Final Report

Face Mask Detection

Team Number	6
Team Members	Haoyu Li Yitong Wang Zoe Zhang Xin Zhao
Prepared On	Dec 9, 2020, Wed
Word count	2316

1.0 Introduction

It has been a substantial change in people's life since the global COVID-19 outbreak. The dress code for public indoor spaces has been updated where the obligation to wear masks is mandatory [1]. However, the current situation is that most of the public indoor spaces like stores and shopping malls assign one or more security standing in front of the entrances. The need for security guards has an increasing trend, which indicates the significance to assure public health [2]. Meanwhile, there are statistics indicating that people attending security guards' jobs have the greatest risk of being infected [3].

Our project aims to introduce a machine learning algorithm that detects a person's face and identifies whether one wears a face mask. With a properly trained CNN model, it can accurately extract features and quickly classifies people with or without masks. The system can be connected with cameras and send notifications to security guards; this way, they don't have to constantly focus on the screen and can safely monitor the entrance without having physical contact with others.

Figure 1.1 Example of a security guard in front of a mall

2.0 Illustration / Figure

The design of a modified Yolo V1 backbone neural network is illustrated in the figures below:

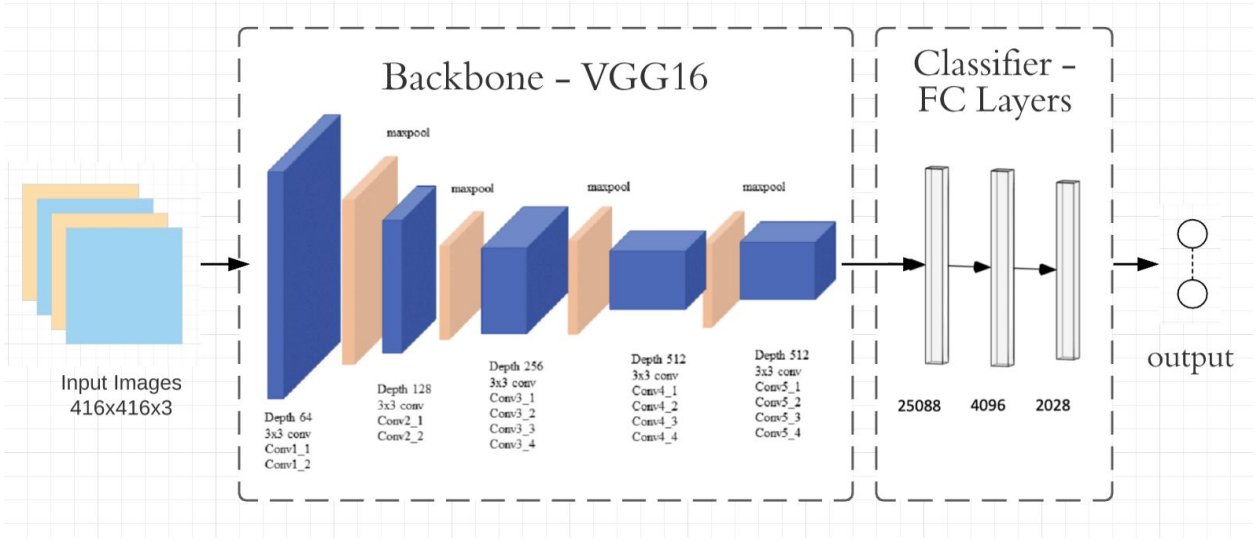

Figure 2.1 Backbone network of the Yolov1 algorithm

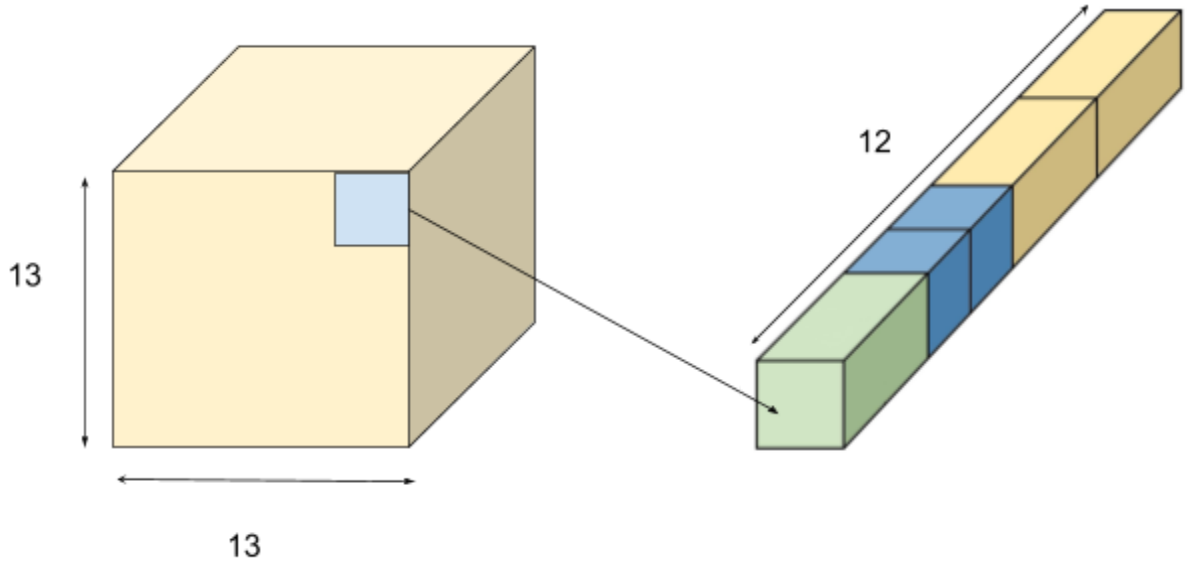

Figure 2.2 The output shape of the proposed model

3.0 Background & Related Work

