Mask Off

Word Count: 1965

Won Jin Kim, 1003611424;
Batuhan Raif Karagöz, 1005117339;
Kieran Cyphus, 1003933648;

Introduction

Although there are now several promising covid19 vaccine candidates going through the process of approval, proper mask usage will still be a key component in a strong response to the pandemic, which is currently well into its second wave globally. Despite this, there is still a significant number of people refusing to use masks, or are wearing it incorrectly which according to UofT epidemiologist Colin Furness, is just as bad as not wearing one at all [6]. The provincial and municipal governments have implemented mask laws to ensure compliance [5], but it is still incredibly difficult to get data on mask usage. The goal of "Mask Off" is to classify different mask usages given a live video stream into either: mask, no mask or uncovered nose. In recent years, image classification has been most successfully solved using neural networks, (as opposed to fuzzy logic or support vector machines) and as such are appropriate for this image classification problem. [8] The applications of this are widespread and beneficial to the community. For example, a camera can be set up inside a store with our classifier and notify if an individual is not wearing a mask properly, and appropriate action can be taken.

Illustration / Figure

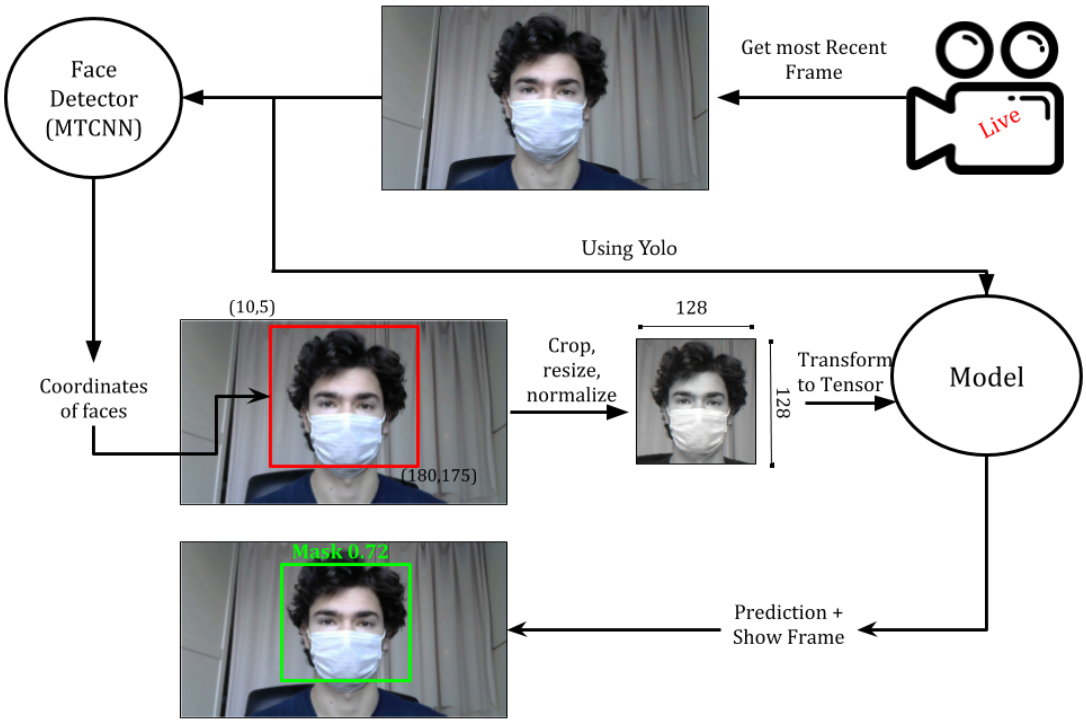


Figure 1: Project Illustration

Background and Related Work

As expected, a lot of work has been done in this area since March, with some notable examples of models being one trained by Nvidia to classify mask / no mask which was built upon a resnet backbone and achieved ~80% accuracy [7]. Additionally, there are quite a few datasets that have been made publicly available to train models on, such as "Mask Wearing Dataset" [3] which is a collection of public images containing potentially masked users and "MaskedFace-Net" [2], which augmented a facial dataset by photoshopping surgical masks onto the faces.

Our models rely on a face detector during live video to feed it the cropped photo. The face detector we used was the pretrained MTCNN [1] (Multi-Task cascaded Convolutional Neural Network), which upon proper implementation, returns the bounding box of all the faces in an image.

Data and Data Processing

In total, we needed to gather images of three class types: masked, not masked, and nose uncovered. Due to the existing work being largely focused on mask / no mask, there is already an imbalance of data with the nose uncovered classification being underrepresented. Additionally, since the YOLO architecture requires a different input structure than the CNN, we will have to modify the datasets so that they can be trained equivalently on both models. To do this, we will be pulling results from three data sources: personally collected, "MaskedFace-Net" [2], and "Mask Wearing Dataset" [3].

