Classification of Masked and Unmasked Faces using Transfer Learning Concept

Memo Karpa
Registration number: [redacted]
Roll Number: [redacted]
Anweshan Roy Chowdhury
Registration number: [redacted]
Roll Number: [redacted]
Sayon Roy
Registration number: [redacted]
Roll Number: [redacted]
Subhajit Majumdar
Registration number: [redacted]
Roll Number: [redacted]

Supervised by
Prof. Antika Sinha and Dr. Samir Malakar
Assistant Professors,
Department of Computer Science,
Asutosh College, Kolkata, India

Final Project Report submitted to the Department of Computer Science, Asutosh College, University of Calcutta in partial fulfilment of the requirements for the Degree of

B. Sc. (Hons.) in COMPUTER SCIENCE

Certificate from Supervisors

This is to certify that the project topic entitled "Classification of Masked and Unmasked Faces using Transfer Learning Concept" has been solely prepared and submitted by Memo Karpa(Roll No.: [redacted], Reg. No.: [redacted]), Anweshan Roy Chowdhury(Roll No.: [redacted], Reg. No.: [redacted]), Sayon Roy(Roll No.: [redacted], Reg. No.: [redacted]) and Subhajit Majumdar(Roll No.: [redacted], Reg. No.: [redacted]), under supervision of Prof. Antika Sinha and Dr. Samir Malakar during the year 2021 in fulfilment of the Computer Science project as a part of final Semester VI examinations as conducted by the University of Calcutta.

(PROF. ANTIKA SINHA)
Assistant Professor,
Department of Computer Science,
Asutosh College, Kolkata-700026

(DR. SAMIR MALAKAR)
Assistant Professor,
Department of Computer Science,
Asutosh College, Kolkata-700026

Acknowledgement

First and foremost, praises and thanks to God, the Almighty, for His showers of blessings throughout our project work for enabling us to complete the project successfully.

We would like to express our deep and sincere gratitude to our project supervisors, Prof. Antika Sinha, Assistant Professor, Department of Computer Science, Asutosh College and Dr. Samir Malakar, Assistant Professor and Head, Department of Computer Science, Asutosh College for giving us the opportunity to do research oriented project work and providing invaluable guidance throughout this project study.

Our Special Thanks to Prof. Gautam Mahapatra, Associate Professor, Department of Computer Science, Asutosh College, University of Calcutta, Kolkata for keenly introducing us to “Introduction to Computational Intelligence”.

We thank the management of the Department of Computer Science, Asutosh College, University of Calcutta, Kolkata for their support to do this work. Finally, we thank all the people who have supported us to complete the research work directly or indirectly.

We are extremely grateful to our parents for their love, prayers, caring and sacrifices for educating and preparing us for our future.

Memo Karpa
Anweshan Roy Chowdhury
Sayon Roy
Subhajit Majumdar

Table of Contents

Abstract

The whole world has gone through a lot and has faced huge challenges for survival in this era of COVID-19 scenario. Although face-masks are now found in each and every household, some people still refuse to abide by the guidelines. To bring these kinds of ignorant individuals under strict inspection, our research study helps in detecting whether a person has a mask on their face or not. Deep learning has equipped us with techniques such as facial recognition and image classification which we use in the process of masked face identification. This system can greatly aid in combating the spread of the virus by raising a relevant alarm whenever an unmasked (or not properly masked) person is detected.

We have prepared a model from a pretrained model with some fully connected layers to classify the images in a dataset consisting of both classes of masked and unmasked faces. With this we have trained a model which at the end, is capable of identifying and distinguishing people wearing (or not wearing) a mask in real time.

We finally found that the overall accuracy at an average for MobileNetV2 at 99%, average for ResNet50 at 97%, and DenseNet201 at 98%. The maximum among them was MobileNetV2, which we implemented through our live feed.In this scenario, when masks are mandatory to enter any place, we are presenting a system that helps the society to identify whether a person has a mask or not thus helping to promote safety.

1 Introduction

With the onset of the *Covid-19* pandemic, the world has gone into a complete lockdown. The Severe Acute Respiratory Syndrome novel Coronavirus - 2, also known as "*SARS nCoV-2*", has recorded more than 200 million cases while causing deaths in excess of 4 million[1]. Besides this, it has affected the lives of millions more by causing the fall in gross domestic product (GDP) [1] worldwide, thus leading to the rise in unemployment. It has further led to the educational institutions to close down for an indefinite period of time, thus affecting the education sector as well [2]. The virus is transmitted via the respiratory route, when people inhale droplets and inhale particles that are released by the infected people, when they talk, cough or sneeze [3].