The Covid-19 outbreak has been spreading for over half a year now, and the tool to help slow the spread is needed urgently and widely. Research says that countries with a higher percentage of the population properly wearing face masks have a lower death rate [4]. Being said that tools supervising the proper wear of face masks and reminding those who failed to follow the guidance from the Public Health of Canada are necessary. Luckily, there are existing solutions provided by various companies for exclusive uses, such as Uber [5]. The users will be required to use their front camera to verify that they have put on their face masks before requesting rides. Although lots of companies have invented tools that perform face mask detection, there is not any website available online that can be used by the general public. Thus, the market currently doesn't have a mature product that has such functionality.

4.0 Data Processing

We used three human faces datasets with and without masks to train and test our model. The Face Mask Dataset [6] contains 1,598 human faces images, the MAFA dataset [7] includes 3,006 images of human faces with face masks, and the WIDER Face dataset [8] contains 3,114 images of human faces without face masks. However, only a portion of these datasets was used since some images contained inconvertible labels, and some images contained irrelevant contents like sport faces images and outdoor face images. After removing the unusable images, we obtained 5,000 images in the training set and 120 images in the test set.

Figure 4.1 image samples from MAFA and WIDER FACE dataset

After obtaining the raw data, we first resized the images to 416x416 by resizing the relatively longer edge of the image and added padding to fill in the black spaces so that the scale of each image stayed unchanged. Then, we used the GaussianBlur function to make some images blurry so that the model could learn more features such as blurry faces. Lastly, we used the ColorJitter function to alter the input color of some images so that the model could learn features from images with overexposed images.

