Face Mask Appropriateness Classifier

Introduction to Deep Learning Systems

Final Report

Sweta Bharti & Graham Jones

Introduction

The current situation of COVID-19 is a worldwide pandemic. It demands constant protection from it's spread through wearing a proper face mask appropriately. COVID safety rules define the following as appropriate and inappropriate ways of wearing a face mask.

Through college emails and on public transit we are constantly reminded of using a face mask appropriately. It's a prevailing situation starting in the year 2020 thus we have worked on creating a model to classify on face images of masks worn into appropriate or inappropriate classes of wearing a mask.

Being a current situation, we don't have the dataset prepared already to work on this classification. But the situation asks for coming up with the best we can to provide our input in prevention of spreading of the virus.

Thus our aim is to create artificial images of these two classes of wearing a face mask appropriately or inappropriately and do propensity matching with the minimal real images available to come up with a good accuracy model capable of classifying on real face images or videos.

Our model can be applied to CCTV video capturing technologies to automatically detect faces wearing masks inappropriately. This can be used as an auto warning generator at college spaces and public transits.

Background Work

COVID-19: Face Mask Detector:
https://www.pyimagesearch.com/2020/05/04/covid-19-face-mask-detector-with-opencv-keras-tensorflow-and-deep-learning/
Classifying Mask vs No Mask. We referenced this method for applying our own variations of putting masks on face images.

Incorrect FaceMask-Wearing Detection: https://www.mdpi.com/2227-9032/9/8/1050/pdf
Classification done on limited real images clicked manually.

SSDMNV2: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7775036/
Classifying Mask vs No Mask. We proportionally mixed Medical Dataset in appropriate mask class.

Some other mask/ no mask work background reports can be seen in the below image.

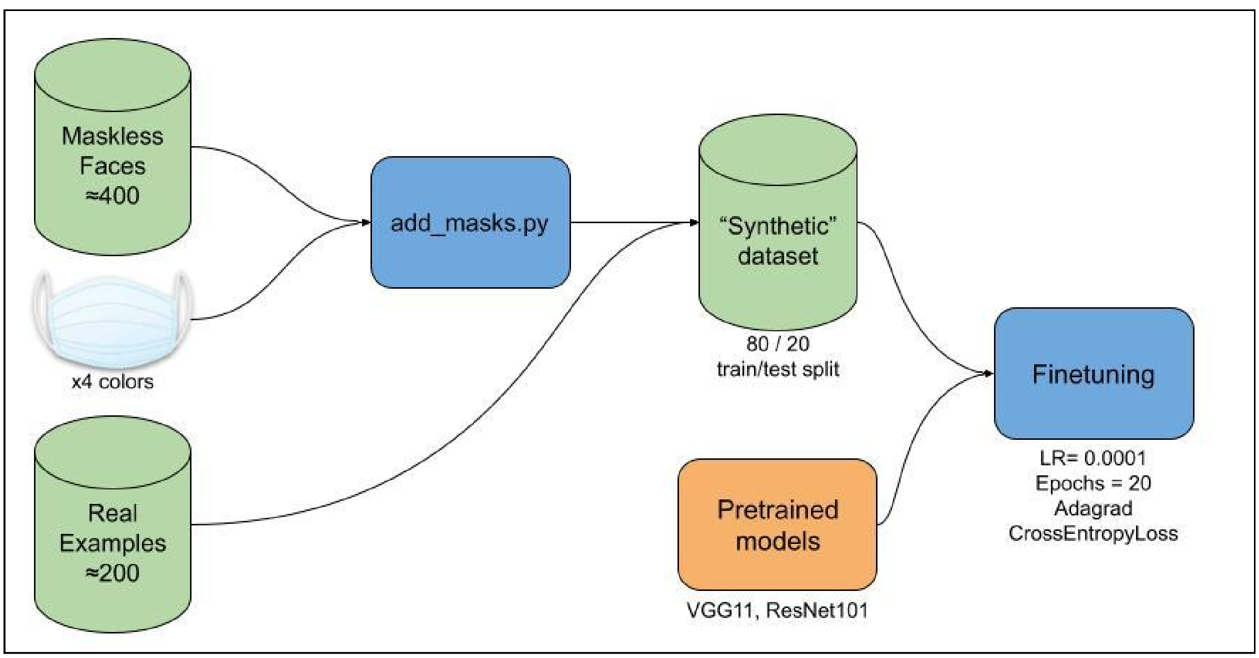
1st Author [ref]	Date	Type of Detection	Face Detector	Classification Model	Software Library	Best Accuracy
Nagrath [18]	March 2021	mask/ no mask	Single shot multibox	MobileNetV2	TensorFlow, OpenCV	92.64%
Mata [20]	April 2021	mask/ no mask	Image Data Generator	CNN	TensorFlow, OpenCV	60%
Jauhari [21]	March 2021	mask/ no mask	Cascade Viola Jones	AdaBoost	Python	90.9%
Sen [22]	February 2021	mask/ no mask	-	MobileNetV2	PyTorch, OpenCV	79.2%
Balaji [23]	2021	mask/ no mask	Viola-Jones detector	VGG-16 CNN	TensorFlow, OpenCV	96%
Kurlekar [24]	April 2021	mask/ no mask	-	CNN	TensorFlow, OpenCV, Caffe	-
Sakshi [25]	March 2021	mask/ no mask	-	MobileNetV2	TensorFlow, Keras	99%
Cheng [26]	2020	mask/ no mask	YOLO v3 -tiny	CNN + SVM	-	-
Loey [27]	January 2021	mask/ no mask	YOLO v3	Resnet50 + SVM	-	99.5%
Rudraraju [29]	September 2020	mask/ no mask/ nose out	Haar cascade classifier	MobileNet	OpenCV, Keras	90%
Wang [30]	January 2021	mask/ no mask/ nose out	Fast RCNN	InceptionV2	OpenCV, Matlab	91.1%
Hussain [31]	April 2021	mask/ no mask/ nose out	YOLO v3	VGG-16, MobileNetV2, InceptionV3, ResNet50	Keras	99.8%