Personally collected

Personal data was collected with use of videos (live or prerecorded) and the MTCNN Face Detector. Participants were to record themselves wearing 3 different masks in each of the 3 mask usage classes, with (ideally) 9 face orientations per mask. During the video, an assistant shined a moving light on the face which moved throughout the video to get a wide range of lighting conditions, as can be seen in the example below. As the video was running, the MTCNN would find the locations of faces, after which we would crop and save these images once every few frames for the CNN data. To convert this into the YOLO format, we first input the coordinates for the box in a PASVAL VOC XML template file, as it's easier to manipulate, and then convert it to the YOLO format using the online tool: Roboflow.
With this method, we collected 1391 images, which made up 17% of our final dataset for the baseline and resnet.

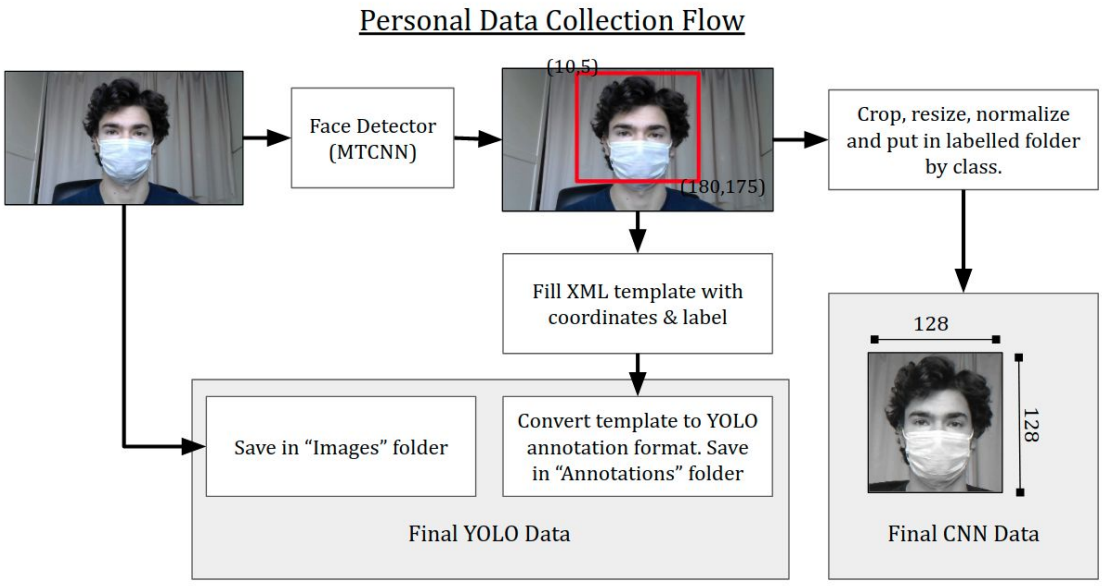


Figure 2: Personal Data Collection Flow

Figure 3: Example Data Collected for one person in one position with one mask

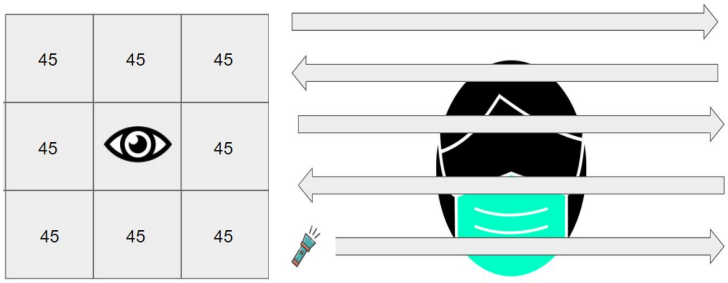


Figure 4: Illustration of Angles and Lighting Conditions for Personal Data Collection

MaskedFace-Net

MaskedFace-Net [2] is a publicly available dataset of over 130 000 images of correctly and incorrectly masked faces. The faces are originally from the Flickr-Faces-HQ Dataset [3], but masks have been photoshopped onto them in both correct and incorrect positions, as can be seen in figure 5 below. The mask locations in the photoshop are good overall, although there are some clear shortcomings. Firstly, all the masks are of the same, light blue colour with the light source coming from the same direction. Secondly, the 'uncovered nose' case adds a very noticeable and unrealistic wrinkle in the middle of the mask . We predicted that this dataset would not be enough to train a mask detector that works for real masks.

Figure 5: Example of Photoshopped Images (notice wrinkle in Uncovered Nose)

Mask Wearing Dataset

This dataset was originally in the YOLOv5 format (bounding boxes with classification labels for each) and consisted of a total of 149 images of public places with multiple people in, with the same three classification of masked, unmasked, and nose uncovered. In total, there were 3500 covered examples, 874 uncovered, and 250 nose uncovered examples. This dataset was partially balanced by augmenting the examples that contained nose uncovered and them combining with other datasets to even out the numbers. A more detailed explanation is given in figure 6.