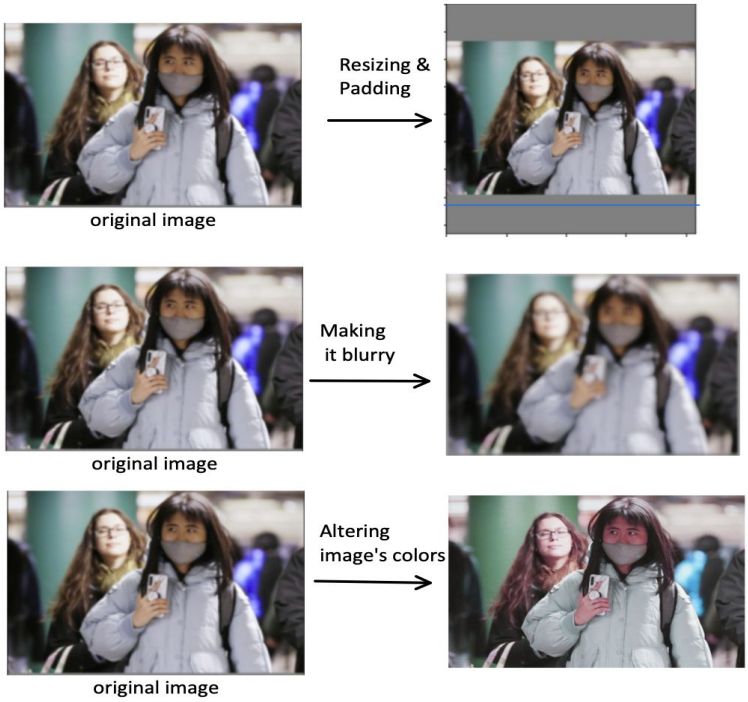

Figure 4.2 Example of image processing steps

Although the label structures were different among all three datasets, they had both bounding box and face type in common for each face in the images. Therefore, we manually modified the labels and converted them into a consistent format, each with a size of 13x13x12. To obtain the label in the desired form, we first segmented the image into 13x13 grids, each grid has a label of size 12. The first two values in a label indicate if a face mask exists in a single grid, whereas the last ten values show the confidence scores and the coordinate ratio of the bounding boxes relative to the size of each grid.

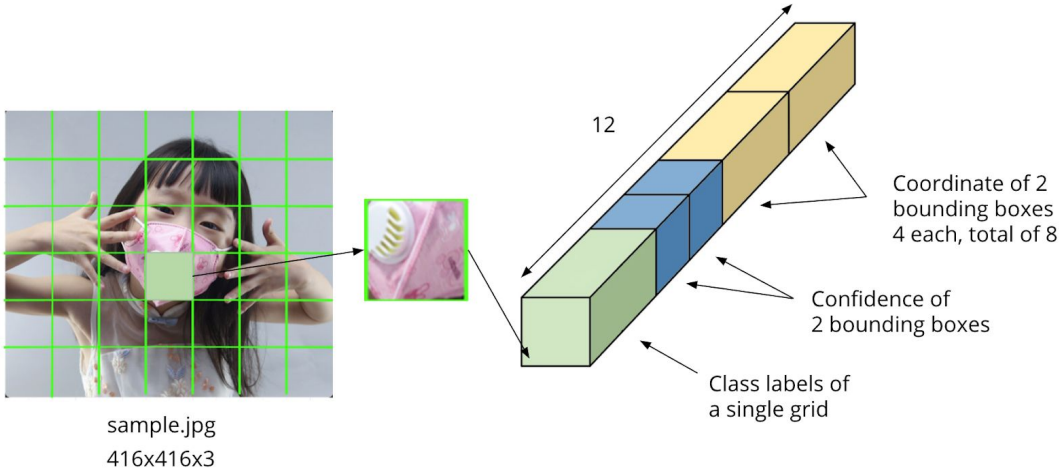

Figure 4.3 The label of one grid in a sample image

5.0 Architecture

The model that we used for this project was a deep convolutional neural network based on Vgg16, the pretrained weight from ImageNet [9] was used to speed up the training process and decrease the probability of overfitting. We modified the fully connected network of Vgg16, adding a reshape layer and sigmoid activation function at the end to match the labels of our dataset. The fully connected layers map the flattened tensor from the last layer of Vgg16 features which has a size of 25088 to a tensor with size 2028 and reshape layer convert the one dimensions tensor to 3 dimensions with size 13x13x12.
The input of our model was a RGB image with size 416x416 and output was 13x13 grids with 12 labels indicating the class of the face and 2 bounding boxes positions in each grid cell. We used the loss function proposed in yolov1 paper [10] to train our network.