To control the spread of this virus, the World Health Organisation has advised a number of measures to be taken by the general public which includes social distancing, hand sanitizers and face masks, apart from vaccinations[2]. According to studies all of these ways are effective in reducing the spread of the virus [4]. One of the most effective, easiest and cheapest ways to prevent further spread is by wearing a surgical or cloth face mask [5]. However, in day-to-day life, the use of face masks by the general public has gradually gone down leading to the rise of the number of Covid-19 cases during the second wave [6]. As such, governments around the world have released regulations on wearing face masks to stop the second wave of the virus from leading to a third and stronger wave[3,4].

[1] Covid-19 Stats: https://www.worldometers.info/coronavirus
[2] WHO Advisory: https://www.who.int/emergencies/diseases/novel-coronavirus-2019/advice-for-public
[3] Indian Govt.: https://www.mohfw.gov.in/pdf/GuidelinesonrationaluseofPersonalProtectiveEquipment.pdf
[4] USA Govt.: https://www.fda.gov/regulatory-information/search-fda-guidance-documents/enforcement-policy-face-masks-and-respirators-during-coronavirus-disease-covid-19-public-health

In today's ever progressing technological world deep learning has applications in numerous fields [7]. Deep learning (also known as deep structured learning) is part of a broader family of machine learning methods based on artificial neural networks with representation learning [8]. Most modern deep learning models are based on artificial neural networks as in deep learning networks (*DNN*) [9]. DNN provides deep learning algorithms like object detection, image classification, and image segmentation [10].

Convolutional neural network (CNN, or ConvNet) is the mostly used *multilayered* (like convolution layers, pooling layers, and fully connected layers) neural network. The CNN is a special architecture of ANNs, proposed by Yann LeCun in 1988. The CNN is designed to automatically and adaptively learn spatial hierarchies of features through a backpropagation algorithm which helps to model other useful algorithms [11]. It is linked with tasks involving computer vision as it can identify, as well as classify facial features precisely, storing image pattern details after the model has been trained and tested [12].

In this project we are applying deep learning algorithms to classify mask-wearing and non mask-wearing faces. The classifier that we have constructed collects images of people's faces, from a live stream, if required, to distinguish them according to classes *"Mask"*[5] and *"No Mask"*[6]. We have already learnt how deep learning and artificial neural networks can be beneficial to extract features from unprocessed data. Thus, in this study, to design the face-mask classifier, we have made use of a convolutional neural network. The project is implemented with the help of *Python* programming language, using the *Keras* library from the *TensorFlow* API. The detection was performed using the *OpenCV*, *Pillow* library in addition to *Javascript* (through *Flask Library*) for livestream.

[5] Mask: A mask is an object normally worn on the face (at least covering the nose to under the mouth), typically for protection (here from Covid-19).
[6] No Mask: The term is used in the context when a person is not wearing a mask at all or properly. The term is interchangeable with "Unmask".

2 Related Works

Convolutional neural networks are being widely used in the implementation of machine learning techniques like natural language processing- handwriting [13] and speech recognition [14], stock predictions [15], and churn prediction [16] among others. Other standard CNN architectures like AlexNet and VGGNet [17] contain packed convolutional layers and are being used extensively for image classification. Even in the field of facial recognition, a combination of CNNs and Region-based Convolutional Neural Networks (or R-CNNs) are being used, which have relatively high accuracy as compared to traditional CNN modules.

Apart from structures like MobileNetV2, DenseNet201 and ResNet50 that we have used, other faster models like YoloV3 [18], YoloV5 [19] [20] and InceptionV3 [21] are also available for faster face and facemask recognition [22] [23]. Inplace of Keras and TensorFlow, the model can also be included by other modules like PyTorch [24]. This detection can be put to use and build an alarm system with the help of an Arduino Module or Raspberry Pi [25]. The use of the detection can be further extended to sense social distancing in crowded places to control the spread of this Covid-19 virus [26].

