

Face Mask Detection System

Rajat Sachdeva, Sonam, Harshita Sharma

Abstract - The World Health Organization (WHO) reports suggest that the two main routes of transmission of the COVID-19 virus are respiratory droplets and physical contact. Wearing a medical mask is one of the prevention measures that can limit the spread of certain respiratory viral diseases, including COVID-19. Therefore, it is necessary to wear a mask properly at public places like supermarkets, shopping malls. In this paper we have proposed a Face Mask Detection system to check whether a person is wearing a mask properly or not. There are only a few researches on face mask detection in video streams. It consists of a two phase architecture for detecting face masks and will be using OpenCV, Tensorflow/Keras, MobileNetV2 and Deep Learning to achieve high accuracy.

Index Terms - OpenCV, MobileNetV2, Tensorflow, Keras, Deep Learning, COVID19, Face Mask

1. Introduction

The coronavirus disease 2019 has already infected more than 20 million people and caused over 750 thousand deaths globally according to the situation report 96 of the World Health Organization(WHO). There are many serious large scale respiratory diseases as well including Severe Acute Respiratory Syndrome(SARS) and Middle East Respiratory Syndrome(MERS). These diseases have increased a lot in the past few years. Therefore, people should be concerned about the health and respiratory diseases. And public health is the top priority of the government of all the countries. WHO also said that wearing face masks can help to stop coronavirus from spreading and the government made it necessary for all the people to wear face masks when going out at public places. And the people who are having respiratory symptoms should definitely wear face masks. Therefore, many service providers and supermarkets require customers to wear masks if they want to use their services. This is the reason that face mask detection system is necessary to help people but there are very few researches related to face mask detection.

Face mask detection is a system that detects whether a person is wearing a mask or not. It is same as an object detection system in which a system detects a particular class of objects. Through building this system we are trying to help ensure people's safety at public places. This system can be implemented in many areas such as supermarkets and shopping malls, schools, colleges, stations and so on. To accomplish this task, we'll be fine-tuning the MobileNet V2 architecture, a highly efficient architecture that can be applied to embedded devices with limited computational capacity

In this paper we have proposed a face mask detection system which is able to detect if a person is wearing a mask or not. It is a two phase system in which first we train our system and then apply the system to detect the faces. To achieve better results, we have used Keras/Tensorflow, MobileNetV2 and OpenCV and trained our model using a large dataset consisting of with and without mask faces. Deploying our face mask detector to embedded devices

could reduce the cost of manufacturing such face mask detection systems, hence why we choose to use this architecture.

2. Related Work

There are only few researches related to Face Mask Detection as it is related to a problem that has arrived recently. Here is the survey of some related work done in this field of study previously:

[1] This research study uses deep learning techniques in distinguishing facial recognition and recognize if the person is wearing a facemask or not. The dataset collected contains 25,000 images using 224x224 pixel resolution and achieved an accuracy rate of 96% as to the performance of the trained model. The system develops a Raspberry Pi-based real-time facemask recognition that alarms and captures the facial image if the person detected is not wearing a facemask.

[2] In this paper, a hybrid model using deep and classical machine learning for face mask detection was presented. The proposed model consists of two components. The first component is designed for feature extraction using Resnet50. While the second component is designed for the classification process of face masks using decision trees, Support Vector Machine (SVM), and ensemble algorithm. Three face masked datasets have been selected for investigation. The Three datasets are the Real-World Masked Face Dataset (RMFD), the Simulated Masked Face Dataset (SMFD), and the Labeled Faces in the Wild (LFW). The SVM classifier achieved 99.64% testing accuracy in RMFD. In SMFD, it achieved 99.49%, while in LFW, it achieved 100% testing accuracy.

[3] In this paper, the proposed model consists of two components. The first component is designed for the feature extraction process based on the ResNet-50 deep transfer learning model. While the second component is designed for the detection of medical face masks based on YOLO v2. Two medical face masks datasets have been combined in one dataset to be investigated through this research. The achieved results concluded that the adam optimizer achieved the highest average precision percentage of 81% as a detector.

3. Methodology

3.1 Tensorflow: TensorFlow is an open source framework developed by Google researchers to run machine learning, deep learning and other statistical and predictive analytics workloads. Like similar platforms, it's designed to streamline the process of developing and executing advanced analytics applications for users such as data scientists, statisticians and predictive modelers. Google has also used the framework for applications that include automatic email response generation, image classification and optical character recognition, as well as a drug-discovery application that the company worked on with researchers from Stanford University.