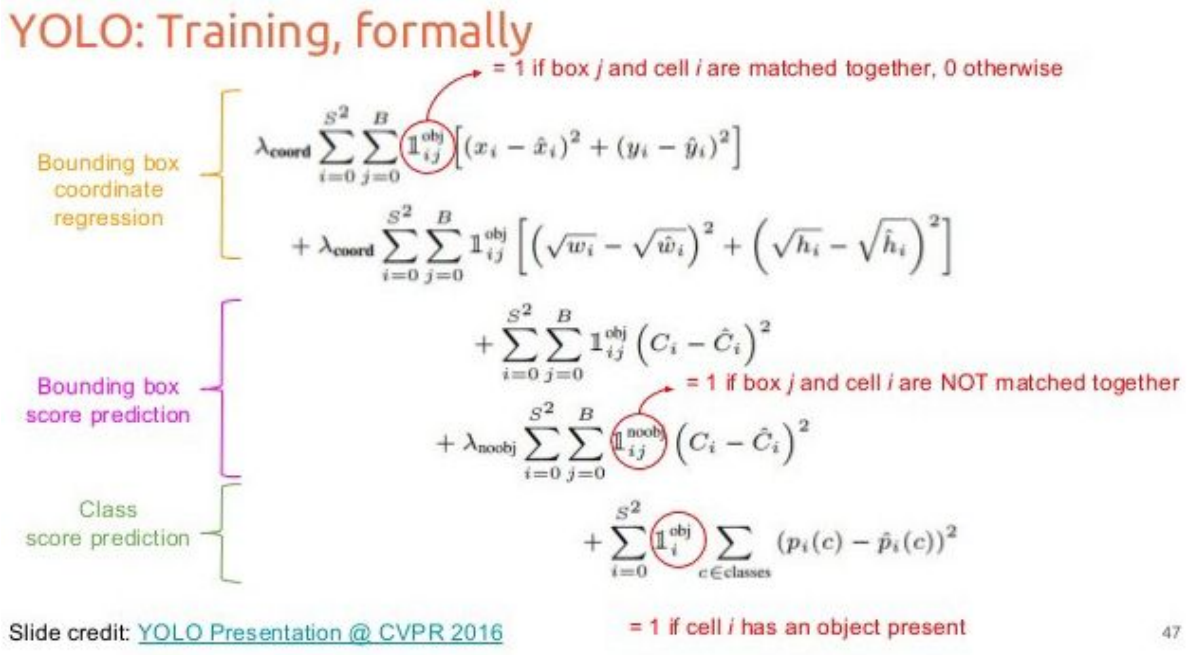

Figure 5.1 Loss function to train our network [10]

6.0 Baseline Model

The baseline model we used in this project was a simple convolution neural network with an input vector size of batch size * the size of images (416 * 416 * 3) and an output size of batch size * 13 * 13 * 12. The output vector of the model indicates two predicted classes, two confidence scores, and eight coordinate ratios of the bounding boxes for each grid in the input images. The structure of the baseline model was consist of the following layers:

- two convolutional layers with input channels of size 3 and output channels of size 3
- two max-pooling layers with kernel sizes equal to 2 and stride sizes equal to 2
- three fully-connected layers with an output size of batch size*13*13*12

The baseline model generated a training accuracy of 4.1% and a test accuracy of 2.4%. This poor performance was expected because it had fewer convolutional layers than our proposed model; therefore, it had less ability to extract and learn complex features.

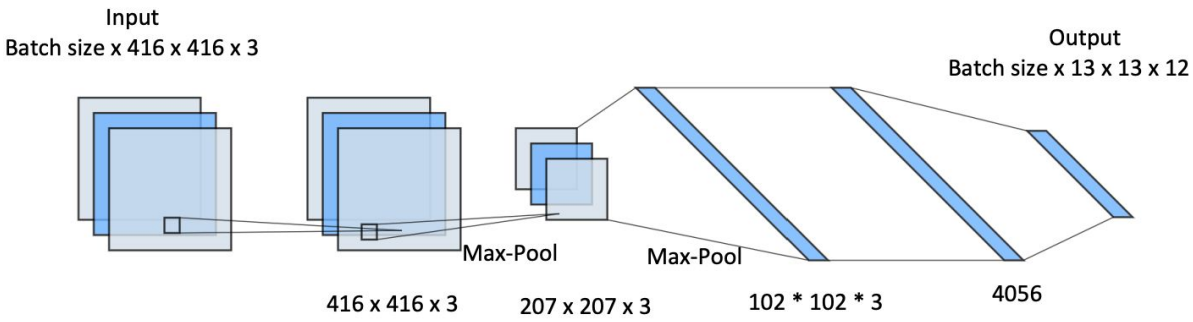

Figure 6.1 The architecture of the baseline model

7.0 Quantitative Results

The qualitative result of the primary model was analyzed from two aspects, the training loss and the confusion matrices of datasets.

7.1 Loss result

The training loss was calculated based on the bounding areas of the faces and the classification of the face which was discussed in section 5.0. In Figure7.1, the horizontal axis was iterations count, and there were 20 epochs with 62 in each that the horizontal axis goes to $62 * 20 = 1240$ iterations. Through the training process, the loss on the inaccurate bounding boxes and miss-classified faces was reduced. The final total training loss in the last epoch was at around 2 from originally the peak value 16 (Figure7.1). From the trend of the loss, we could conclude that the model was not underfitting.