3 Methodology

The working of the artificial neural network (or ANN) is similar to that of the human brain incorporating mathematical functions through the dense networks of neurons and axons or connective networks. The CNN is derived from these ANNs, and is highly used in the field of Artificial Intelligence. One of the primary uses of the CNN, is to recognize images and processing pixel data specifically, because of CNN's high accuracy. `Figure 1` below gives us a glimpse of what the pictures used by the deep learning processes for the implementation look like.

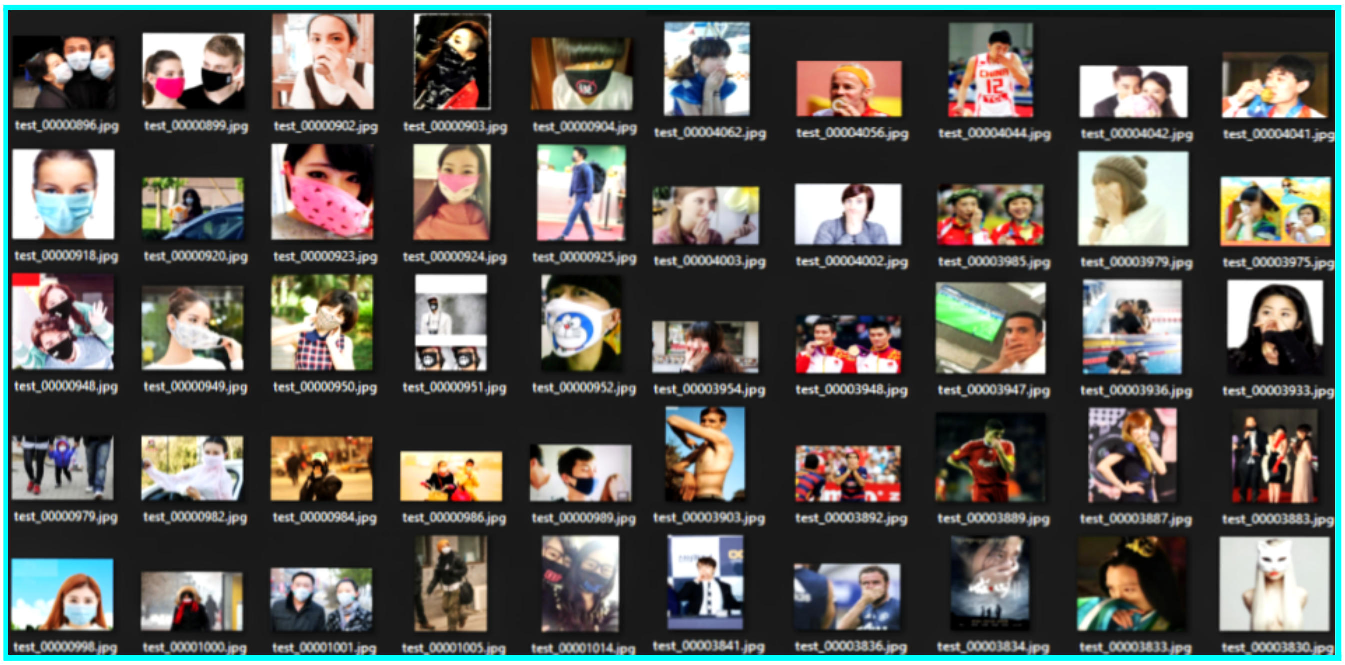


Figure 1: Example of Masked and Unmasked Face Images

The CNN follows a hierarchical model which works on building a network, like a funnel, and finally gives out a fully-connected layer where all the neurons are connected to each other and the output is processed. It finds its main application in the domains of object recognition, image classification, facial recognition, etc. The components of CNN as used in our model are as given below in `Figure 2`. This may vary from project to project according to its network design.

