Masks Unmasked

APS360 Project Final Report

Prepared by:

Maitreyi Joshi	
Shaziah Gafur	
Snehal Bhattacharya	
Ramakrishna Natarajan	

Word Count: 2485, no penalty

1.0: Introduction

“The current pandemic has necessitated the use of masks to protect ourselves as well as the community from the virus.However it is nearly impossible for humans to continuously monitor and enforce mask wearing laws in crowded places like malls and public transport [1].” As we return to this ‘new normal’, “Governments need surveillance on people in crowded public areas, to ensure that wearing face masks laws are applied [2].” “This could be applied through the integration between surveillance systems and Artificial Intelligence models [2].”

Our project aims to automate this process of identifying the defaulters by classifying faces as “with” or “without” mask. The amount of effort required for a human being to identify defaulters and the inevitable lapses when performing this task in a crowded place makes Machine Learning a perfect solution to automate this problem. In addition, we incorporate age detection into our model to check if kids can be exempted from wearing a mask [3]and if there are high risk age groups violating the rules.

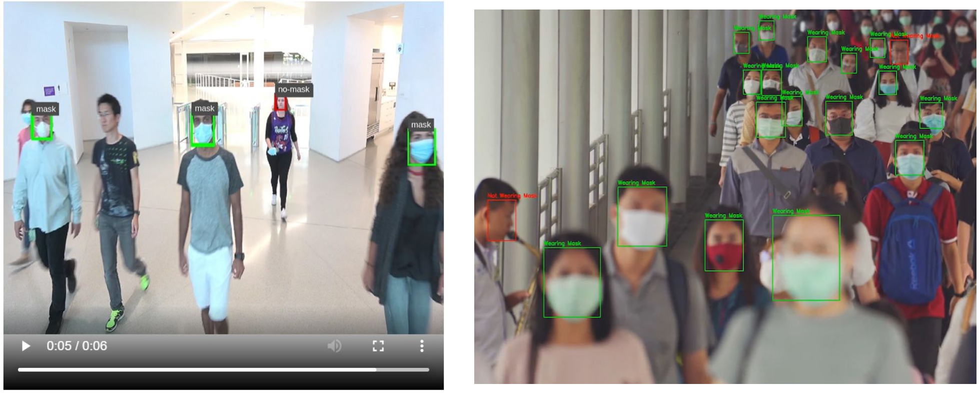


Fig 1.1- Some real life applications of our project[4][5]-This is the end goal of our project.

2.0: Illustration

The following flow chart summarizes our Mask Detection pipeline.

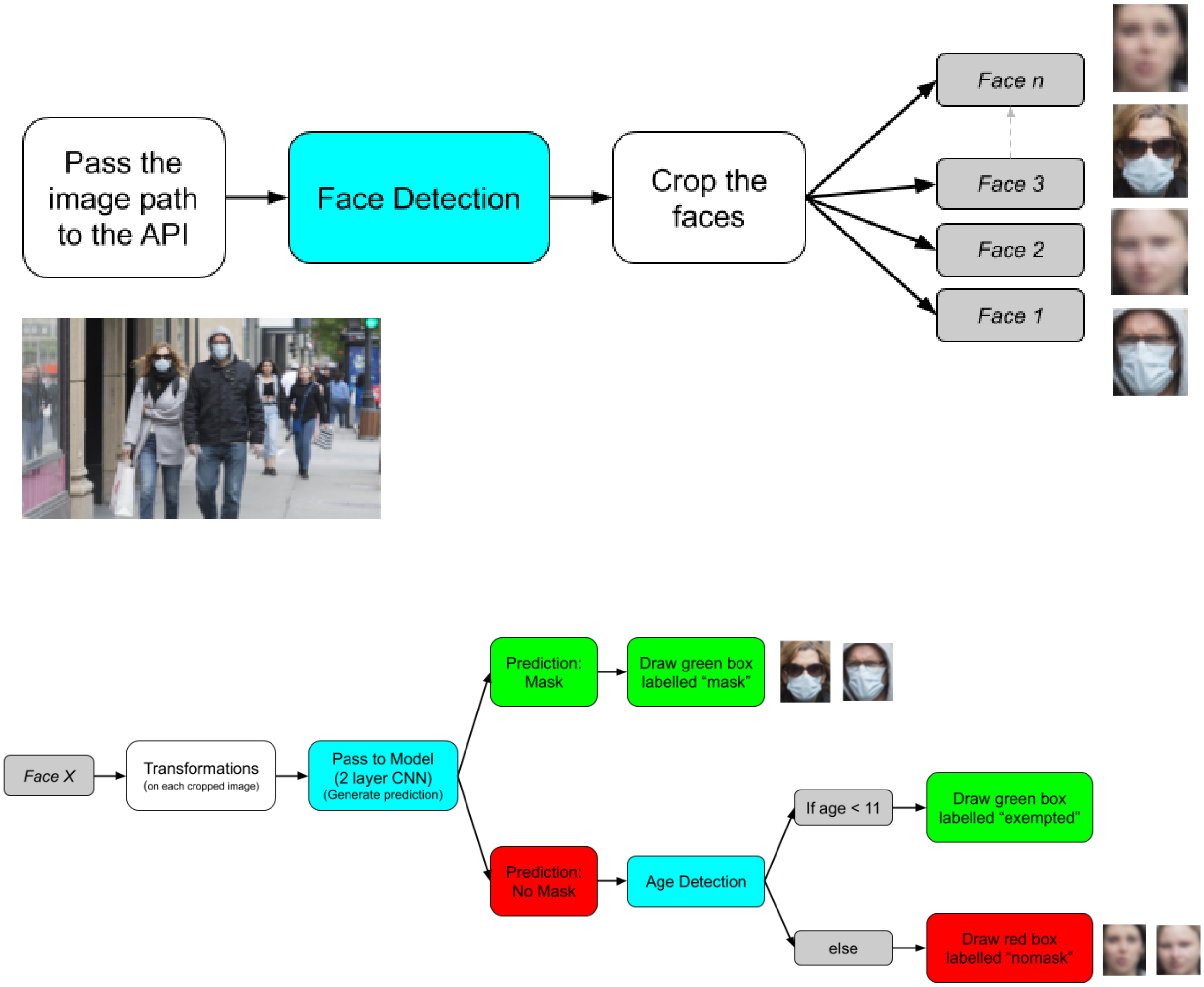
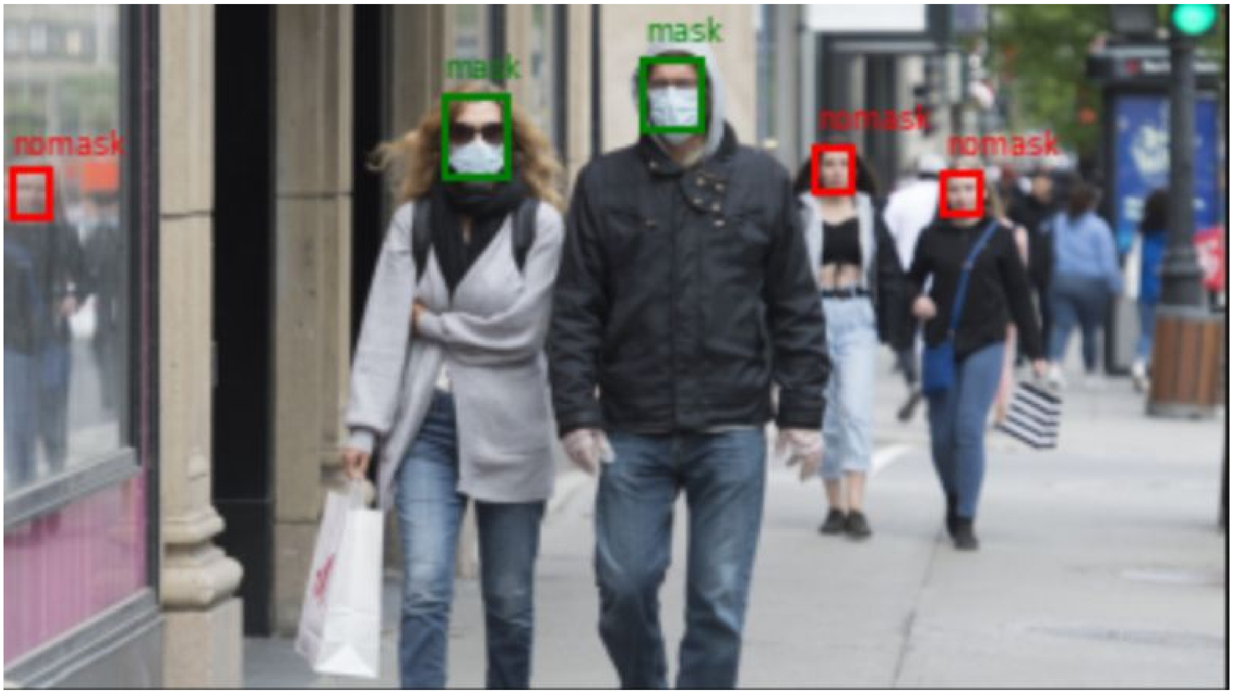