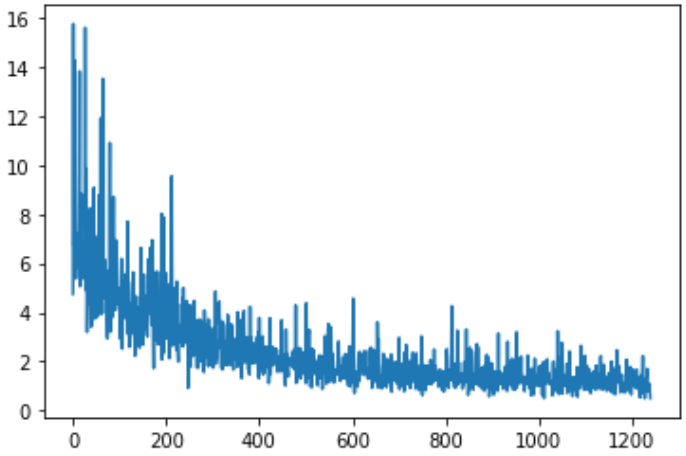

Figure 7.1 Loss per iteration

We also evaluated the quantitative result by taking a more detailed look at the training loss on only the classification of faces into mask and without mask categories. From Figure7.2, neglecting the only peak value 7, the loss on classification dropped from around 3 to below 0.5.

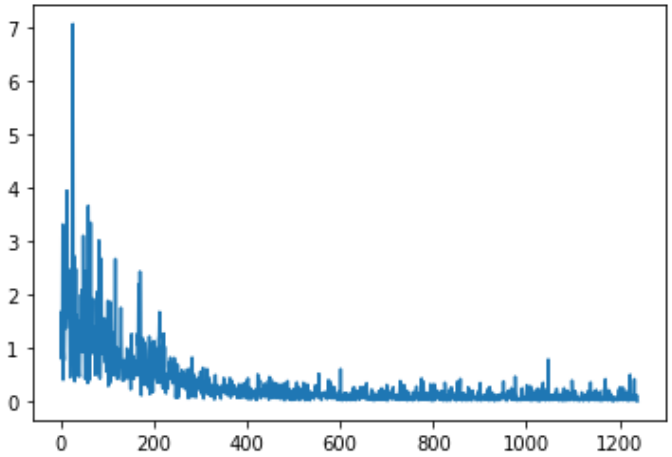

Figure 7.2 Loss on classification per iteration

7.2 Confusion matrix analysis

Meanwhile, we evaluated the training outcome by collecting the information on the training dataset to fill in the confusion matrix. Since the model returned bounding boxes around the faces and also gave the predicted class of the bounded areas, thus the confusion matrices had separated columns for 2 fields. We set up a bounding box area difference threshold[1] that filters out the imprecise bounding results. With the threshold being set to 0.4, the imprecisely bounded results were counted as FN. Also, the un-bounded faces were counted into FN. The mistakenly bounded non-face area results were counted as FP. For the bounding boxes that passed the threshold, TP, FN, FP and TN were calculated regularly with positive being with mask and negative being without mask.

Based on the information in Table7-1, the accuracy of the training dataset was calculated using the formula $Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)}$, which has a value of 83.56%.

Table 7-1 Confusion Matrix on Training dataset

Training dataset	Predicted Positive (classification)	Predicted Negative (classification)	Predicted Positive (bounding box)	Predicted Negative (bounding box)
Actual Positive	7820 (TP)	14 (FN)	-	2051(FN)
Actual Negative	30 (FP)	3072 (TN)	48(FP)	-

In order to evaluate the performance of the face detection model, we computed the results using labeled test dataset. Using the same interpretation method, the accuracy on the testing dataset had a value of 67.43%. Since the accuracy for testing dataset was lower than the performance on training dataset, we think that the model had a slight chance of being overfitted.

Table 7-2 Confusion Matrix on Testing dataset

Testing dataset	Predicted Positive (classification)	Predictive Negative (classification)	Predicted Positive (bounding box)	Predicted Negative (bounding box)
Actual Positive	352 (TP)	0 (FN)	-	182 (FN)
Actual Negative	0 (FP)	62 (TN)	18 (FP)	-

To summarize the results from analyzing both training and testing datasets, comparing the FP and FN fields for bounding boxes prediction, the results falls towards FN more, which means that the model was more likely to miss a face than inaccurately recognize something else as a face.