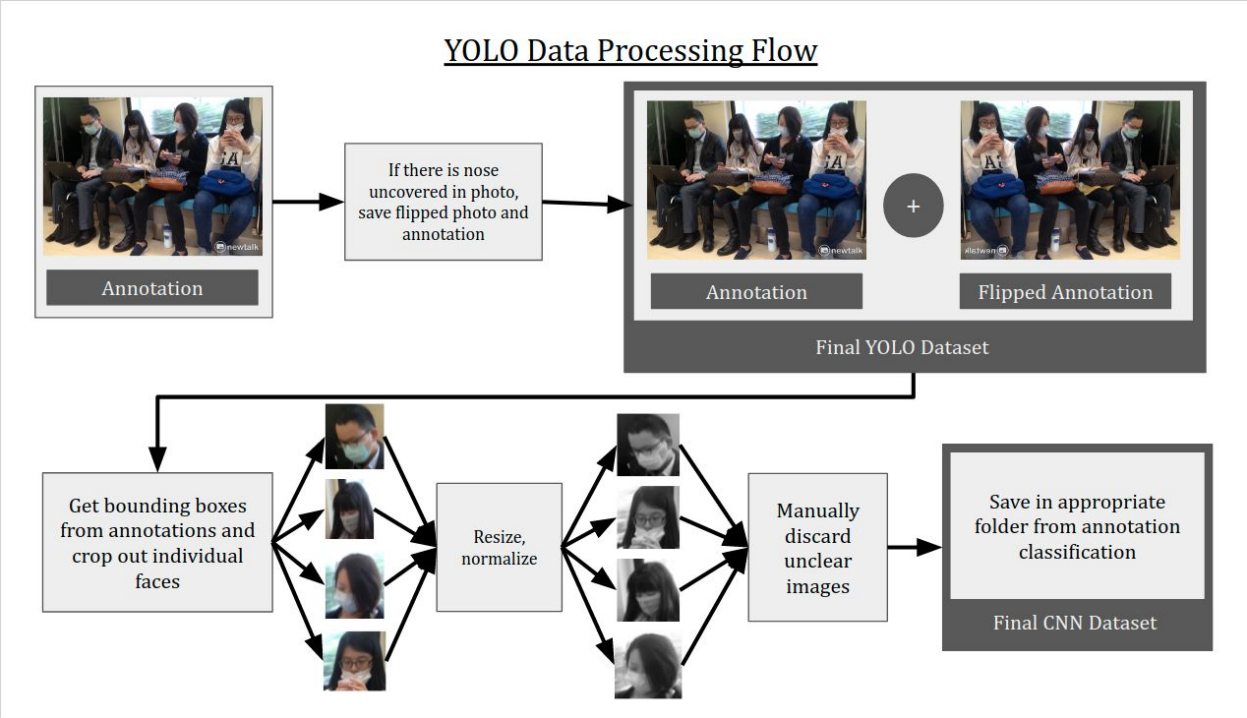


Figure 6: YOLOv5 Data Processing Flow

An image was determined to be "unclear" if it couldn't be correctly labelled by a human in under a second. Examples of some bad photos are below.

Figure 7: Unclear Images from Mask Wearing Dataset

Since these were each a part of a larger photo, it wasn't possible to remove them from the YOLOv5 dataset (given the time) and so we can expect to see some caps on accuracy there.

Final (non-YOLOv5) Dataset:

Due to the small number and scarce composition of the unphotoshopped images we had, we felt obliged to use the photoshopped dataset in addition. In the end, we achieved roughly a 50/50 balance between photoshopped and unphotoshopped images, with a final

count of 8167 128x128 pixel images. Graphs highlighting the composition of the final dataset can be found below. Our final split was 70%/10%/20% for training, validation and testing respectively.