Approach

Since this is a novel task the dataset for training the model did not already exist. Therefore, the first step of our approach is to generate a synthetic dataset of correctly/incorrectly worn masks from images of faces. This is done with a python program called add_mask.py. This loops through some source folder of pictures of unmasked faces and assigns each picture (with P = 0.33) to one of the three classes. It uses OpenCV to identify facial landmarks and then superimposes a scaled image onto the face. Pictures of different colors of masks were created in order to improve generalization.

At this point the synthetic data is ready. Real images of correctly and incorrectly masked were found on the web and manually added into their respective classes folders to provide the model with more diverse examples and prevent it from overfitting to the specific features of the mask generation program. From there the training was run from a Jupyter notebook named mytraining.ipynb. In this file pretrained models from PyTorch's library of models were downloaded. Training and validation was performed on an 80/20 split of the synthetic dataset. These models were finetuned for 20 epochs and the trained model was saved for further testing. After the models were trained they were tested on an additional test set consisting entirely of real images to determine more realistic accuracy results.

Once the trained models were available they could be run on real time webcam footage. Using OpenCV to grab frames from a stream of video data the model would predict the displayed class.



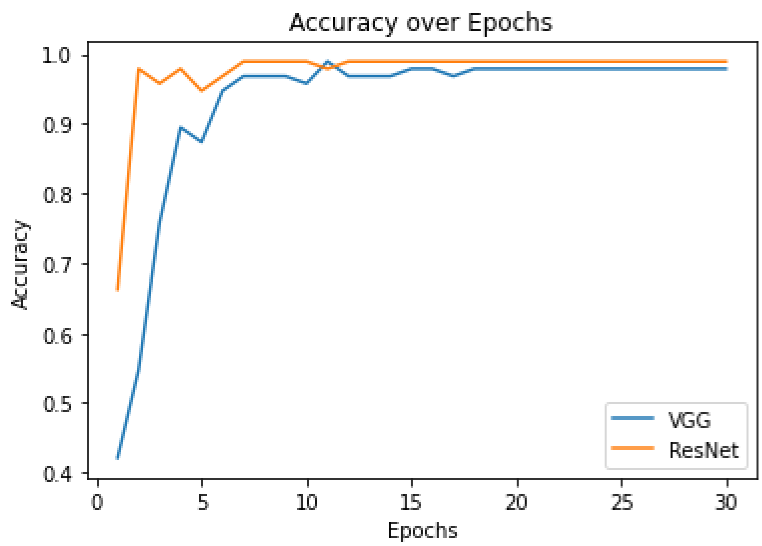
Overview of the architecture for this project

Fig: Synthetic examples of masks worn correctly

Synthetic examples of masks worn incorrectly

Running the model on webcam input

Results



This figure depicts the test accuracy over 30 epochs for both VGG11 and ResNet101

Accuracy Results on Entirely Real Dataset

Model	Mask Correct	Mask Incorrect	Without Mask	Overall Accuracy
ResNet	15/16	8/13	30/30	89.8%
VGG	14/16	11/13	29/30	91.5%

Inference Timing Results (ms/image)

Model	GPU	CPU	Size	Number of Params
ResNet	1.7	52.6	592MB	42M
VGG	0.6	38.2	617MB	128M

The VGG model is almost 3x faster when performing inferences on a GPU and therefore is likely a better candidate for deployment.

Conclusion

The accuracy results demonstrate that existing DL models can be trained quickly to classify incorrect mask wearing. The results also demonstrate that synthesized data can be useful in training the networks without needing to manually acquire as many “real” examples. The experiences with previously attempted synthetic datasets highlighted that the distribution of the synthetic data is important, data is king in machine learning because it exposes the network to additional novel information which it can use to better generalize. However when that data is being generated artificially there might not be additional information present. Despite training with a significantly larger dataset the generalization was poor.

References

Code is available at: https://github.com/GrahamJones95/Mask_Classifier
Data is available at:
https://drive.google.com/drive/folders/17Fd0uTag6hISmoblUixee6vrOx3GRX6t?usp=sharing