[1] Threshold is the IOU of the bounded area, calculation discussed in Section 5.0

8.0 Qualitative Results

The qualitative result of the primary model will be discussed over the testing examples. We labeled images from the testing dataset. The boxes indicated the results from face detection, and the green boxes meaning that the classification result from the pre-discovery faces wearing masks while the red boxes mean they were not.

8.1 Performance on different face resolutions

The model was able to perform a satisfying prediction on finding complete faces with different resolutions. The given images contain different scales of the human faces. For example, the below 2 images are a wide shot (Figure 8.1) and a close-up shot (Figure 8.2) respectively. Since all images have been resized to the same, the resolutions of the faces are different resolutions, where the close-up shot has a relatively higher resolution than the wide shot. Meanwhile, comparing the outputs from both sample images, the performances are similar, indicating that the model is capable of identifying complete faces with both high and low resolution.

Figure 8.1 Wide shot example

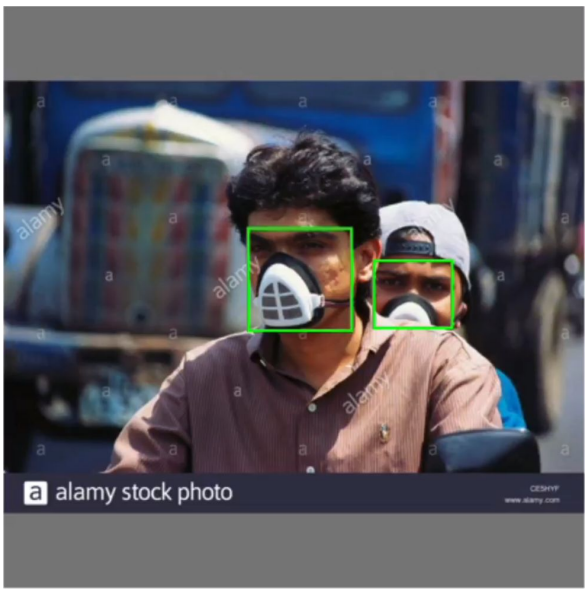

Figure 8.2 Close-up shot example

8.2 Performance on incomplete faces

The previous analysis was based on a condition that the faces in the images must be complete for the model to correctly identify. Meanwhile, we got an experimental result that the model was not able to identify incomplete faces as efficient as complete faces. In Figure 8.3, there are two faces that should be bounded while those were not. Shown in separate figures 8.4 and 8.5, the two faces were not complete with face 1 missing the partial top and left face, and face 2 missing the right part of his(her) face. In Figure 8.3, the model was able to identify all complete faces while only able to identify some of the incomplete faces.

Figure 8.3 Sample image with incomplete faces

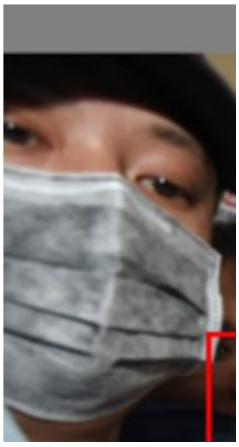

Figure 8.4 Face1

Figure 8.5 Face2

9.0 Evaluate model on new data

To evaluate our model's performance in real-life and on unseen data, we performed a real-time face mask detection through webcam. Since our project will be used together with CCD (closed-circuit television camera), the inference speed and detection accuracy will be taken into considerations in most of the scenarios.

To complete the above test, we used python with OpenCV to catch the frame stream of our computer's webcam and display the results on the monitor. The caught frames will be sent to our model to get the predictions, and the results from the models are drawn on the frames. The inference speed is also monitored by the fps (frames per second) shown on the output video stream; for this example, we used Nvidia RTX 2070 as an accelerator for the model.

Since we cannot go out and get the data in public places because of the pandemic, we simulated the real-life situation indoors. One of us took a video and performed some movements to see how well the model's detection was in different situations.

Figure 9.1 Real-time detection using our model and webcam

The screenshots show that our model performs well on detecting faces with various postures and lighting conditions and classifying faces with or without a mask. It is interesting to see that even if someone covers his nose with a hand, the model will still classify it as a no-mask, but if the nose is covered by something else, the model will think of it as a mask.

