detection models to recognize whether a person is wearing or not wearing or wearing a mask incorrectly. Due to limited computation resources, large networks like Faster R-CNN could not be trained on batch size larger than 4 and image size larger than 320 x 320. To ensure that we evaluate all of the state-of-the-art object detection models fairly, we trained them using batch size 4 and image size of 320 x 320. The result shows that Yolov5s trained on low setting, outperformed the other models that are trained on low setting too. To find out the best model, we continue to train all of the models except Faster R-CNN using medium settings. The result shows that Yolov5s still outperformed the other models. We tried to push the models further by training them on larger images (640 x 640). However, only Yolov5 can be trained due to limited computing resources. Hence, we proceed to train Yolov5 models. In the end, we found out that Yolov5s trained on image size of 640x640 and batch size of 32 had the best result with an overall mAP@0.5 of 86.8% and mAP@.5:.95 of 59.7%.

## 8. REFERENCES

- [1] Adhikarla, E., & Davison, B. D. (2021). Face mask detection on real-world webcam images. *GoodIT 2021 - Proceedings of the 2021 Conference on Information Technology for Social Good*, 139–144. <https://doi.org/10.1145/3462203.3475903>
- [2] Abdullayev, A., & Lim, C. G. The Performance Evaluations of Street Scene Detection by using Intelligent Object Detectors.
- [3] Almghraby, M., & Okasha Elnady, A. (2021). Face Mask Detection in Real-Time using MobileNetv2. *International Journal of Engineering and Advanced Technology (IJEAT)*, 2249–8958. <https://doi.org/10.35940/ijeat.F3050.0810621>
- [4] Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2013). Rich feature hierarchies for accurate object detection and semantic segmentation. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 580–587. <https://doi.org/10.1109/CVPR.2014.81>
- [17] Girshick, R. (2015). Fast R-CNN. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. <https://arxiv.org/abs/1504.08083v2>
- [6] G. Yang et al., "Face Mask Recognition System with YOLOV5 Based on Image Recognition," 2020 IEEE 6th International Conference on Computer and Communications (ICCC), 2020, pp. 1398-1404, doi: 10.1109/ICCC51575.2020.9345042.
- [7] Hussain, S., Yu, Y., Ayoub, M., Khan, A., Rehman, R., Wahid, J. A., & Hou, W. (2021). IoT and Deep Learning Based Approach for Rapid Screening and Face Mask Detection for Infection Spread Control of COVID-19. *Applied Sciences*, 11(8), 3495. <https://doi.org/10.3390/app11083495>
- [8] Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. (2017). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. <https://arxiv.org/abs/1704.04861v1>
- [9] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition (pp. 770–778). <http://image-net.org/challenges/LSVRC/2015/>
- [10] Ieamsaard, J., Charoensook, S. N., & Yammen, S. (2021, March). Deep Learning-based Face Mask Detection Using YoloV5. In 2021 9th International Electrical Engineering Congress (iEECON) (pp. 428-431). IEEE.
- [11] Jignesh Chowdary, G., Pun, N. S., Sonbhadra, S. K., & Agarwal, S. (2020). Face Mask Detection Using Transfer Learning of InceptionV3. *Big Data Analytics*, 81–90. [https://doi.org/10.1007/978-3-030-66665-1\\_6](https://doi.org/10.1007/978-3-030-66665-1_6)
- [12] Jiang, M., Fan, X., & Yan, H. (2020). RetinaMask: A Face Mask detector. <https://arxiv.org/abs/2005.03950v2>
- [13] Kumar Addagarla, S., Kalyan Chakravarthi, G., & Anitha, P. (2020). Real Time Multi-Scale Facial Mask Detection and Classification Using Deep Transfer Learning Techniques. *Article in International Journal of Advanced Trends in Computer Science and*

Engineering, 9(4), 4402–4408.  
<https://doi.org/10.30534/ijatcse/2020/33942020>

[14] Loey, M., Manogaran, G., Taha, M. H. N., & Khalifa, N. E. M. (2021). 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*, 65. <https://doi.org/10.1016/J.SCS.2020.102600>

[15] Li, Y. (2021, September). Facemask detection using inception V3 model and effect on accuracy of data preprocessing methods. In *Journal of Physics: Conference Series* (Vol. 2010, No. 1, p. 012052). IOP Publishing.