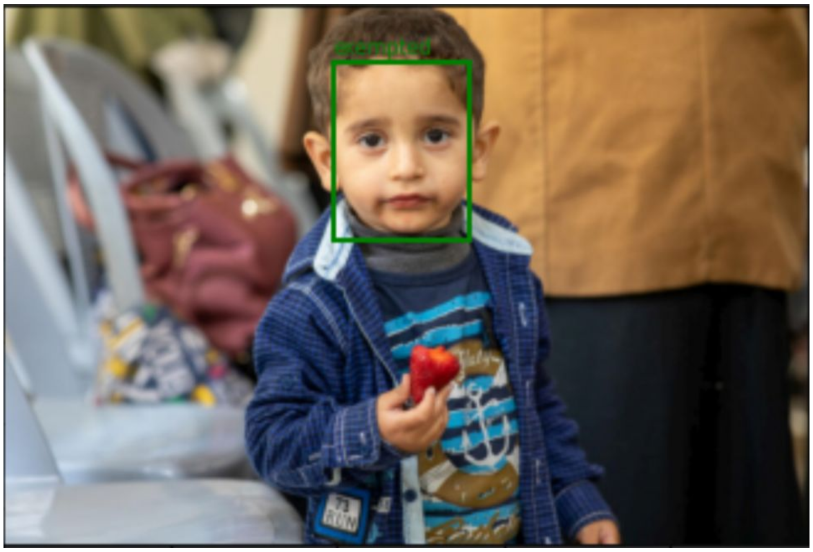
Fig:2.1.The image detection algorithm is "face recognition" from torch.mtcnn

The following image is an example output that we receive when we call our Mask Detection API on an image

Example Image 1: Mask and No Mask



Example Image 2: Exempted



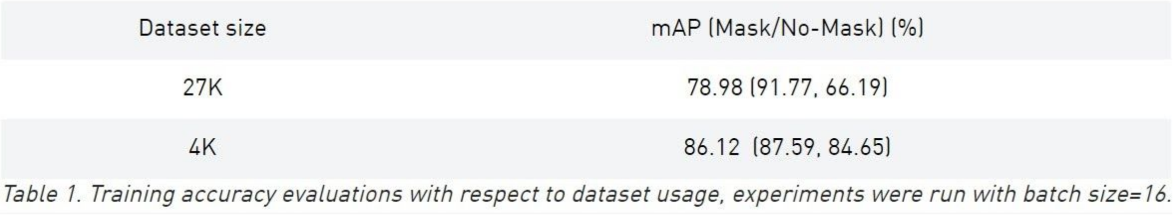
3.0: Background and Related Work

“With face masks becoming mandatory in most countries,many Organizations particularly those in areas of Machine Learning,have pursued AI-based faceMask detection [6].”

3.1: Nvidia face mask detection model

“Nvidia’s face mask detection model is trained on its Transfer Learning Toolkit [4][6]”. Trained on “ResNet-18 model as backbone, and input image with a resolution of 960×544 [4][6]”, the reported “accuracy varies between 78-86% based on the size of the data set [4][6]”. The data sets used are “FDDB,WiderFace datasets (without mask),MaFA and Kaggle Medical mask datasets (with masks)[4][6]”.

Table 3.1: State of the art results for Face-mask detection[B]



Dataset size	mAP (Mask/No-Mask) (%)
27K	78.98 (91.77, 66.19)
4K	86.12 (87.59, 84.65)

Table 1. Training accuracy evaluations with respect to dataset usage, experiments were run with batch size=16.

3.2: Research

“Another relevant work in this field is found in a research paper that proposes hybrid model for face mask detection[6].” “The proposed model consists of ResNet 50 for feature extraction and 3 algorithms-decision trees,Support Vector Machines(SVM) and ensemble algorithm for face mask classification [7][6].” “It compares the performance of three classifiers on 3 different data sets based on validation,testing accuracy,performance metrics, and consumed time[7][6].”

“The best performance is of SVM as reported from the paper(Accuracy~99 %) [7].”
“The classifier created achieved a 100 percent accuracy when it was tested on the Labeled Faces in the Wild(LFW) dataset [7].”

4.0: Data Processing

4.1: Data Description

The following table illustrates our datasets and the preprocessing we conducted on them.

Datasets	Description	Pre Processing	Number of images used
1.Kaggle Medical mask dataset.[8]	1.Mostly High Resolution labelled masked and unmasked images	1.No pre-processing. 2.Images used from here for balancing dataset	1500 images to supplement other processed data.
2.Kaggle Medical Mask Dataset TF Records[9]	1.Images with multiple people-masked and unmasked, XML files with bounding box coordinates	1.XML script parsing and cropping faces out of photo based on coordinates 2.Blurred images removed.	2250 out of 2300 images used.
3.Wider Face Dataset[10]	1.Photos of multiple unmasked people in a single image with no labels available.	1.Face detection- (face_recognition from torch.mtcnn) algorithm applied 2-Bounding boxes generated 3.Images cropped	2250 images.

4.2: Parsing XML Files

The Kaggle Medical Masks Records dataset contained images of multiple masked and unmasked people.Each image in this dataset also had a corresponding XML file. Using the coordinates from XML files, we obtained the bounding boxes and cropped out faces from the photos, and saved them to appropriate directories based on the labels.

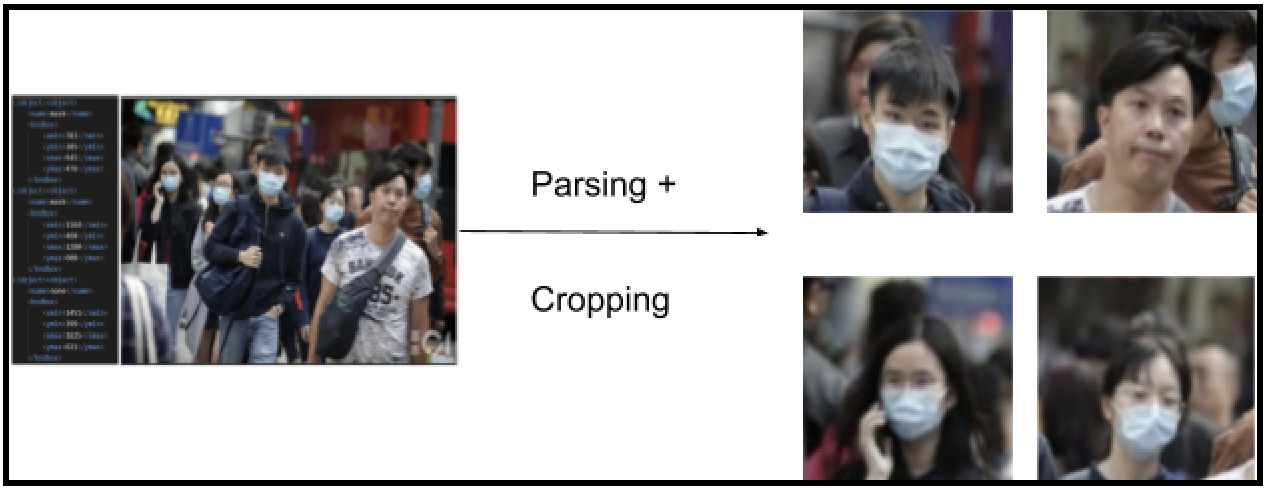


Fig 4.2: Snippet of the XML parsing code and corresponding images that it cropped.

4.3: Eliminating Blurry Pictures

Some of the images in our dataset were blurry-particularly the ones which were in the background or not in focus. We identified the Laplacian variance of the image using the OpenCV Library. The Laplacian variance for blurry images are small and for sharper images are large as can be seen below.

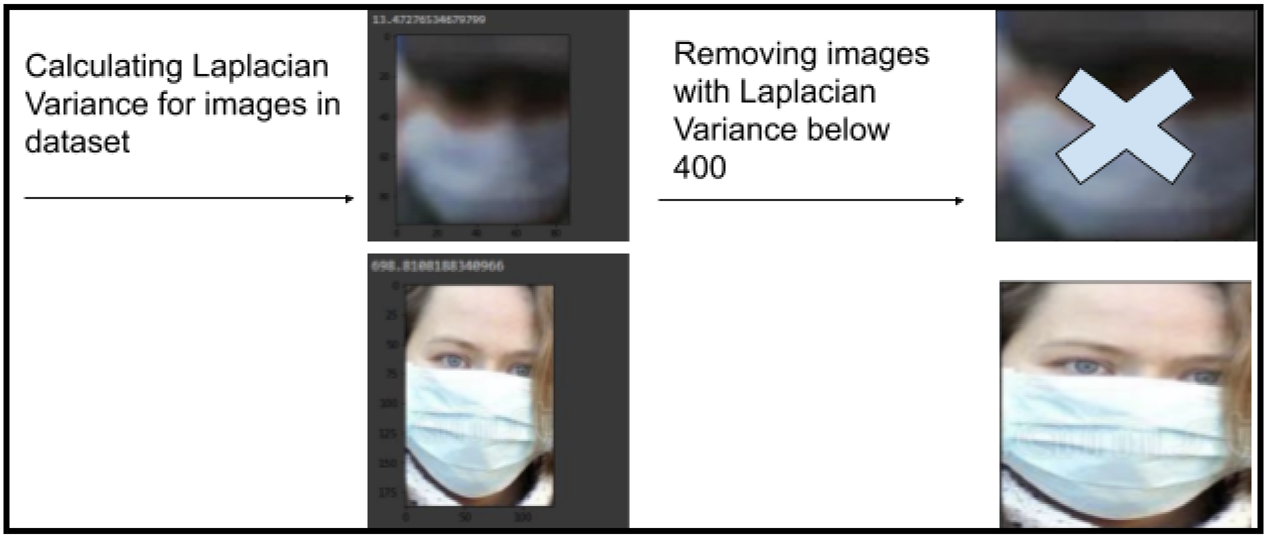
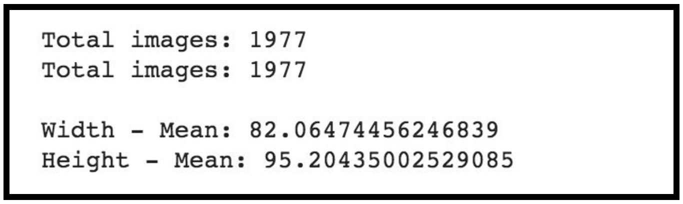

Calculating Laplacian Variance for images in dataset
13.472765346679799
698.8108188340966
Removing images with Laplacian Variance below 400

Fig 4.3: Showing how we used Laplacian Variance to prune data.