The fps shown in one of the screenshots is 38, and it is sufficient for real-time detection since most of the webcams or CCDs are working on 24 fps.

To summarize, our model meets our expectation of detecting and classifying faces on newly generated data and still keeps a relatively low inference time.

10.0 Discussion

Quantitatively, the false positive rate of classifying an identified face indicates that the model misclassified a non-mask face as a masked face, which means people without masks are allowed to pass the mask check. Thus, minimizing the classification false positive rate is essential to work on to prevent the situation discussed.

There are various types of face-covering available on the market. For example, there are medical face masks, dust masks, respirators, etc. The model currently can classify faces regardless of the face-covering type. However, some coverings need to be detected as negative such as covering using hands or wearing a face mask without covering the nose. Currently, our model can successfully distinguish covering using hands from regular face masks. Also, from section 9.0, we can see that our model didn't classify correctly when the covering is an AC remote (Figure 9.1), which is classified mistakenly as positive. A logical guess is that our model classifies based on the color difference of the face. If the face mask's location has a different color from other parts of the face, it is considered positive. This is an interesting observation that the team will further investigate. Lastly, our model doesn't perform well if two or more human faces are too closed because the performance is limited by the grid size we defined in the model as each grid can only detect one human face.

11.0 Ethical Considerations

The input collection and bias in the dataset are two major ethical concerns of the model. The input requires real-time pictures and videos, which could raise ethical issues as it is done mandatorily and without consent beforehand. This is an ethical issue between personal rights and public health that should be considered when implementing the model. The classification of the faces could be biased because of the bias in the training dataset. Our training dataset is composed of pictures mostly from China and some from Canada, the US, UK, etc. Thus the model has some bias when classifying people from different countries. This could be improved by gathering images from more sources to reduce the bias of the dataset.

12.0 Link to Github or Colab Notebook

Google Colab
https://drive.google.com/file/d/11hdaWS9QNTxrlbhT9ksTuhtbHHGqSkz9/view?usp=sharing

13.0 References

[1]"About e-Laws", *Ontario.ca*, 2020. [Online]. Available: https://www.ontario.ca/laws/about-e-laws#ccl. [Accessed: 18- Oct- 2020].

[2]"Security guards in high demand amid COVID-19", *Iheartradio.ca*, 2020. [Online]. Available: https://www.iheartradio.ca/max-104-9/news/security-guards-in-high-demand-amid-covid-19-1.12308335. [Accessed: 18- Oct- 2020].

[3] A. Joshi, "Coronavirus: Security guards are most at risk of dying with COVID-19, figures show", *Sky News*, 2020. [Online]. Available: https://news.sky.com/story/coronavirus-security-guards-are-most-at-risk-of-dying-with-covid-19-figures-show-12015241. [Accessed: 18- Oct- 2020].

[4]P. Care, "Still Confused About Masks? Here's the Science Behind How Face Masks Prevent Coronavirus", *University of California San Francisco*, 2020. [Online]. Available: https://www.ucsf.edu/news/2020/06/417906/still-confused-about-masks-heres-science-behind-how-face-masks-prevent. [Accessed: 18- Oct- 2020].

[5]S. Kansal, "Protecting One Another | Uber Newsroom US", *Uber Newsroom*, 2020. [Online]. Available: https://www.uber.com/newsroom/protecting-one-another/. [Accessed: 18- Oct- 2020].

[6]Purohit, A. (2020, July 30). Face Mask Dataset (YOLO Format). Retrieved December 08, 2020, from https://www.kaggle.com/aditya276/face-mask-dataset-yolo-format

[7]MAFA (MAsked FAces) - CKAN", 221.228.208.41, 2020. [Online]. Available: http://221.228.208.41/gl/dataset/0b33a2ece1f549b18c7ff725fb50c561. [Accessed: 18- Oct- 2020].

[8]The Chinese University of Hong Kong, T. (n.d.). WIDER FACE: A Face Detection Benchmark. Retrieved December 08, 2020, from http://shuoyang1213.me/WIDERFACE/

[9]"ImageNet", *Image-net.org*, 2020. [Online]. Available: http://image-net.org/download. [Accessed: 18- Oct- 2020].

[10]Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016, May 09). You Only Look Once: Unified, Real-Time Object Detection. Retrieved December 10, 2020, from https://arxiv.org/abs/1506.02640