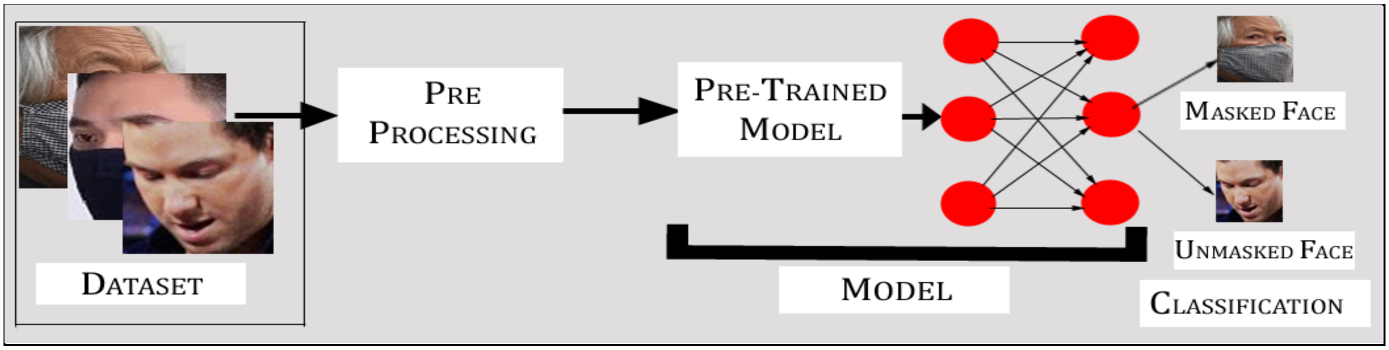


Figure 2: Face Mask Recognition Framework

3.1 Data Preparation (Image Acquisition and Pre-processing)

The foremost step of the current project is image acquisition. High quality images consisting of people with and without face masks were available from various published works. These were made accessible along with the annotation records in the link:

https://drive.google.com/drive/folders/1pAxEBmfYLoVtZ01BT3doxmesAO7n3ES1

The available annotation records of the file were processed which included the bounded region of the faces along with a binary classification (0/1). 0 and 1 stand for unmasked and masked images respectively. The obtained images are made available locally from the cloud server. The images were subjected to classification based on their annotated classes. Each face in every image was labelled with carefully prepared bounding box data found in their respective annotation records. These images were subjected to segmentation, based on the attached annotated records after which they finally became the basis of our model training dataset.

3.2 Extracting Features

For extracting features from images, we have followed transfer learning protocol. For this, we have used three well-known CNN models, namely MobileNetV2[7], DenseNet201[8], ResNet50[9] trained on Imagenet dataset. Imagenet dataset is a large dataset having 1000 classes. The pre-trained weights are directly used to extract features. Please note that we have not fine tuned the models. All these models are briefly described below.

[7] MobileNetV2: https://arxiv.org/abs/1801.04381
[8] DenseNet201: https://arxiv.org/abs/1608.06993
[9] ResNet50: https://arxiv.org/abs/1512.03385v1

3.2.1 MobileNetV2

MobileNetV2 is a convolutional neural network architecture that seeks to perform well on mobile devices. It is based on an inverted residual structure where the residual connections are between the bottleneck layers. The intermediate expansion layer uses lightweight depthwise convolutions to filter features as a source of non-linearity. As a whole, the architecture of MobileNetV2 contains the initial fully convolution layer with 32 filters, followed by 19 residual bottleneck layers. The MobileNetV2 architecture initialization is given below:

```
tf.keras.applications.MobileNetV2
(
    input_shape=None, alpha=1.0,
    include_top=True, weights="imagenet",
    input_tensor=None, pooling=None,
    classes=1000, classifier_activation="softmax",
    **kwargs
) → keras.Model
```