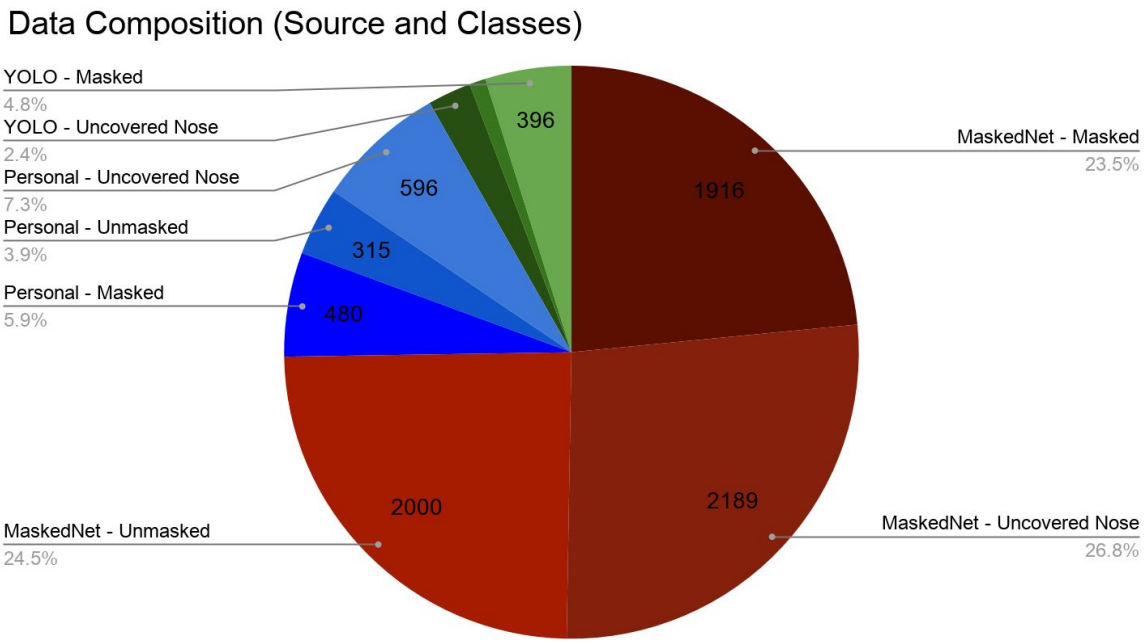


Figure 8: Data Composition Divided By Source and Classes

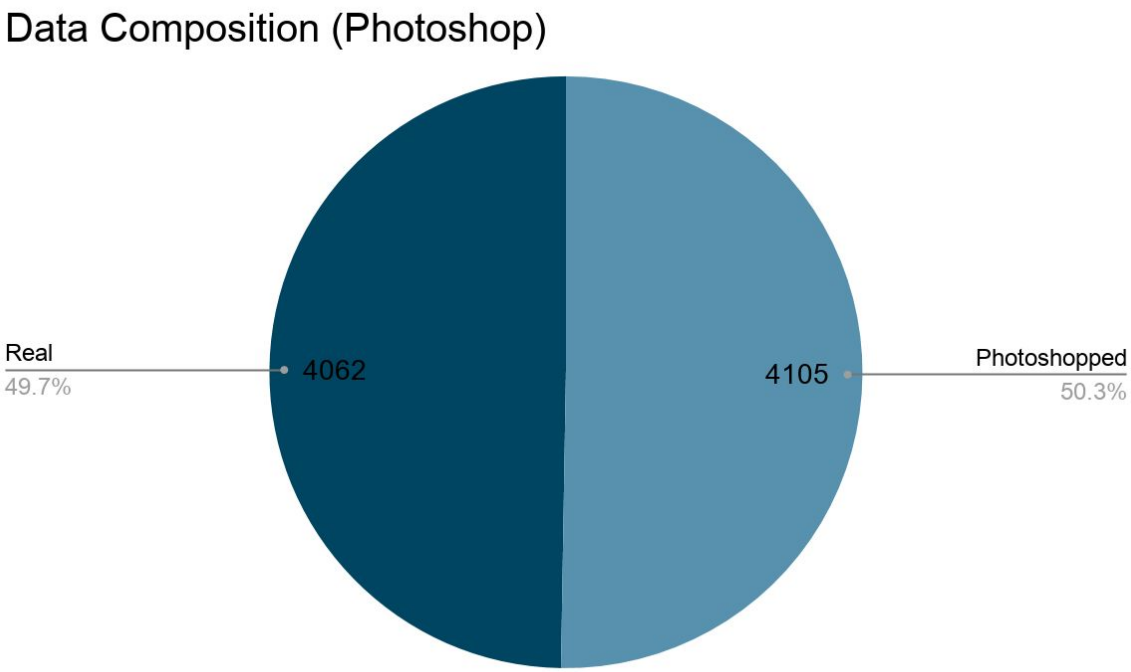


Figure 9: Data Composition Divided By Real vs Photoshopped

Baseline Model

Our baseline model relied on the MTCNN facedetector in order to function. It was a CNN with 3 convolutional layers followed by 2 fully connected layers also utilizing batch normalization. As shown in the flow diagram, MTCNN returns coordinates of the faces in the video stream, which are used to crop out the face. This image is then resized (to 128x128), and normalized based on the means and standard deviations of the entire dataset. Finally it is input to our own CNN and a final prediction is made.

Our Resnet model also relies on the MTCNN, and was trained on the same dataset, so we can easily compare these two models with any performance metrics. However, since the YOLOv5 model also does the face detection itself and uses a different dataset to train, the comparison between the baseline will be qualitative.