We used 400 Laplacian variance as a threshold and eliminated all the blurry images that had a Laplacian variance higher than the threshold.This threshold was chosen based on us going through the a few images in the dataset to inspect its quality and to ensure that we did not end up removing too many images.

4.4: Eliminating Low Resolution Pictures

We eliminated images with very low resolution from our training data as they would be of very low quality after scaling and poses a problem in training.
We performed a statistical analysis on our dataset to find out the mean height and width of our images , the results of which are below.



```
Total images: 1977
Total images: 1977

Width - Mean: 82.06474456246839
Height - Mean: 95.20435002529085
```

We used these values to find the outliers in our dataset with heights and widths that varied greatly from the mean and eliminated them.
In addition, we applied data augmentation techniques to reduce overfitting. These include random horizontal flip and transforming images to grayscale for faster training.
The following image shows a set of the final images after pre processing (Figure 4.3).

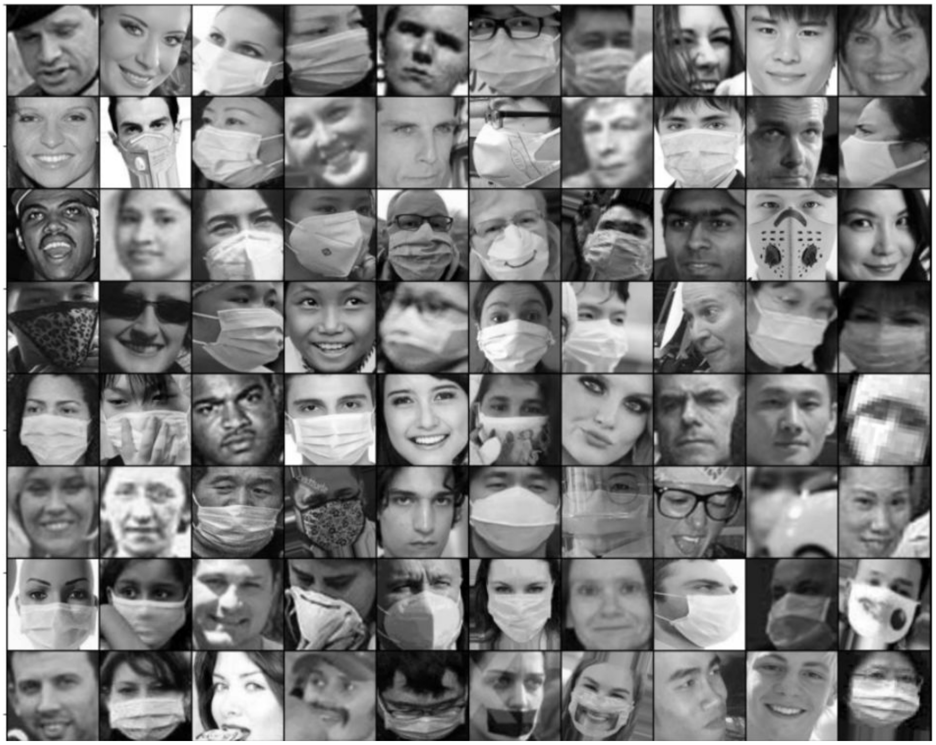


Figure 4.3: Final Processed data images

5.0: Architecture

“The following image illustrates the CNN model (Mask Net) we built for classifying images as masked or unmasked. We decided to treat it like a multi-class classification problem with the output being two neurons and use Cross Entropy loss and Adam Optimizer. 2 neurons make it easy for the model weights to train[1].” The striking feature of this model is its computational and design simplicity. For comparison, our model’s memory use is about 40 times lower compared to VGGNet[11] and has about 30 times less number of weights.

The second figure also indicates the transfer learning option that we tried. We used Google Net to extract features (by freezing all layers) and then training a simple ANN classifier on top of it (with input equal to number of features produced and output equal to 2 neurons)

For age classification, we used pyAgender 0.0.9. It is a simple Face Age & Gender detection tool on OpenCV. Age detection runs only after the mask detection model classifies “no mask”.

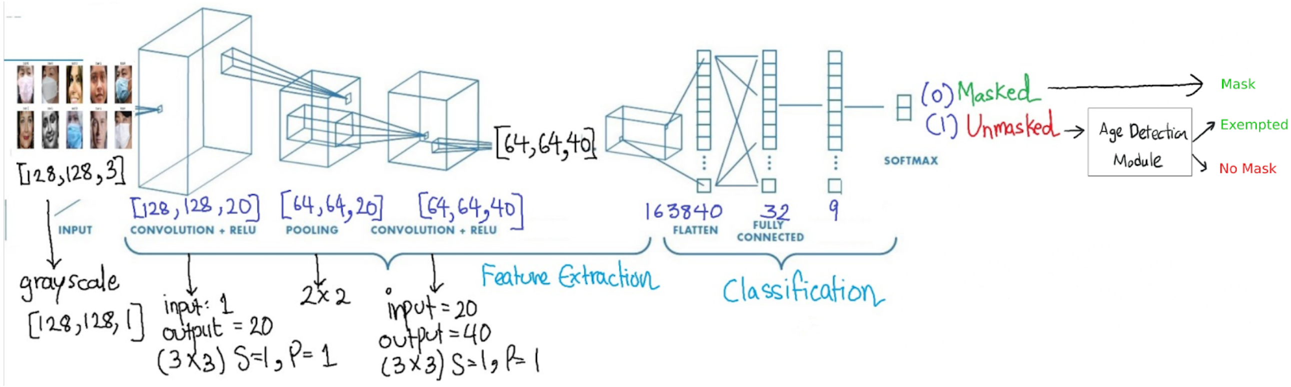
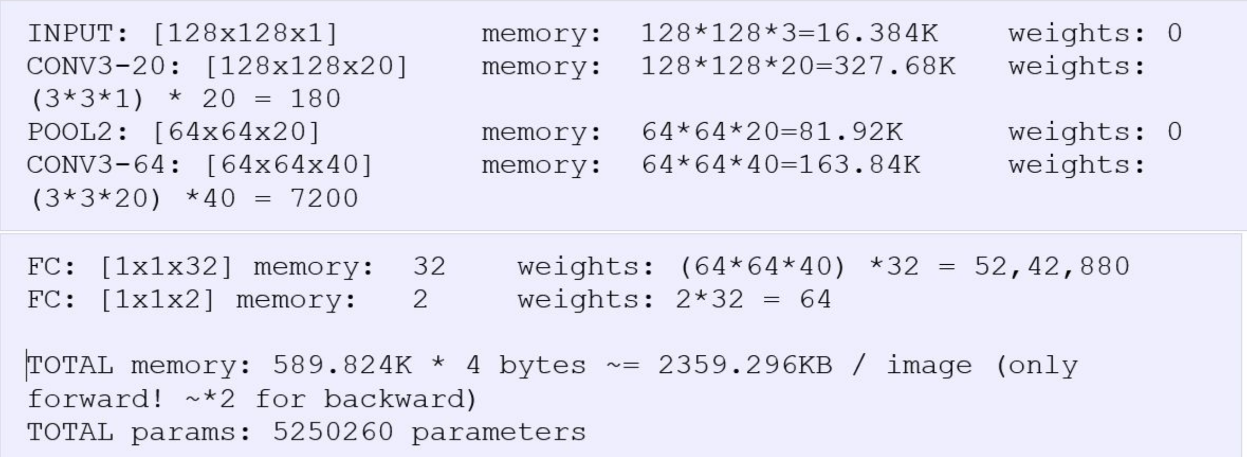


Fig 5.1: Our model Architecture[1] (CNN-Mask Net)



```
INPUT: [128x128x1]        memory:  128*128*3=16.384K    weights: 0
CONV3-20: [128x128x20]    memory:  128*128*20=327.68K   weights:
(3*3*1) * 20 = 180
POOL2: [64x64x20]         memory:  64*64*20=81.92K      weights: 0
CONV3-64: [64x64x40]      memory:  64*64*40=163.84K     weights:
(3*3*20) *40 = 7200

FC: [1x1x32] memory:  32    weights: (64*64*40) *32 = 52,42,880
FC: [1x1x2] memory:   2     weights: 2*32 = 64

TOTAL memory: 589.824K * 4 bytes ~= 2359.296KB / image (only
forward! ~*2 for backward)
TOTAL params: 5250260 parameters
```

Fig 5.2: Computational Efficiency of the CNN we created[11].