3.2 Keras: While deep neural networks are all the rage, has been a barrier to their use for developers new to machine learning. There have been several proposals for improved and simplified high-level APIs for building neural network models, all of which tend to look similar from a distance but show differences in closer examination. Keras (is one of the high level neural networks APIs) a deep learning API written in Python, running on top of the machine learning platform TensorFlow. It was developed with a focus on enabling fast experimentation. Being able to go from idea to result as fast as possible is key to doing good research. Keras contains numerous implementations of commonly used neural-network building blocks such as layers, objectives, activation functions, optimizers, and a host of tools to make working with image and text data easier to simplify the coding necessary for writing deep neural network code. The code is hosted on GitHub, and community support forums include the GitHub issues page, and a Slack channel.

3.3 OpenCV: OpenCV (Open source computer vision library) is a library of programming functions mainly aimed at real-time computer vision. The library is cross platform. OpenCV was built to provide a common infrastructure for computer vision and applications. The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, track moving objects, etc.

3.4 Deep Learning: Deep learning is an artificial intelligence (AI) function that imitates the workings of the human brain in processing data and creating patterns for use in decision making. Deep learning is a subset of machine learning in artificial intelligence that has networks capable of learning unsupervised from data that is unstructured or unlabeled. Also known as deep neural learning or deep neural network.

Deep learning methods aim at learning feature hierarchies with features from higher level of the hierarchy formed by the composition of lower level features. Automatically learning features at multiple levels of abstraction allow a

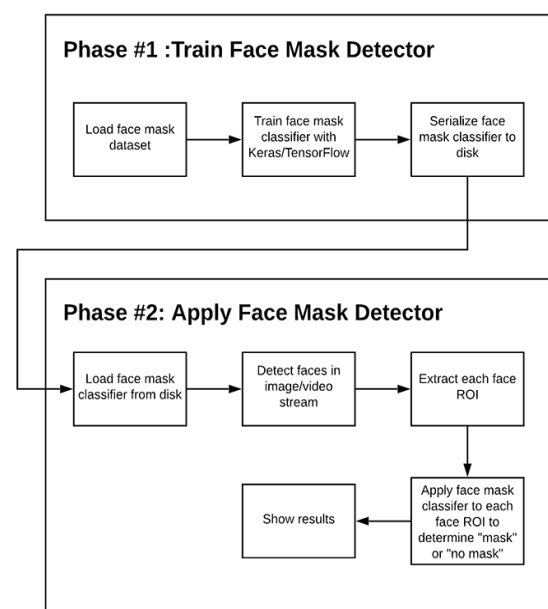
system to learn complex functions mapping the input to the output directly from data, without relying completely on human crafted features.

3.5 MobileNetV2: MobileNetV2s are small, low latency, low power models parameterised to meet the resource constraints of a variety of use cases. MobileNetV2 improves the state-of-the-art performance of mobile models on multiple tasks and benchmarks as well as across a spectrum of different model sizes. It is a very effective feature extractor for object detection and segmentation. It is used for depth-wise separable convolutions as efficient building blocks. MobileNetV2 has two main features: the first is Linear bottlenecks between the layers and the second is Shortcut connections between the bottlenecks.

The bottlenecks of the MobileNetV2 encode the intermediate input and outputs while the inner layer encapsulates the model's ability to transform from lower level concepts such as pixels to higher level descriptors such as image categories. It gives better accuracy.

3.6 Phases: In order to train a custom face mask detector, we broke our project in two distinct phases, each with its own respective sub-steps:

- **Training:** Here we have focused on loading our face mask detection dataset from disk, training a model (using Keras/TensorFlow) on this dataset, and then serializing the face mask detector to disk.
- **Deployment:** Once the face mask detector is trained, we then moved on to loading the mask detector, performing face detection, and then classifying each face as with_mask or without_mask.



3.7 Dataset: The dataset we have used in the project is created by Prajna Bhandary.

This dataset consists of 1,376 images belonging to two classes:

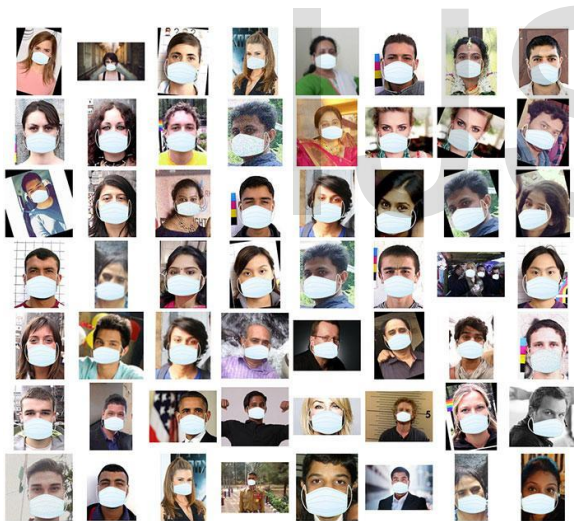
- with_mask: 690 images
- without_mask: 686 images

The goal is to train a custom deep learning model to detect whether a person is or is not wearing a mask.