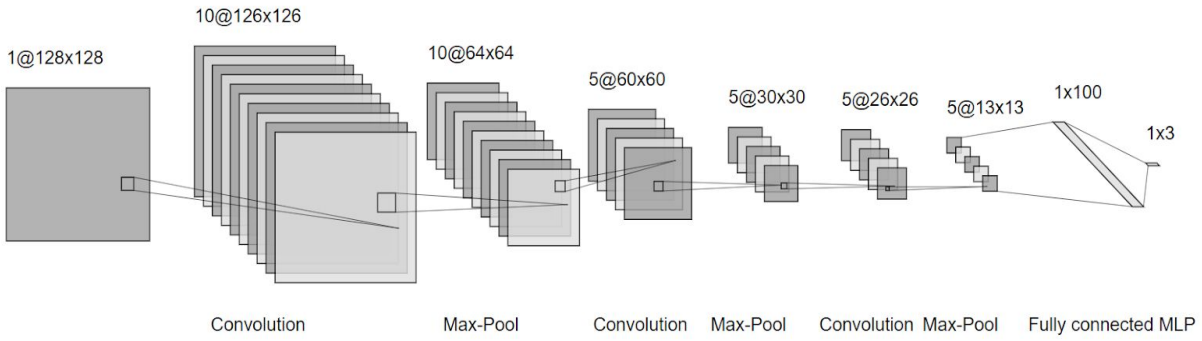


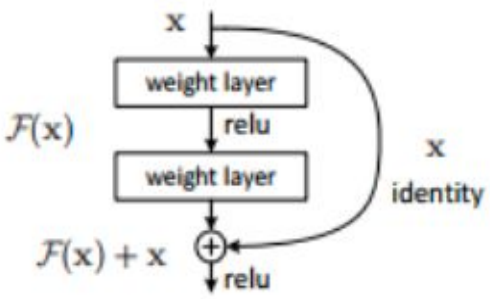
Figure 12: Summary of Baseline Model

Architecture

We decided to experiment with two different model types: a pretrained Resnet18 and YOLOv5. Resnet18 performs similarly to the baseline in that it needs a preprocessed image of just one person's face, whereas YOLOv5 takes in the whole image and outputs the classifications and their location in the photo.

Resnet18

Made of "residual blocks" with skip connections that are supposed to help deal with the exploding / disappearing gradient problem. In total, there are eight residual blocks, as well as one primary convolutional layer to resize the initial image, and for this model we removed the usual 1000 node MLP at the end and replaced it with 3 nodes, one for each of our classes. The weights of they residual layers are as follows:



layer name	output size	18-layer
conv1	112×112	
conv2_x	56×56	[3×3, 64; 3×3, 64] ×2
conv3_x	28×28	[3×3, 128; 3×3, 128] ×2
conv4_x	14×14	[3×3, 256; 3×3, 256] ×2
conv5_x	7×7	[3×3, 512; 3×3, 512] ×2
	1×1	

Figure 10: Residual block (left), Summary of Resnet 18 Architecture (right)

YOLOv5

Similar to the pretraining process done with resnet18, the output of CNN results in features maps that are connected to a 3 node MLP at the end to produce classification results with the location of the corresponding bounding box. It additionally tells us the confidence of the classification. A diagram is included below, but it is out of the scope of the report to explain it fully.

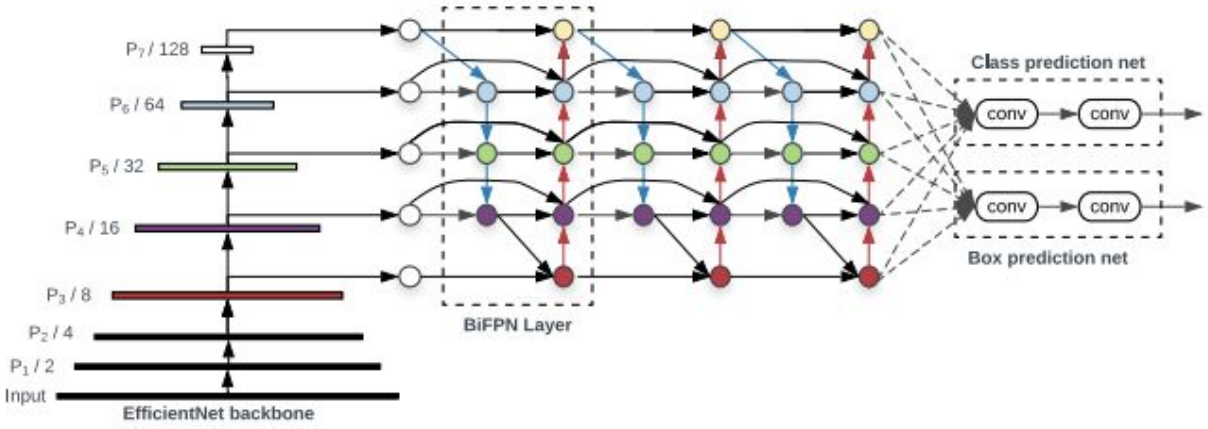


Figure 11: Summary of YOLOv5 Architecture

Quantitative Results

Again, here it is not possible to directly compare the results of the baseline and Resnet with those of YOLOv5 due to their differences in input and output. That being said, we can directly see the final training, validation and test results of the first two.