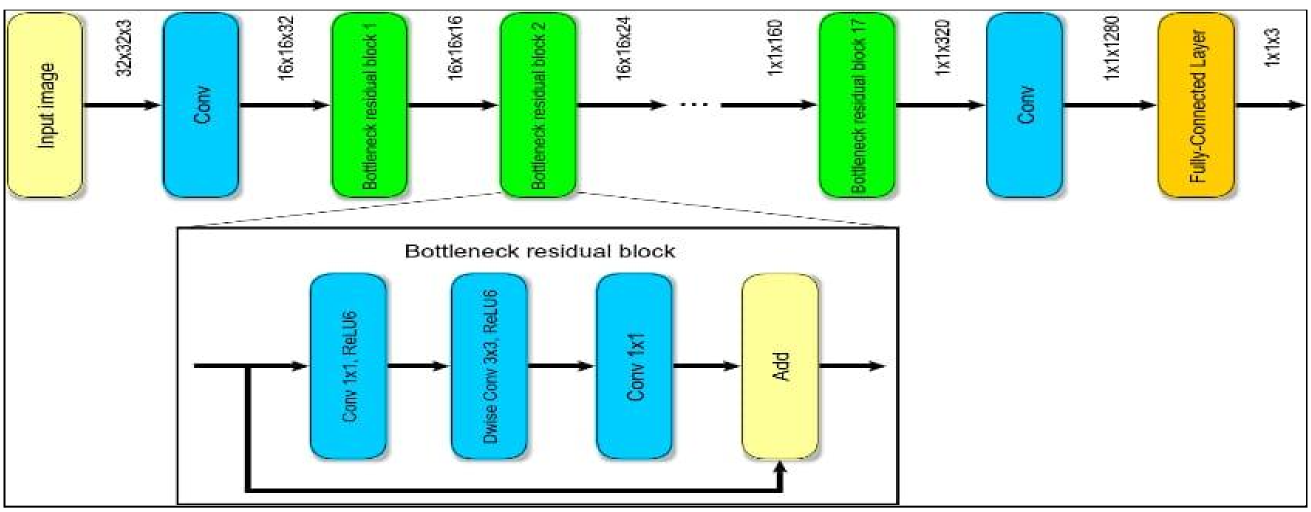


Figure 3: MobileNetV2 CNN Structure

MobileNetV2 is very similar to the original MobileNet, except that it uses inverted residual blocks with bottlenecking features. It has a drastically lower parameter count than the original MobileNet. MobileNets support any input size greater than 32 x 32, with larger image sizes offering better performance.

3.2.2 DenseNet201

DenseNet201 is a convolutional neural network that is 201 layers deep. One can load a version of the network trained on more than a million images from the ImageNet database. The pretrained network can classify images into 1000 object categories, such as keyboard, mouse, pencil, and many animals. As a result, the network has learned rich feature representations for a wide range of images. The network has an image input size of 224-by-224. The DenseNet201 architecture for initialization is given below:

```
tf.keras.applications.densenet.DenseNet201
(
    include_top=True, weights='imagenet',
    input_tensor=None, input_shape=None,
    pooling=None, classes=1000
) → keras.Model
```

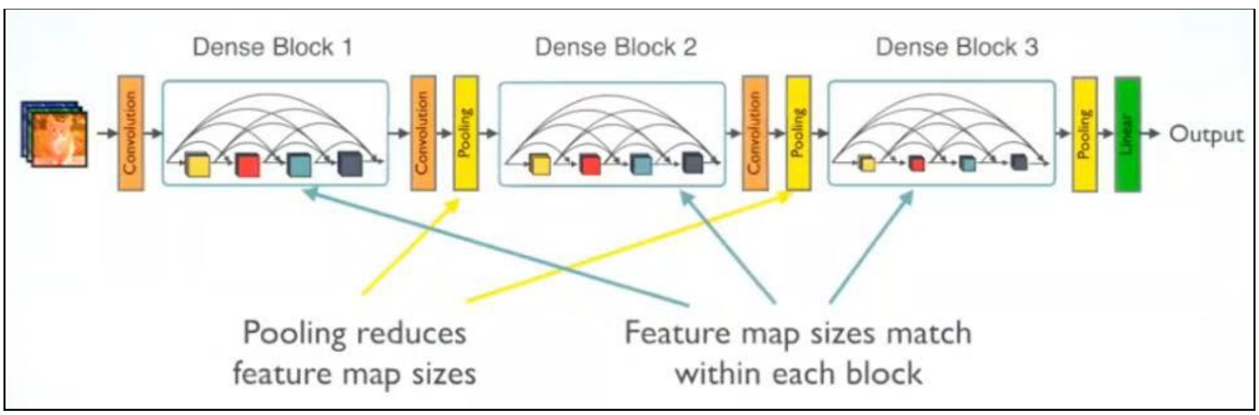


Figure 4: DenseNet201 CNN Structure

3.2.3 ResNet50

ResNet50 is a convolutional neural network that is 50 layers deep. One can load a version of the network trained on more than a million images from the ImageNet database. The pretrained network can classify images into 1000 object categories, such as keyboard, mouse, pencil, and many animals. As a result, the network has learned rich feature representations for a wide range of images. The network has an image input size of 224-by-224. The ResNet50 architecture for initialization is given below:

```
tf.keras.applications.resnet50.ResNet50
(
     include_top=True, weights='imagenet',
     input_tensor=None, input_shape=None,
     pooling=None, classes=1000, **kwargs
 ) → keras.Model
```