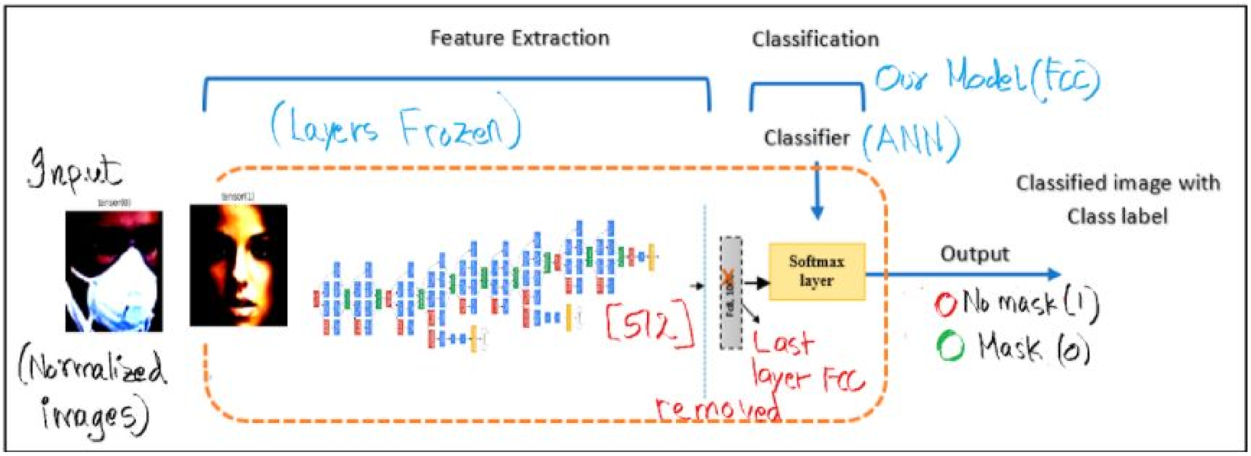


Fig 5.3: The Transfer Learning Solution: Feature Extraction using Google Net and ANN classifier trained.

6.0: Baseline Model

Our team selected a simple Random Forest algorithm as the best choice for the baseline model. Random Forest classification generally works by randomly creating a series of subsets to produce several decision trees[12]. These trees are designed to be uncorrelated such that their individual errors are insignificant when the entire group of trees are considered[12]. Each decision tree produces its own classification output. The most favoured class among the decision trees is determined to be the Random Forest Classifier prediction [12].

The baseline model used the same image processing techniques as the input to our primary CNN model.

The RandomForestClassifier function from the sklearn.ensemble library was applied in a similar manner as in Lab 0.

7.0: Quantitative Results

Our Random Forest baseline, produced an accuracy of 81.2%. Both candidates (the transfer learning solution and our Mask Net-CNN) performed better than the baseline.

After multiple trials, the best set of hyperparameters was found to be as following:

- Number of Epochs: 10 to 20
- Learning Rate: 0.001
- Batch Size:128

The standard choice of 80-20 Train-Test split worked well.

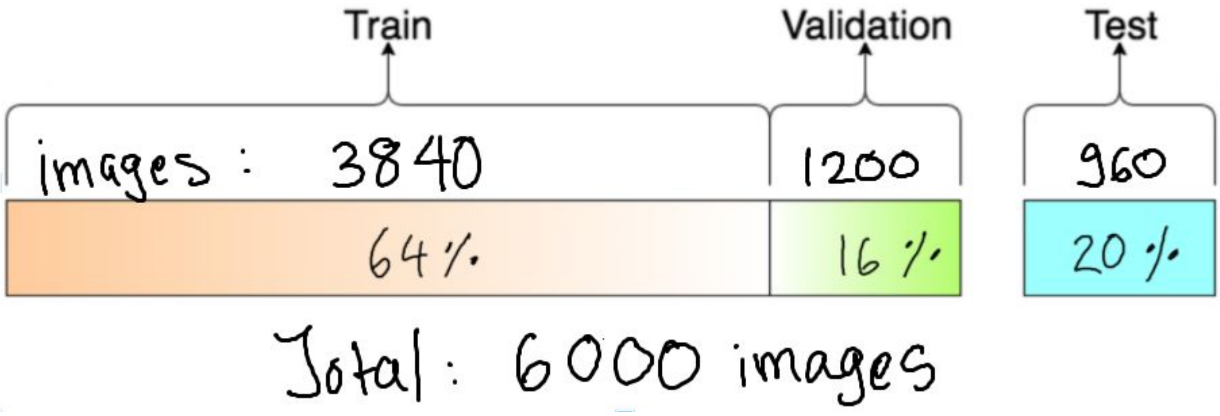


Fig 7.1: The split ratio used on our dataset.

Table 7.1 and 7.2 below illustrates the quantitative results for 2 of our candidate models-one of which was chosen as the final model.

Table 7.1: Mask Net CNN results

Experiment No.	Experiment Explanation and Numerical Results	Observations
1. Mask Net-CNN	Data augmentation was introduced on the data set to prevent overfitting (initially observed in training-refer appendix for curves without data augmentation) Specifically, a Random Horizontal Flip was applied, producing the loss function (Figure 7.2a) and training and validation curves (Figure 7.2b).	After data augmentation was applied,the model took longer to train but overfitting has reduced (compared to case without data augmentation-refer appendix) The final accuracy was as follows: ● Training - 91.8% ● Validation - 89.8%

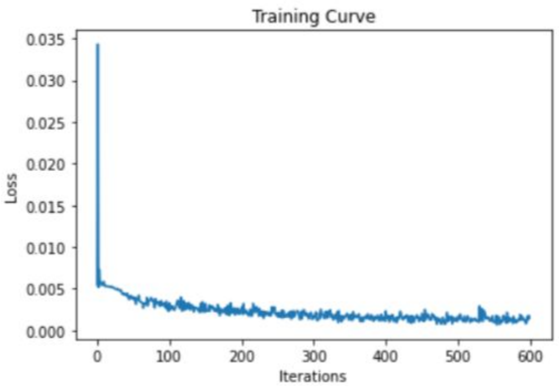
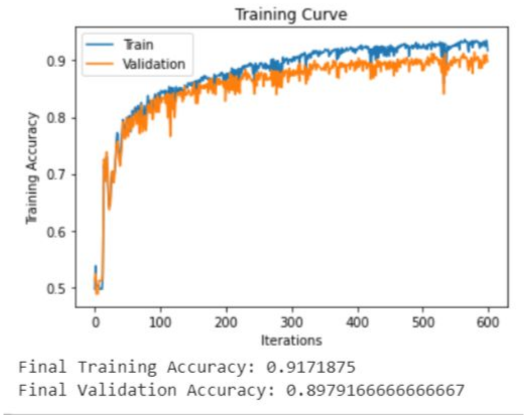


Fig 7.2a



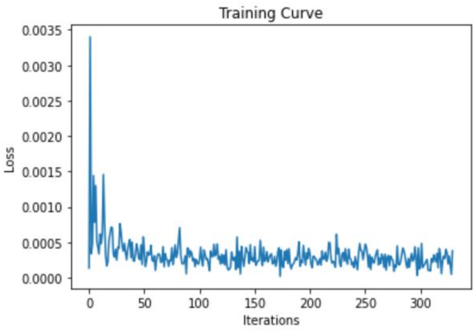
```
Final Training Accuracy: 0.9171875
Final Validation Accuracy: 0.8979166666666667
```

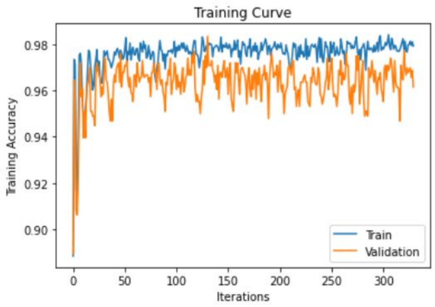
Fig 7.2b

Fig 7.2a: Loss function on model with data augmentation.
Fig 7.2b: Training and Validation curves on model with data augmentation.

Table 7.2: Transfer Learning Model results

Experiment No.	Experiment Explanation and Numerical Results	Observations
2.Transfer Learning Model	This is the Transfer Learning solution-Google Net+ANN classifier. The loss function (Figure 7.5a) and training and validation curves (Figure 7.5b) are illustrated.	The model took half as long to train (compared to Mask Net) and produced much higher accuracy, even with data augmentation applied. There is little to no overfitting The final accuracy was as follows: ● Training - 97.9% ● Validation - 96.1%





```
Final Training Accuracy: 0.9791666666666666
Final Validation Accuracy: 0.9614583333333333
```

Fig 7.3a

Fig 7.3b

Fig 7.3a: Loss function of Transfer Learning Solution.
Fig 7.3b: Training and Validation curves -Transfer Learning Solution.