	Training Accuracy	Validation Accuracy	Test Accuracy
Baseline (CNN)	96.3%	95.6%	96.7%
Resnet	98.6%	97.6%	97.8%

Table 1: Accuracies of Baseline and Resnet Models

As we see, there is very little overfitting happening on both the models, with Resnet slightly outperforming the baseline. Additionally, looking at the confusion matrix below for the baseline, we can see that it struggled the most when differentiating between mask and uncovered nose, which is reasonable given that optically, they are the most similar.

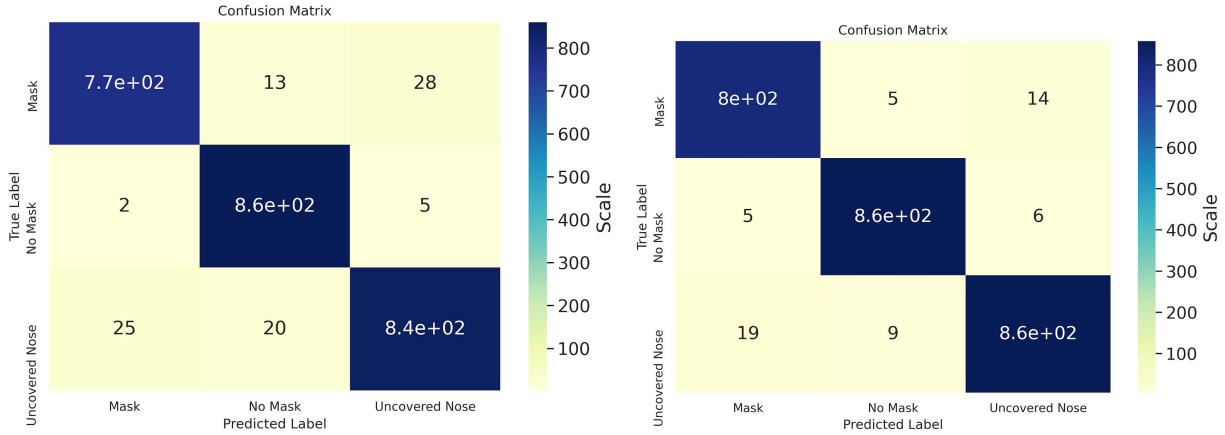


Figure 13: Confusion Matrices of The Baseline (left) and Resnet (right) Models for Same Data

The YOLOv5 model achieved a final test accuracy of 0.841. This seems a lot lower than the previous two models, however it should be kept in mind that YOLOv5 trained on a different dataset, and also finds the location of the faces alongside classification. As such, a one on one quantitative comparison between the two sets of models cannot be made.

Qualitative Results

We noticed that there was a significant difference in the confidences of the 3 models while testing live, although all 3 worked decently. Our baseline (in our testing) never had a 'confidence' of more than 0.6, while the resnet was able to achieve stellar confidences of >0.95.