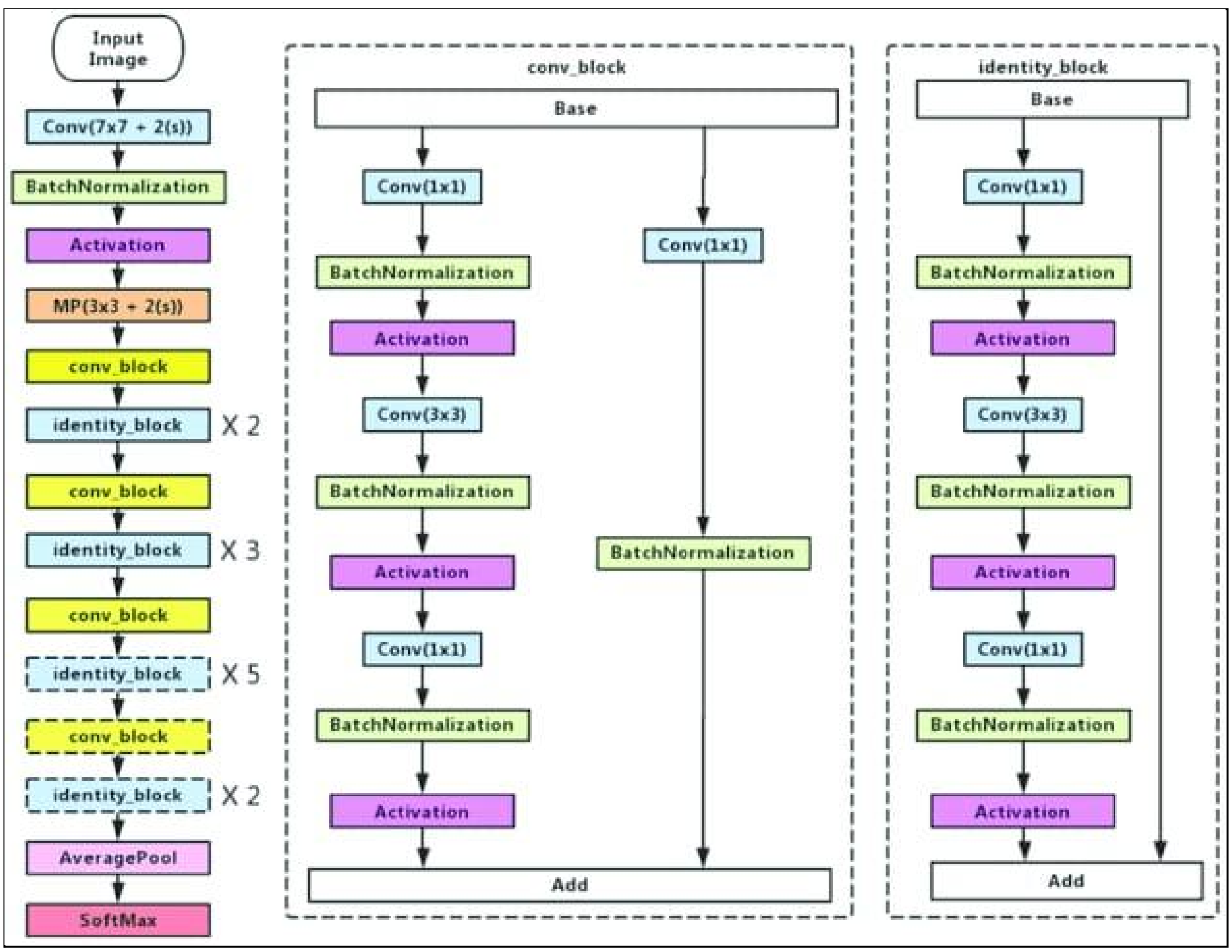


Figure 5: ResNet50 CNN Structure

3.3 Classifying Images

The images were converted into 3 dimensional arrays of pixels, and appended to lists in accordance to their labels (*masked or unmasked*). The `"image array"` was split into a ratio of `80:20`, for training and testing respectively.

The images were augmented & a raster of samples were obtained on the basis of *`rotation`*, *`zoom`*, *`width & height shift`* and *`shear`*[10] *range*. The image was *flipped* along the horizontal axis. All the images were then *`filled`*[11] onto a predefined canvas. The aforesaid factors successfully create an augmented dataset which is now ready to pass through any CNN Structure (namely *`ResNet50, MobileNetV2, DenseNet201`* etc).