[16] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C. (2015). SSD: Single Shot MultiBox Detector. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9905 LNCS, 21–37. [https://doi.org/10.1007/978-3-319-46448-0\\_2](https://doi.org/10.1007/978-3-319-46448-0_2)

[17] Mbunge, E., Simelane, S., Fashoto, S. G., Akinuwa, B., & Metfula, A. S. (2021). Application of deep learning and machine learning models to detect COVID-19 face masks - A review. *Sustainable Operations and Computers*, 2, 235–245. <https://doi.org/10.1016/J.SUSOC.2021.08.001>

[18] Moher, D., Liberati, A., Tetzlaff, J., Altman, D. G., Altman, D., Antes, G., Atkins, D., Barbour, V., Barrowman, N., Berlin, J. A., Clark, J., Clarke, M., Cook, D., D'Amico, R., Deeks,

[19] Mandal, B., Okeukwu, A., & Theis, Y. (2021). Masked Face Recognition using ResNet-50. <https://arxiv.org/abs/2104.08997v1>

[20] Nelson, J. (2021, September 21). YOLOv5 is Here: State-of-the-Art Object Detection at 140 FPS. Roboflow [Blog](https://blog.roboflow.com/yolov5-is-here/). <https://blog.roboflow.com/yolov5-is-here/>

[21] N. Fatima, S. A. Siddiqui and A. Ahmad, "IoT based Border Security System using Machine Learning," 2021 International Conference on Communication, Control and Information Sciences (ICCISc), 2021, pp. 1-6, doi: 10.1109/ICCISc52257.2021.9484934.

[22] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards Real-Time Object Detection

with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6), 1137–1149. <https://arxiv.org/abs/1506.01497v3>

[23] Roy, B., Nandy, S., Debojit Ghosh, , Debarghya Dutta, , Pritam Biswas, , & Das, T. (2020). MOXA: A Deep Learning Based Unmanned Approach For Real-Time Monitoring of People Wearing Medical Masks. *Transactions of the Indian National Academy of Engineering* 2020 5:3, 5(3), 509–518. <https://doi.org/10.1007/S41403-020-00157-Z>

[24] Redmon, J., & Farhadi, A. (2018). YOLOv3: An Incremental Improvement. <https://arxiv.org/abs/1804.02767v1>

[25] Sharma, V. (2020). Face Mask Detection using YOLOv5 for COVID-19 (Doctoral dissertation, California State University San Marcos).

[26] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2818–2826).

[27] Solawetz, J. (2021, September 21). YOLOv5 New Version - Improvements And Evaluation. Roboflow Blog. <https://blog.roboflow.com/yolov5-improvements-and-evaluation/>

[28] Singh, S., Ahuja, U., Kumar, M., Kumar, K., & Sachdeva, M. (2021). Face mask detection using YOLOv3 and faster R-CNN models: COVID-19 environment. *Multimedia Tools and Applications*, 80(13), 19753–19768. <https://doi.org/10.1007/S11042-021-10711-8/FIGURES/8>

[29] Shashi Yadav. (2020). Deep Learning based Safe Social Distancing and Face Mask Detection in Public Areas for COVID-19 Safety Guidelines Adherence. (2020). <https://doi.org/10.22214/ijraset.2020.30560>

[30] Sanjaya, S. A., & Rakhmawan, S. A. (2020). Face Mask Detection Using MobileNetV2 in the Era of COVID-19 Pandemic. 2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy, ICDABI 2020. <https://doi.org/10.1109/ICDABI51230.2020.9325631>

[31] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). MobileNetV2: Inverted Residuals and Linear Bottlenecks. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 4510–4520. <https://doi.org/10.1109/CVPR.2018.00474>

[32] Tan, M., Pang, R., & Le, Q. V. (2020). Efficientdet: Scalable and efficient object detection. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 10781-10790).