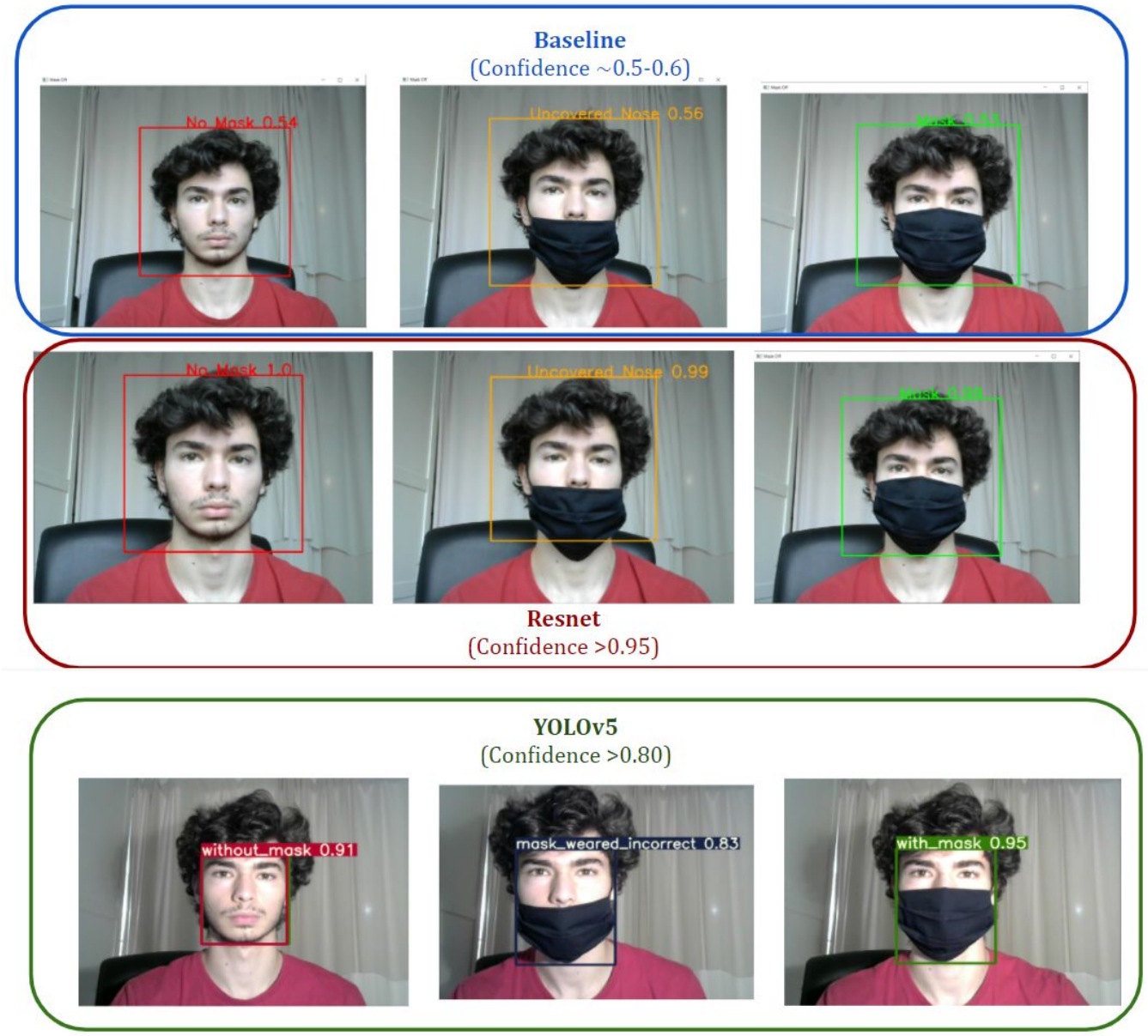


Figure 14: Example of program with different models showing confidences

YOLOv5, on the other hand, has good 0.8 confidence in its decision, but since in this case the participant wasn't present in the training set for YOLOv5, we naturally expect it to be a bit lower, thus not making the comparison entirely fair.

Discussions and Learnings

Photoshopped Data

In Figure 15 below, we can see the performance of models trained and validated under different data sets. The model which was trained only on the photoshopped dataset but validated on real data is clearly not suitable for use. For all the real images, the model predicts the individual is never wearing a mask, implying that the features the model learned to look for aren't reflective of real use cases.

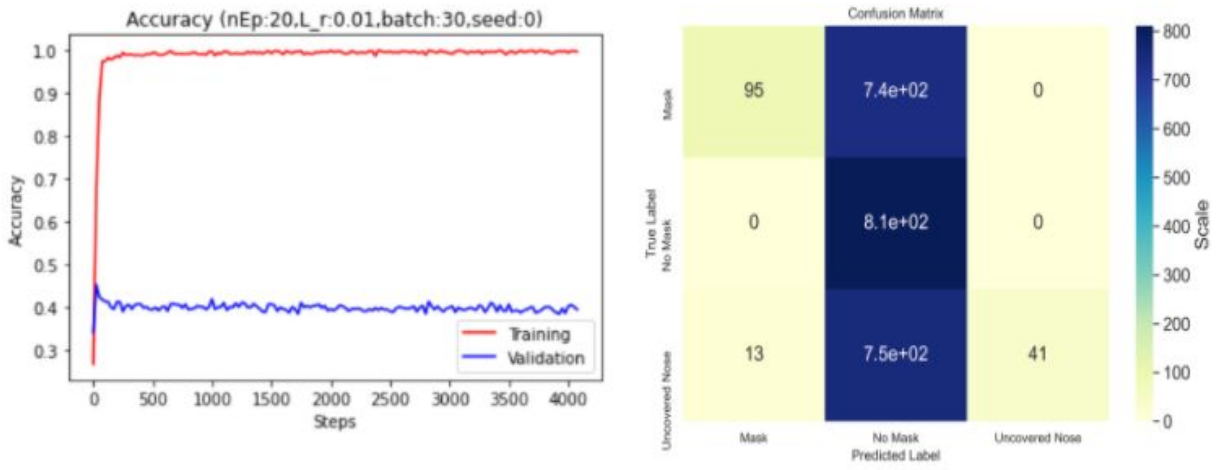


Figure 15: Training, validation accuracy and confusion matrix for trained on photoshop, validated real Baseline Model

The reverse of this, a model trained on real data and validated on photoshopped data (below) shows that the model does pretty well and so while there are similar features between the two datasets, just training on the photoshopped data alone won't be enough.