To create this dataset, Prajna had the ingenious solution of:

- Taking normal images of faces
- Then creating a custom computer vision Python script to add face masks to them, thereby creating an artificial (but still real-world applicable) dataset

Facial landmarks are applied to face to build the dataset of people wearing masks. Facial landmarks allow us to automatically infer the location of facial structures, including eyes, eyebrows, nose, mouth, jawline.



3.8 Implementation: We trained the images of people not wearing a face mask. The next step is to apply face detection. Here we've used a deep learning method to perform face detection with OpenCV. Then the next step is to extract the face ROI with OpenCV and NumPy slicing. And from there, we apply facial landmarks, allowing us to localize the eyes, nose, mouth, etc.

Next, we need an image of a mask. This mask will be automatically applied to the face by using the facial landmarks. We will repeat this process for all the input images.

If you use a set of images to create an artificial dataset of people wearing masks, you cannot "reuse" the images

without masks in your training set.

The structure of project is like this

```
$ Face-Mask-Detection
├── dataset
│   ├── with_mask [690 entries]
│   └── without_mask [686 entries]
├── face_detector
│   ├── deploy.prototxt
│   └── res10_300x300_ssd_iter_140000.caffemodel
├── detect_mask_video.py
├── mask_detector.model
├── plot.png
└── train_mask_detector.py
2 directories, 8 files
```

The set of tensorflow.keras imports allow for:

- Data augmentation
- Loading the MobilNetV2 classifier (we will fine-tune this model with pre-trained ImageNet weights)
- Building a new fully-connected (FC) head
- Pre-processing
- Loading image data

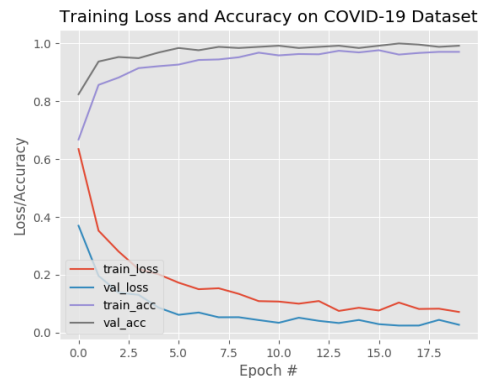
We have used scikit-learn (sklearn) for binarizing class labels, segmenting our dataset, and printing a classification report.

We prepared MobileNetV2 for fine tuning. Fine-tuning is a strategy I nearly always recommend to establish a baseline model while saving considerable time. Fine-tuning setup is a three-step process:

- Load MobileNet with pre-trained ImageNet weights, leaving off head of network
- Construct a new FC head, and append it to the base in place of the old head
- Freeze the base layers of the network. The weights of these base layers will not be updated during the process of backpropagation, whereas the head layer weights will be tuned.

During training, we have applied on-the-fly mutations to our images in an effort to improve generalization. This is known as data augmentation, where the random rotation, zoom, shear, shift, and flip parameters are established.

The imutils paths implementation helped us to find and list images in our dataset. And we have used matplotlib to plot our training curves.



Face mask detector training accuracy/loss curves demonstrate high accuracy and little signs of overfitting on the data. We're now ready to apply our knowledge of computer vision and deep learning using Python, OpenCV, and TensorFlow/Keras to perform face mask detection.

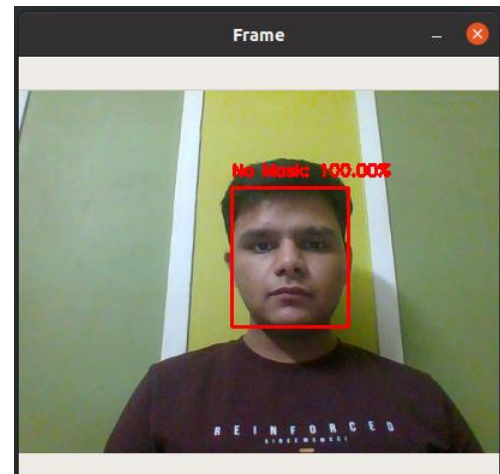
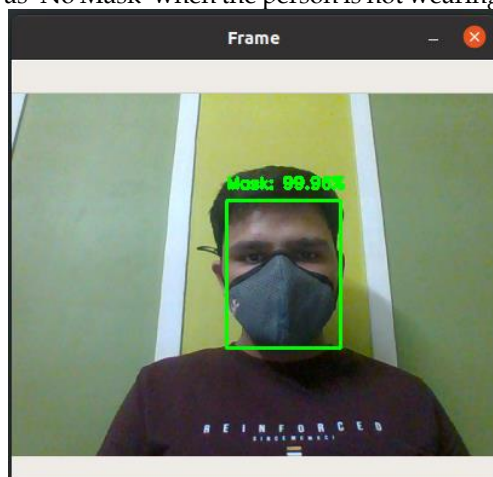
4. Results

```
[INFO] evaluating network...
```

	precision	recall	f1-score	support
with_mask	0.99	1.00	0.99	383
without_mask	1.00	0.99	0.99	384
accuracy			0.99	767
macro avg	0.99	0.99	0.99	767
weighted avg	0.99	0.99	0.99	767

We obtained ~99% accuracy on our test set.