Performance on Test Data of both the candidate models

Mask Net(Our CNN)
Final Test Accuracy: 94.5%

Transfer Learning(Google Net)
Final Test Accuracy: 98%

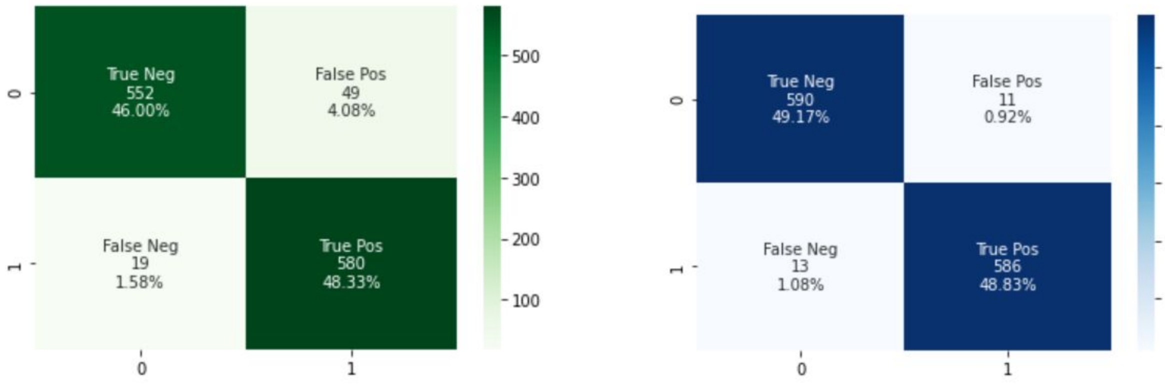
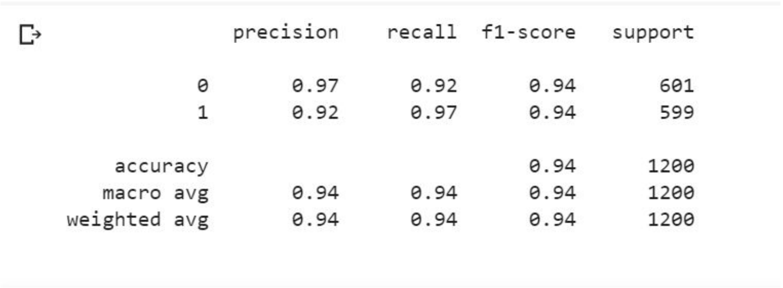


Fig 7.4: Confusion Matrices for both the candidate models. Outputs of “mask” and “no mask” are represented with 0 and 1 respectively. The number in each of the 4 quadrants describes the total number of times that particular scenario was the outcome.

The most frequent scenarios are True Negative (TN) and True Positive (TP), indicating it is accurate(94.3% of input test data). The False Positive (FP)section occurs 2.5 times more than False Negative (FN). Therefore, the model’s errors are more commonly misclassifications of masked images than unmasked images.

Additional measurements including Precision, Recall and F1 score have been computed (see Figure 7.5). Precision is the ratio of TP with the sum of TP and FP. Recall is the ratio of TP with the sum of TP and FN. F1 is computed as such:

$$F_1 = \frac{2\,TP}{2TP + FP + FN}$$



```
              precision    recall  f1-score   support

           0       0.97      0.92      0.94       601
           1       0.92      0.97      0.94       599

    accuracy                           0.94      1200
   macro avg       0.94      0.94      0.94      1200
weighted avg       0.94      0.94      0.94      1200
```

Figure 7.5: Precision, Recall and F1 Score calculations of the test data.

All three scores: Precision (92%), Recall (97%) and F1 (94%), are close to 100%. The Precision score is lower than the Recall score, implying that the model produces many more false positives than false negatives. For our project, cases with false positives are less costly than those with false negatives because tending to misclassify masked images as “no mask” portrays our model as more “sensitive”. It is safer to incorrectly assume an individual is not wearing a mask than to incorrectly assume an individual is wearing a mask. Thus, we would expect our model to be successful if it has high precision and recall scores, but also has the recall score higher than precision.

8.0: Qualitative Results

8.1: Correct predictions

Our model is effective on images with semi-transparent masks (Figure 8.1), on images where humans are wearing other types of coverings, such as sunglasses and a headscarf (Figure 8.2), and images with masks that look abnormal (Figure 8.3), including one example of a mask partially covered with a hand.

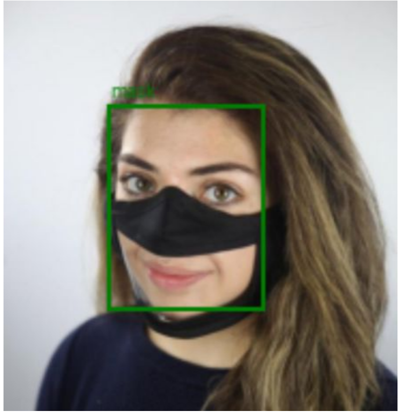


Figure 8.1: Output of model on image with semi-transparent mask.

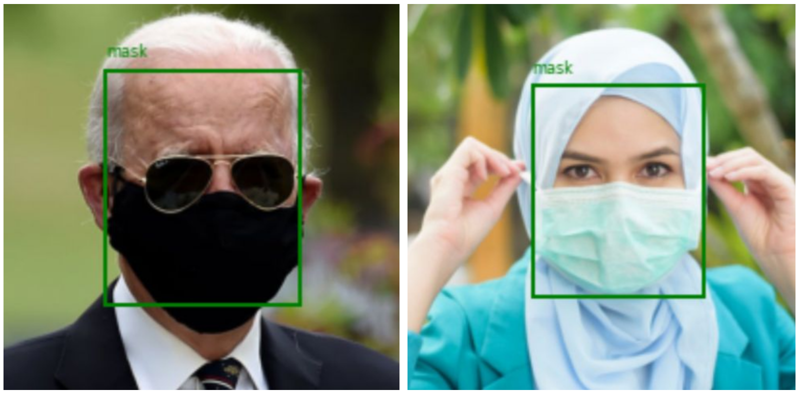


Figure 8.2: Output of model on images of people wearing non mask coverings with the mask, including sunglasses (left) and headscarf (right).

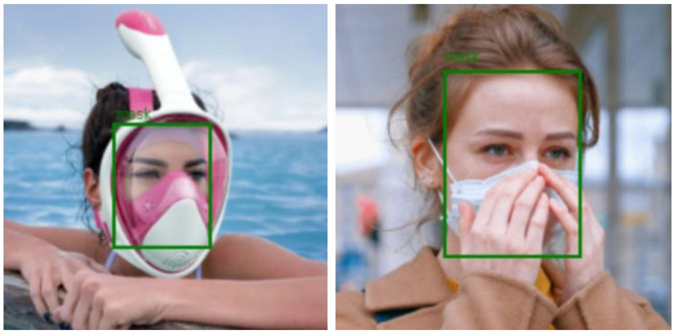


Figure 8.3: Output of model on images of people wearing unusual-looking masks.

8.2: Incorrect predictions

Our model did not perform well on images in three cases: where the mask contained writing or designs on it (Figure 8.4), where the face was partially obstructed from clear view (Figure 8.5a), and where the face was tilted away from a front-facing orientation (Figure 8.5b).

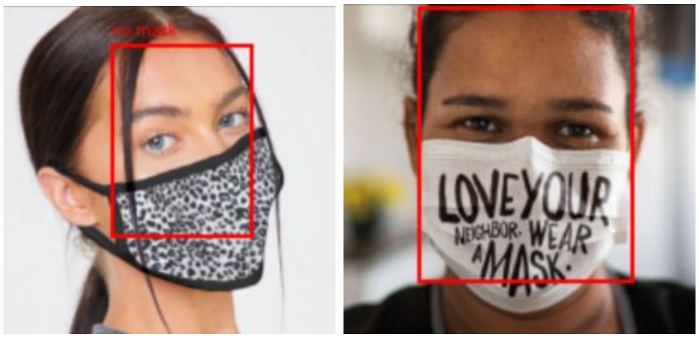
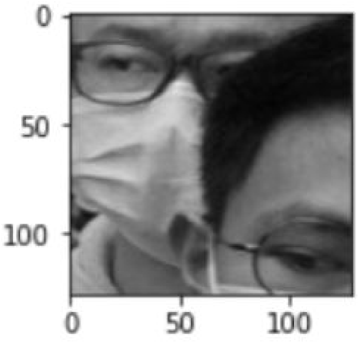


Figure 8.4: Output of model on images with masks having writing and designs.



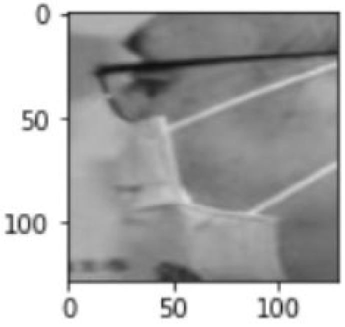


Figure 8.5a: Model predicted no mask on image with obstructions.