The processed images then underwent feature extraction through the *convolutional layers*. These layers in a CNN summarize the presence of features in an input image.

Rectified Linear Unit (ReLU) is an "activation function" for our CNN. The output of this function is linear when the input is positive, and zero when the input falls below zero. ReLU is easier to train and often achieves better performance. *SoftMax* is an "activation function" that converts a vector of numbers into a vector of probability. It is used to normalize the output, where each value in the output is the probability of each class.

Pooling layers provide an approach to *down-sampling* features maps by summarizing the presence of features in patches of the feature map. Two common pooling methods are *average pooling* (e.g. *AveragePooling2D*) and *max pooling* (e.g. MaxPooling2D) that summarize the average and the most-activated presence of a feature respectively. A *pooling layer* is a new layer added after the convolutional layer. Specifically, after a nonlinearity (e.g. *ReLU*) has been applied to the feature maps output by a convolutional layer.

[10] Shearing an image will be distorted along an axis, mostly to create or rectify the perception angles.
[11] Filling an image diagonally from the top-left corner to bottom-right corner onto a predefined canvas.

An input tensor used as image input for making the model from the image dataset is now ready for transfer learning. *Transfer learning* is a research problem in machine learning where a model developed for a task is reused as the starting point for a model on a second task. We also specify to not include the fully-connected layer at the top of the network `(i.e. image_top = False)`.

All these parameters are passed into a CNN Structure, from which we separate the output tensor, before applying the aforementioned activation functions. The trainable argument for the layer was `False` to ensure all our parameters cannot be modified after initiation. The *ADAM* `optimizer` finally makes our model trainable with a transfer learning rate (e.g. `0.0001` for MobileNetV2) which is set for optimization.

After training and testing, the acquired models were used to detect the faces of people both masked and unmasked from a video camera feed, from our participants' mobile phones connected on a single network. The advantage of such a source keeps the cost of experimentation low, easy to set up both remotely and independently. Certain parameters of our input feed were changed in the course of experimentation, which rendered different results for better analysis and comparison from our model(s).

Our experiment was performed in a laptop having an `Intel®` `Processor Core(TM) i5-7200U CPU` with a base speed `2.50GHz`. An `Intel®` `HD Graphics 620 3 GB` and an `AMD Radeon™ R5 M420 6 GB`. The RAM used was `Micron 8GB 2400MHz`. Our primary storage is `Serial ATA 1TB 5400RPM`.

4 Results & Discussion

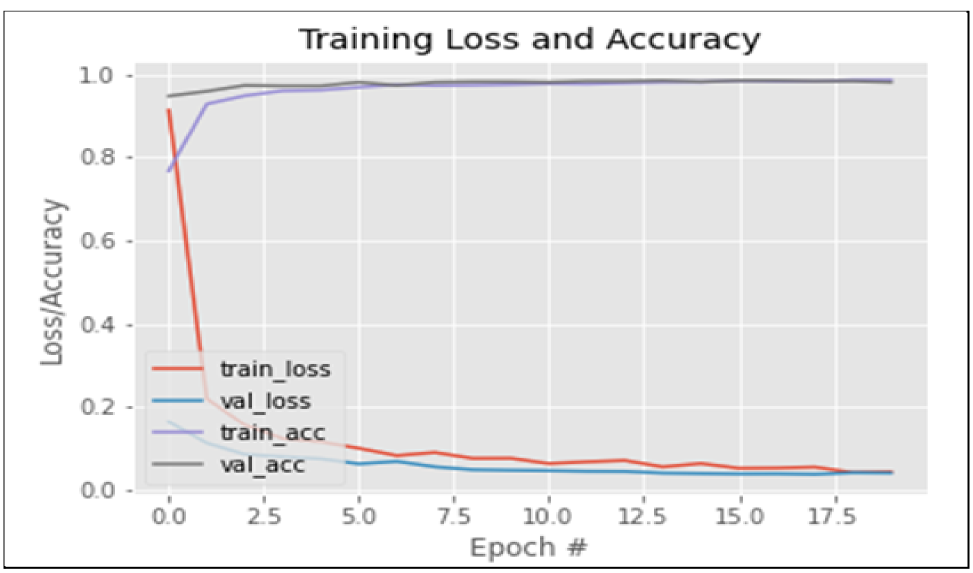
The total dataset is partitioned into an 80:20 ratio where 80 percent is reserved for training and the remaining 20 percent is for testing. From 80 percent, again it is partitioned into 70:30 where 30 percent is for validation data. Data augmentation is also included into the final model. The result models were achieved after comparison of three CNN Models i.e, MobileNetV2, ResNet50, DenseNet201.In the pretrained model, all layers are put on freeze and non fine tuning. Let's discuss each base model briefly.