We have successfully trained our model and tested it on a real time face using the laptop's camera. Our face mask detector correctly labeled the person's face as either 'Mask' or 'No Mask'. As you can see in this image that the face is labeled as 'Mask' when the person is wearing a mask and labeled as 'No Mask' when the person is not wearing a mask.



5. Conclusion

In this paper, a face mask detection system was presented which was able to detect face masks. The architecture of the system consists of Keras/Tensorflow and OpenCV. A novel approach was presented which gave us high accuracy. The dataset that was used for training the model consists of more than 3800 face images. The model was tested with real-time video streams. The training accuracy of the model was around 99%. This model is ready to use in real world use cases and to implement in CCTV and other devices as well. Deploying our face mask detector to embedded devices could reduce the cost of manufacturing such face mask detection systems, hence why we choose to use this architecture.

6. References

- [1] Militante, Sammy V., and Nanette V. Dionisio. "Real-Time Facemask Recognition with Alarm System using Deep Learning." 2020 11th IEEE Control and System Graduate Research Colloquium (ICSGRC). IEEE, 2020.
- [2] Loey, Mohamed, et al. "A hybrid deep transfer learning model with machine learning methods for face mask detection in the era of the COVID-19 pandemic." Measurement 167 (2020): 108288.
- [3] Loey, Mohamed, et al. "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 (2020): 102600.
- [4] Chowdary, G. Jignesh, et al. "Face Mask Detection using Transfer Learning of InceptionV3." arXiv preprint arXiv:2009.08369 (2020).

- [5] Loey, Mohamed, et al. "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* (2020): 102600.
- [6] Sandler, Mark, et al. "Mobilenetv2: Inverted residuals and linear bottlenecks." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018.
- [8] Viola, Paul, and Michael J. Jones. "Robust real-time face detection." *International journal of computer vision* 57.2 (2004): 137-154.
- [9] Yuan, Liping, et al. "A convolutional neural network based on TensorFlow for face recognition." *2017 IEEE 2nd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*. IEEE, 2017.
- [10] Kumar, Ashu, Amandeep Kaur, and Munish Kumar. "Face detection techniques: a review." *Artificial Intelligence Review* 52.2 (2019): 927-948.
- [11] Cheung, Brian. "Convolutional neural networks applied to human face classification." *2012 11th International Conference on Machine Learning and Applications*. Vol. 2. IEEE, 2012.
- [12] Vani, A. K., et al. "Using the Keras Model for Accurate and Rapid Gender Identification through Detection of Facial Features." *2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC)*. IEEE, 2020.
- [13] Yadav, Shashi. "Deep Learning based Safe Social Distancing and Face Mask Detection in Public Areas for COVID-19 Safety Guidelines Adherence." *International Journal for Research in Applied Science and Engineering Technology* 8.7 (2020): 1368-1375.
- [14] Das, Arjya, Mohammad Wasif Ansari, and Rohini Basak. "Covid-19 Face Mask Detection Using TensorFlow, Keras and OpenCV."
- [15] Liu, Xiaopeng, and Sisen Zhang. "COVID- 19: Face masks and human- to- human transmission." *Influenza and Other Respiratory Viruses* (2020).
- [16] Cheng, Vincent CC, et al. "The role of community-wide wearing of face mask for control of coronavirus disease 2019 (COVID-19) epidemic due to SARS-CoV-2." *Journal of Infection* (2020).
- [7] Saxen, Frerk, et al. "Face Attribute Detection with MobileNetV2 and NasNet-Mobile." *2019 11th International Symposium on Image and Signal Processing and Analysis (ISPA)*. IEEE, 2019.
- [17] Wong, Sunny H., et al. "COVID-19 and public interest in face mask use." *American Journal of Respiratory and Critical Care Medicine* 202.3 (2020): 453-455.
- [18] Greenhalgh, Trisha, et al. "Face masks for the public during the covid-19 crisis." *Bmj* 369 (2020).
- [19] Inamdar, Madhura, and Ninad Mehendale. "Real-Time Face Mask Identification Using Facemasknet Deep Learning Network." Available at SSRN 3663305 (2020).
- [20] Rowley, Henry A., Shumeet Baluja, and Takeo Kanade. "Neural network-based face detection." *IEEE Transactions on pattern analysis and machine intelligence* 20.1 (1998): 23-38.