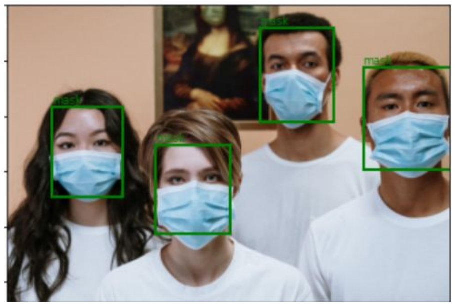
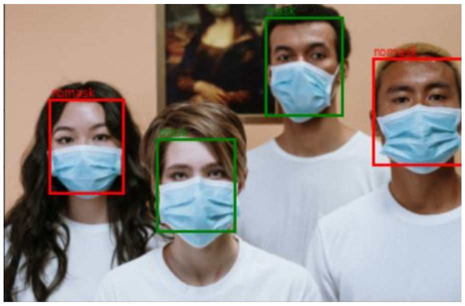
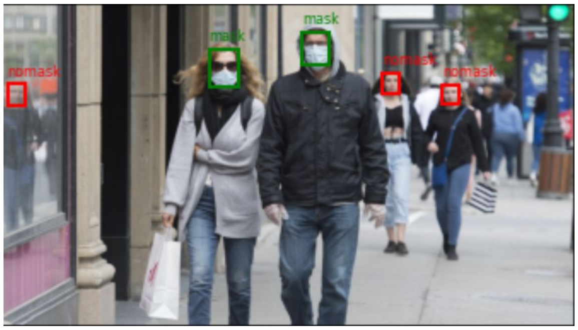
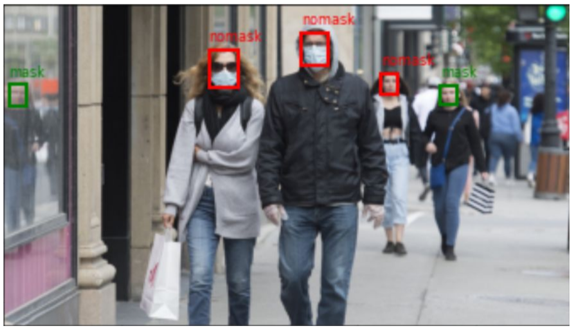
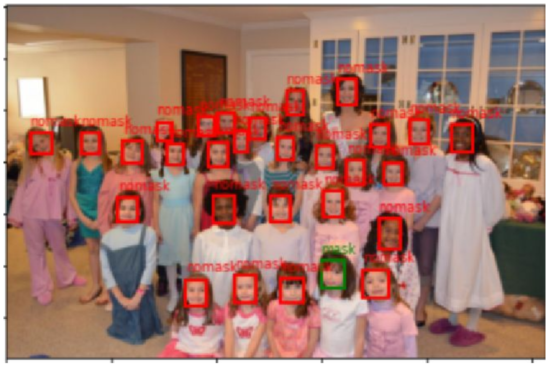
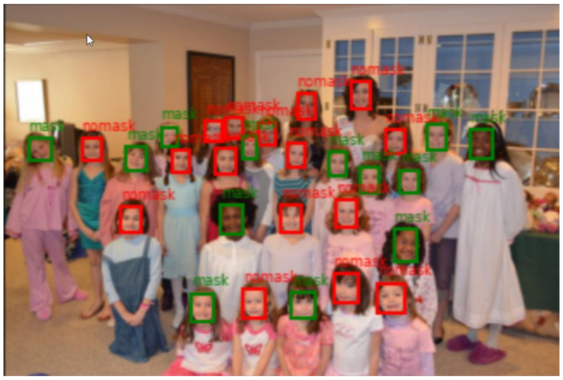
Fig 8.5b: Model predicted no mask on image with face titled from direct front-facing orientation.

9.0: Evaluate model on new data and Discussion

We took special efforts to evaluate our model on new data samples obtained from the internet.This provided us invaluable insight into making the final model choice. The following table compares the performance of the Google Net Transfer Learning Solution and the CNN Mask Net on the new data. CNN Mask Net performed consistently better than the Google Net solution.

The results were surprising because Quantitative results indicated that Google Net Transfer Learning Solution seemed to offer better performance on the test data . These are all images very similar to the training data set and we expected Google Net to perform well on these samples. However, our Mask Net-CNN seems to perform excellent as illustrated in these samples below in Table 9.1.

Table 9.1: Comparison of Mask Net and Google Net models performance

Mask Net	Google Net
	
	
	

As shown in Figure 9.1[13], we froze the convolution base of Google Net and used it as a feature extractor (Quadrant 4) and trained an ANN classifier on top of it. However, our dataset was large. Hence, the approach of training some layers in the convolution base and leaving others frozen would have provided a higher accuracy when using transfer learning (Quadrant 2). However, we were limited by the computational capabilities of Google Collaboratory and hence it took as unreasonably large amounts of time to train any layers from Google Net for our large dataset. Therefore, this approach could not be implemented.

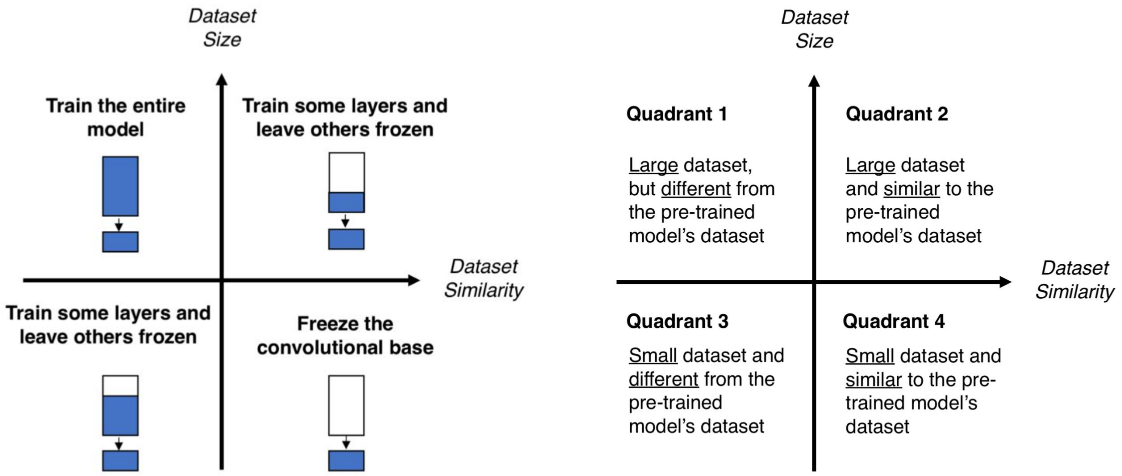

Dataset Size
Train the entire model
Train some layers and leave others frozen
Dataset Similarity
Train some layers and leave others frozen
Freeze the convolutional base
Dataset Size
Quadrant 1
Large dataset, but different from the pre-trained model's dataset
Quadrant 2
Large dataset and similar to the pre-trained model's dataset
Dataset Similarity
Quadrant 3
Small dataset and different from the pre-trained model's dataset
Quadrant 4
Small dataset and similar to the pre-trained model's dataset

Fig 9.1: Explanation for the intended feature extractor[13].

But the additional testing on new samples convinced us of the ability of our Mask Net-CNN to successfully classify images as mask or no mask. Thus the final model choice was our tuned CNN (Figure 9.2).

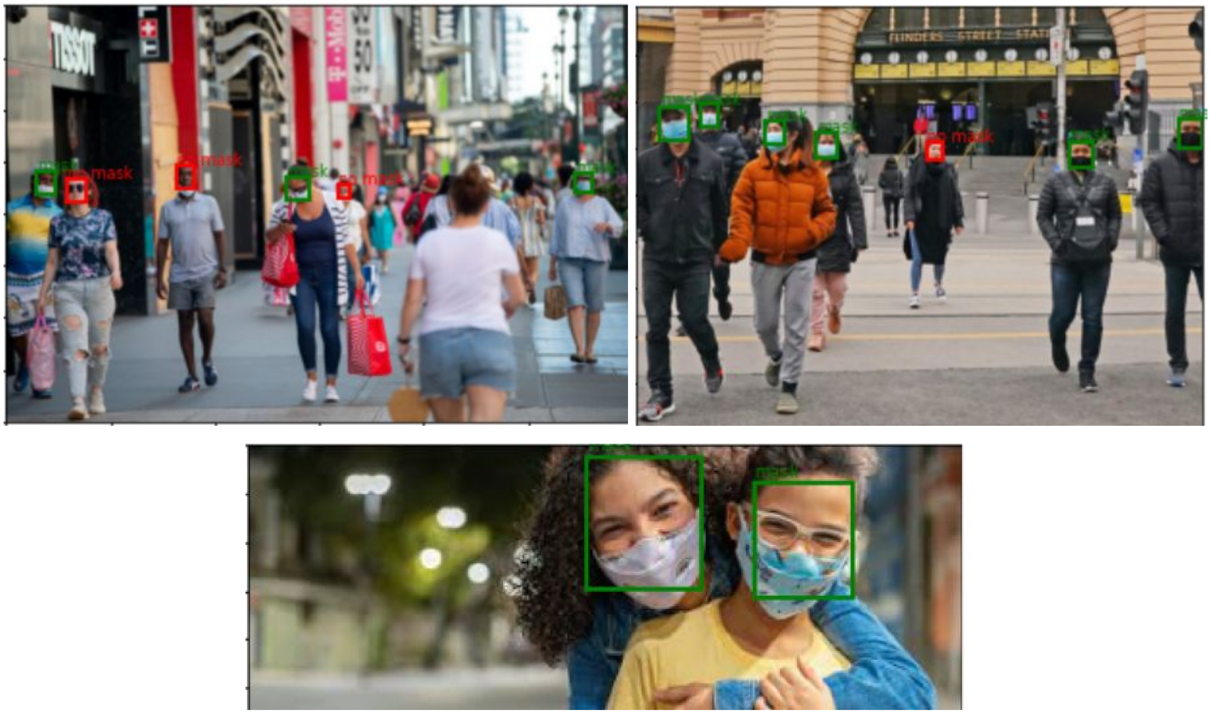


Figure 9.2: Sample outputs from Mask Net-CNN.