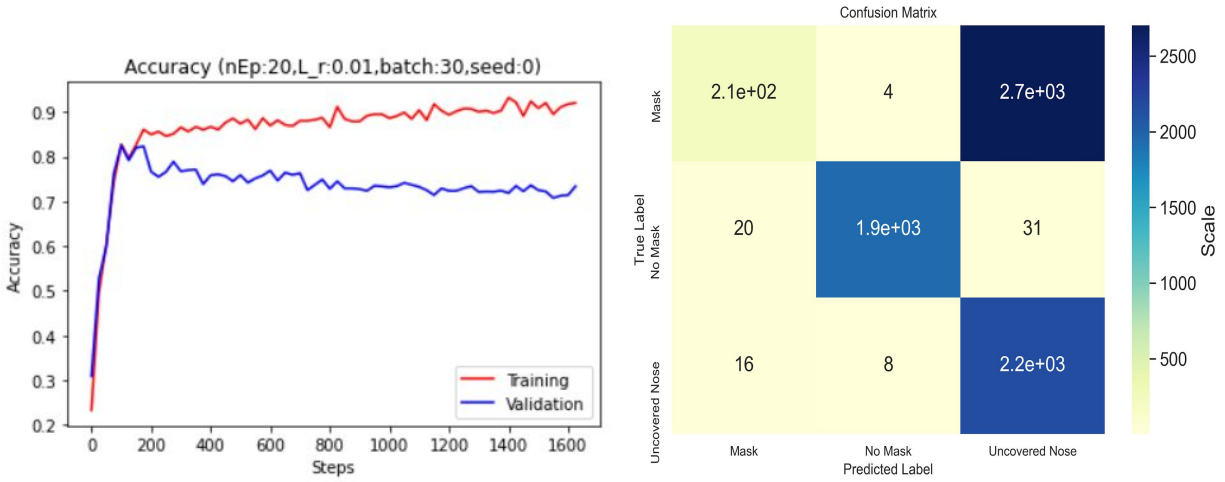


Figure 16: Training, validation accuracy and confusion matrix for trained on real, validated on photoshop

While the above data are interesting for the sake of trying to determine the characteristics of the data, we shouldn't actually include any of the photoshopped images in our validation or test test. Including photoshopped images would destroy the purpose of seeing how well the model performs on real data, and since for this project we didn't realize this until after we calculated our results, in future iterations this would be essential to do and compare findings to ensure they are similar.

Ethical Framework

A table of stakeholders and effects are listed below.

Stakeholder	**Axis**	**Increase/ Decrease**	**Reasoning**
Pedestrians	Autonomy	Decrease	Decreased as they don't have a choice of whether to participate or not.
	Non-Maleficence	Increase	Encouraging mask use should reduce transmission rates which will increase Non-Maleficence.
Medical Workers	Beneficence	Increase	Better localized data allows for better forecasting of future numbers, which allows them to perform their jobs better and increase Beneficence.
	Non-maleficence	Increase	Ensures transmissions are lowered in a high risk environment, which reduces further suffering.
Commercial / NGO Land Employees	Non-Maleficence	Increase	Makes it easier for the organization to enforce local laws, not only helping to reduce transmission rates, but also reduce the chance of the organization to receive any fines.
	Beneficence	Increase	Can redirect resources previously used for checking to further their mission.
Government / Law Enforcement	Justice	Increase	Using a standardized model means everyone is (theoretically) treated the same which increases justice. However, there is no guarantee this is true and could actually systematically penalize groups of people, thus reducing justice.

	Non-Maleficence	Increase	Better data allows for better forecasting and planning

References

[1] timesler, 2020. mtcnn[Source code].
Available: https://github.com/timesler/facenet-pytorch

[2]cabani, 2020. MaskedFace-Net[dataset].
Available: https://github.com/cabani/MaskedFace-Net

[3] J. Nelson, "Mask Wearing Object Detection Dataset", Roboflow, 2020. [dataset].
Available: https://public.roboflow.com/object-detection/mask-wearing

[4] Alberta Health Services, "COVID-19 Scientific Advisory Group Rapid Response Report", 2020. [Online]. Available:
https://www.albertahealthservices.ca/assets/info/ppih/if-ppih-covid-19-sag-mask-use-in-community-rapid-review.pdf

[5]City of Toronto, " COVID-19: Orders & Bylaws", 2020. [Online]. Available:
https://www.toronto.ca/home/covid-19/covid-19-what-you-should-do/covid-19-orders-directives-by-laws/

[6] K. Breen, "Coronavirus face coverings under the nose equivalent to 'not wearing a mask': experts" , 2020. [Online]. Available:
https://globalnews.ca/news/7362921/coronavirus-experts-masks-properly-worn/

[7]"Image Classification - an overview | ScienceDirect Topics", *Sciencedirect.com*, 2020. [Online]. Available:
https://www.sciencedirect.com/topics/computer-science/image-classification.

[8]"Image Classification - an overview | ScienceDirect Topics", *Sciencedirect.com*, 2020. [Online]. Available:
https://www.sciencedirect.com/topics/computer-science/image-classification

Permissions

Kieran Cyphus
permission to post video: yes
permission to post final report: yes
permission to post source code: yes

Batuhan Raif Karagöz
permission to post video: yes
permission to post final report: yes
permission to post source code: yes

Won Jin Kim
permission to post video: yes
permission to post final report: yes
permission to post source code: yes