This strong performance observed is mostly likely due to the dataset we created by merging 3 datasets. We repurposed WiderFace dataset which was originally used for image detection and scene classification, to use it for face mask classification.We also merged two face mask datasets allowing for a very diverse set of humans and masks, therefore, the model was able to perform strongly on a variety of inputs.

Here are more examples which indicate the age detection outputs along with face mask classification (Figure 9.3). For images of children, the output indicates exempted because they are in the age group (6-11). The group of adults are not exempted because they are of high risk COVID group.

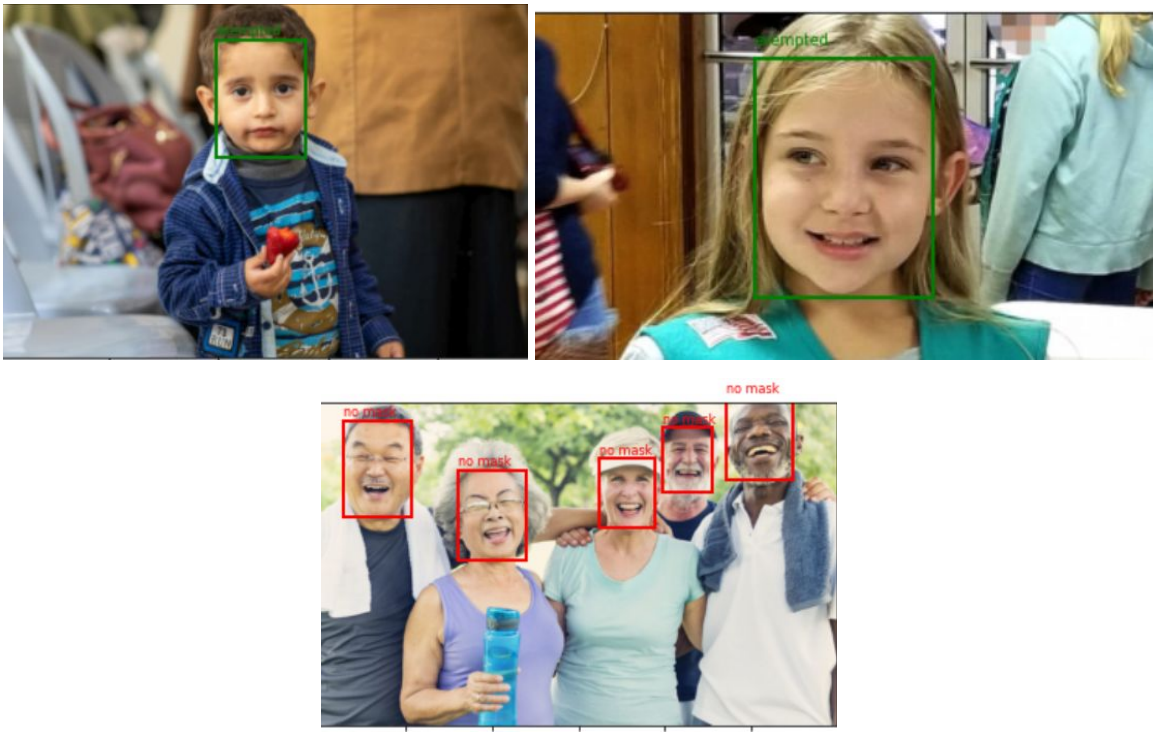


Fig 9.3: More examples with age detection and mask classification.

However, there were a few cases when our model misclassified (Figure 9.4). When faces are obstructed, or overlap each other, our model misclassifies those images. In addition, when the bounding box coordinates are inappropriately generated and the faces are not centered on the photograph, misclassifications are reported.

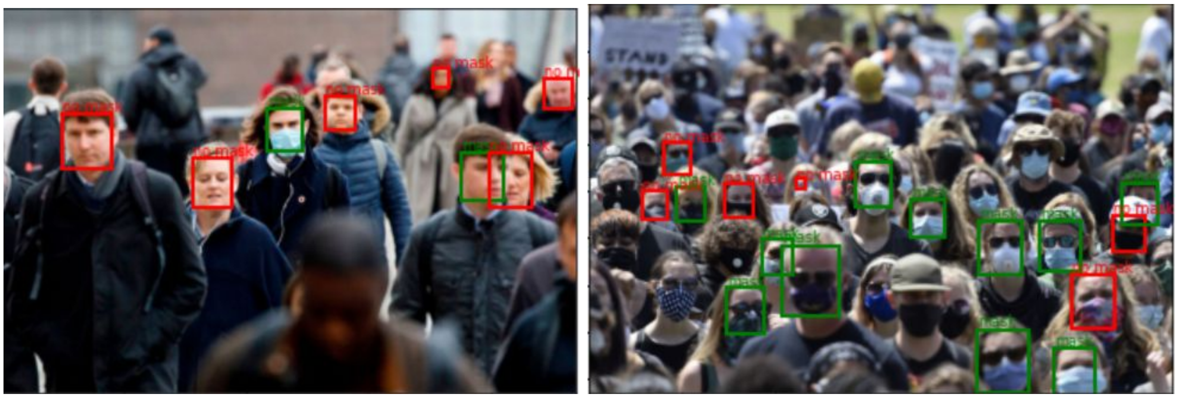


Fig 9.4: Example of misclassified images.

Another scenario when the model makes incorrect predictions is when the images have high amounts of blur or very poor resolution (Figure 9.5).

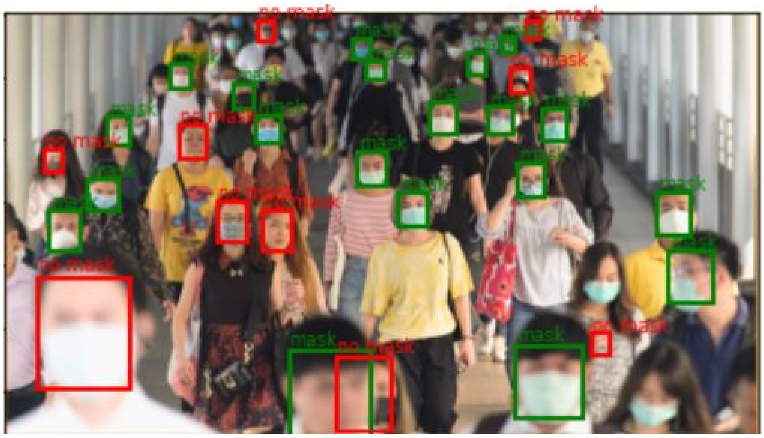


Fig 9.5: Examples of faces with high amounts of blur which leads to misclassification.

When people face away from the camera such that the percentage of the photograph occupied by the face is small, misclassifications are reported (Figure 9.6). All of these misclassifications can be explained by the fact that our training data consisted mostly of high resolution images with the blurry photos removed. In addition mostly all images in the training data set had perfectly cropped, photos with faces centered on the photograph hence this behaviour of the model is justified.

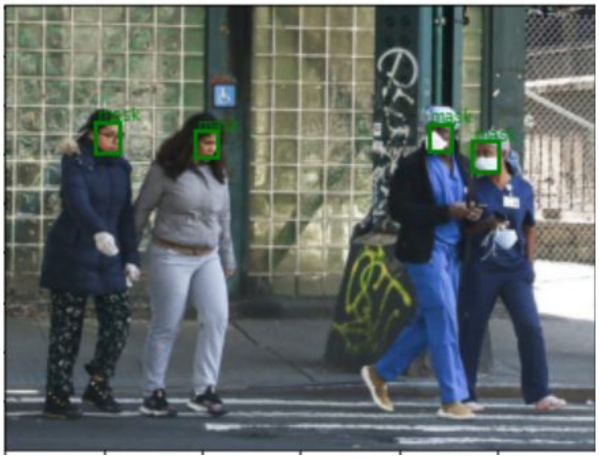


Fig 9.6: Example of faces facing away from the camera which leads to misclassification.

Lastly we noticed that the model misclassified more images of children than adults (Figure 9.7). This is because our training dataset had been created from an adult face mask dataset and the proportion of child photos in the dataset were comparatively lower.

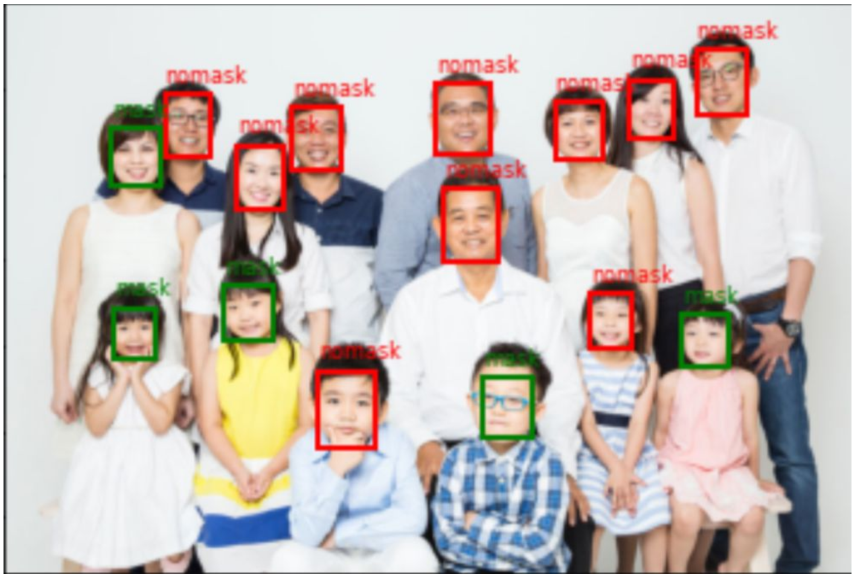


Fig 9.7: Family picture with adults and children. Higher number of children were misclassified.

If given more time and more computational power, we would incorporate more children photographs, photos with slight blur and photos where participants face away from the camera, in our training data so the model learns to classify those kinds of images. High blur or poor resolution in the photo results from the face detection algorithm cropping out the photos incorrectly which may be impacted by the angle and camera distance at which the photo has been captured. Thus, when this model is applied in a real-life scenario such as processing a live camera feed, we will consider the position of the camera and train our model on the images generated by capturing people's faces at those angles.

Connecting it back to what we learnt in lectures about drawbacks of CNN-CNN’s do not encode the information that maps position and orientation of images into their predictions. In addition CNN’s are spatially invariant to 3D space which may explain the misclassifications seen in our model.

10.0: Ethical Considerations

Some ethical considerations that were given priority by my team includes consent of images taken and also the discriminatory behaviour of our AI

10.1: Consent of Images Taken

Our dataset mainly deals with faces of people wearing masks. Thus, the images of the people should have been taken with their consent to protect their privacy[14].

We made sure all the datasets we have chosen for this project respects the Personal Data Protection Act [14].

10.2: Discriminatory Behaviour of the AI

The training of our AI is going to be made on images of people wearing masks. We will make sure the people behind the masks are of all genders and diverse racial backgrounds.

Our goal is to make the AI perform equally well on images of anyone wearing a mask.

11.0 Project Difficulty

We believe our project falls under the moderately difficult category. This section elaborates on the elements of the projects that we found to be challenging and time consuming

11.1: Detection of Multiple Faces

Our proposed face detection technology can work on multiple faces in a single image. We did this by using an existing face detection algorithm to separate all the faces in the image and individually pass it to our model for predictions.

11.2: Difficulty in Data Gathering

It was difficult to find a dataset of cropped faces of people wearing and not wearing masks. Most of the datasets we found were full sized images (uncropped images) that we cannot use to train the model. We had to develop XML parsing script and a basic face detection and cropping script to use the datasets we found. Further explanation on the characteristics of the datasets we found are discussed in Section 4.0.

11.3: Difficulty in Implementing Age Detection

To add further complexity to our project, we decided to implement age detection to further classify if the person not wearing a mask is below the required age to wear one.

The implementation is especially challenging when we have to deal with multiple faces in a single image. As shown in the flow chart in Section 2.0, each cropped image is first passed to our face detection model and only passed on to age detection if it was classified as not wearing a mask.

The most readily available age detection technology does not work on just the cropped face. So, to accommodate this, we had to work around and figure out a way to run age detection and mask detection simultaneously to achieve what we wanted to.

12.0 References

[1]M. Joshi, "APS360 Project Progress Report", 2020.

[2] M. Loey, G. Manogaran, M. H. N. Taha, and N. E. M. Khalifa, "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*, Nov. 2020. Accessed on: Nov. 29, 2020. [Online]. Available: https://doi.org/10.1016/j.scs.2020.102600

[3] Who.int. 2020. Coronavirus Disease (COVID-19): Children And Masks. [online] Available : https://www.who.int/news-room/q-a-detail/q-a-children-and-masks-related-to-covid-19 [Accessed 17 October 2020].

[4] Kulkarni, A., Vishwanath, A., Shah, C., Praveen, V., Wang, Y., Docca, A. and Dinea, C., 2020. Implementing A Real-Time, AI-Based, Face Mask Detector Application For COVID-19 | NVIDIA Developer Blog. [online] NVIDIA Developer Blog. Available: https://developer.nvidia.com/blog/implementing-a-real-time-ai-based-face-mask-detector-application-for-covid-19/ [Accessed 10 November 2020].

[5] "Detecting Masks in Dense Crowds in Real-time with AI and Computer Vision", Medium, 2020. [Online]. Available: https://towardsdatascience.com/detecting-masks-in-dense-crowds-in-real-time-with-ai-and-computer-vision-9e819eb9047e. [Accessed: 10- Nov- 2020].

[6]M. Joshi, R. Natarajan, S. Bhattacharya and S. Gafur, "APS360 Project Proposal Team 27", 2020.

[7] M. Loeya, G. Manogaran, M. H. N. TahadNour, and N. E. M. Khalifa, "A hybrid deep transfer learning model with machine learning methods for face mask detection in the era of the COVID-19 pandemic," Measurement, vol. 167, July. 2020. Accessed on Nov. 30, 2020. [Online] Available: https://doi.org/10.1016/j.measurement.2020.108288

[8] Kaggle.com. 2020. Face Mask ~12K Images Dataset. [online] Available: https://www.kaggle.com/ashishjangra27/face-mask-12k-images-dataset [Accessed 17 October 2020].

[9] Kaggle.com. 2020. Medical Masks Dataset Images Tfrecords. [online] Available: https://www.kaggle.com/ivandanilovich/medical-masks-dataset-images-tfrecords [Accessed 17 October 2020].

[10]"WIDER FACE: A Face Detection Benchmark", *Shuo Yang 1213.me*, 2020. [Online]. Available: http://shuoyang1213.me/WIDERFACE/. [Accessed: 09- Dec- 2020].

[11] A. Byun, F.-F. Liu, R. Krishna, and D. Xu, “Convolutional Neural Networks (CNNs / ConvNets),” CS231n Convolutional Neural Networks for Visual Recognition, Jan-2020. [Online]. Available: https://cs231n.github.io/convolutional-networks/. [Accessed: 2020]

[12] T. Yiu, "Understanding Random Forest." Towards Data Science. Updated June 12, 2019. [Article].Available: https://towardsdatascience.com/understanding-random-forest-58381e0602d2, Accessed on: Dec. 4, 2020.

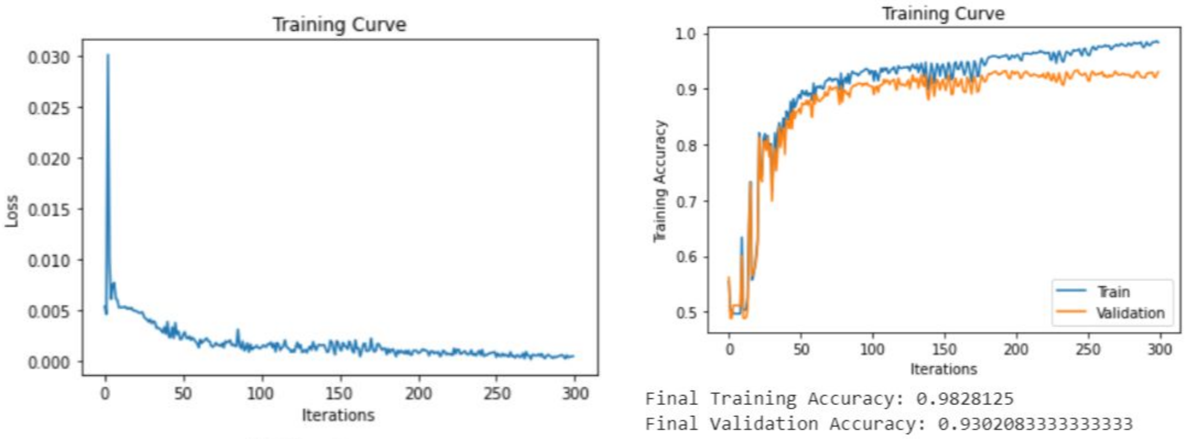
[13]"Transfer learning from pre-trained models", *Medium*, 2020. [Online]. Available: https://towardsdatascience.com/transfer-learning-from-pre-trained-models-f2393f124751. [Accessed: 09- Dec- 2020].

[14] Priv.gc.ca. 2020. The Personal Information Protection And Electronic Documents Act (PIPEDA) - Office Of The Privacy Commissioner Of Canada. [online] Available: https://www.priv.gc.ca/en/privacy-topics/privacy-laws-in-canada/the-personal-information-protection-and-electronic-documents-act-pipeda/[Accessed 17 October 2020].

13.0 Appendix

The scenario where the model was overfitting-after which data augmentation was applied

Trial No	Experiment Results and Numericals and Results	Observations
1.Mask Net-CNN	Model performance without having data augmentation applied produced the loss function (Figure 7.2a) and training and validation curves (Figure 7.2b) below.	Due to the absence of data augmentation, the model produced a very high accuracy. The final accuracy was as follows: ● Training - 98.3% ● Validation - 93.1% It seems the model is overfitting to the training data.



```
Final Training Accuracy: 0.9828125
Final Validation Accuracy: 0.9302083333333333
```

7.2a   7.2b

Figure 7.2a: Loss function on model without data augmentation

Figure 7.2b:Training and Validation curves on model without data augmentation