4.1 MobileNetV2

In the *"MobileNetV2" CNN Structure*, the resultant accuracy and validation accuracy were both 99% with epochs up to 45. The f1-score and training loss-accuracy is shown in Table 1 below and plotted on Graph 1. Using this model, 99.00% test accuracy was achieved.

Table 1: Performance of MobileNetV2 Model for Classification of Masked and Unmasked Face images

	Precision	Recall	F1-Score	Accuracy
with_mask	0.98	0.99	0.99	0.99
without_mask	0.99	0.99	0.99	



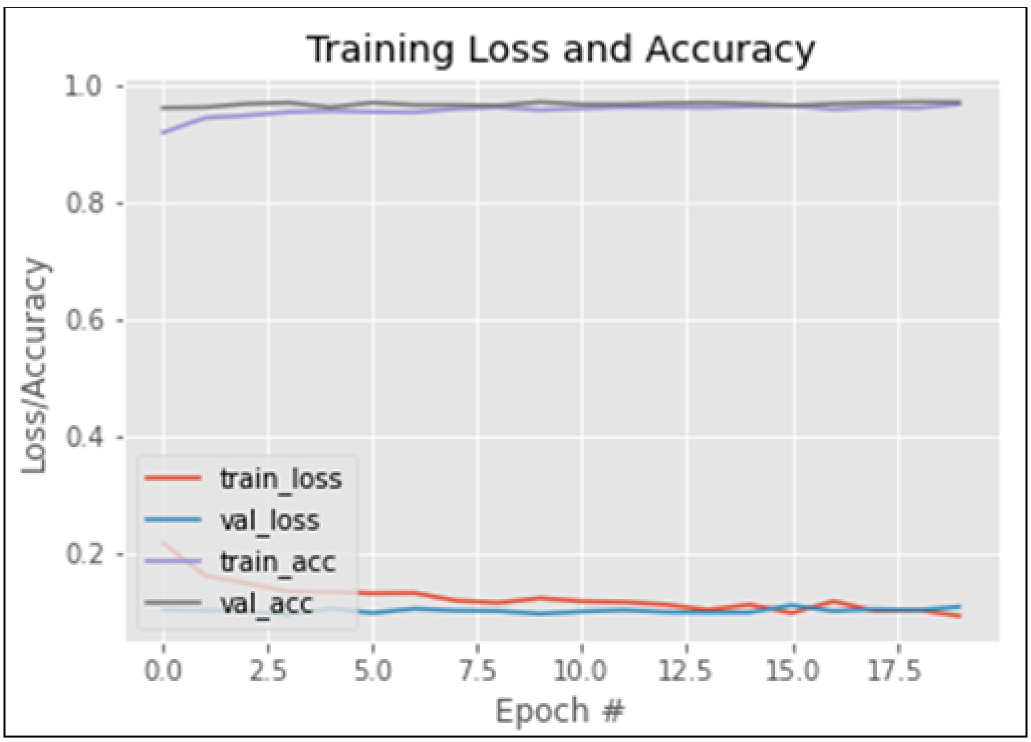
Graph 1: Loss/Accuracy Performance Test to Epoch during MobileNetV2 Model Training

4.2 DenseNet201

In *"DenseNet201" CNN Structure*, the resultant accuracy and validation accuracy were both `97%` with `epochs` up to `45`. The `f1-score` and training `loss-accuracy` is shown in `Table 2` below and plotted on `Graph 2`. In this model, `96.86%` test accuracy was achieved.

Table 2: Performance of DenseNetV2 Model for Classification of Masked and Unmasked Face images

	Precision	Recall	F1-Score	Accuracy
with_mask	0.97	0.95	0.96	0.98
without_mask	0.97	0.98	0.98	



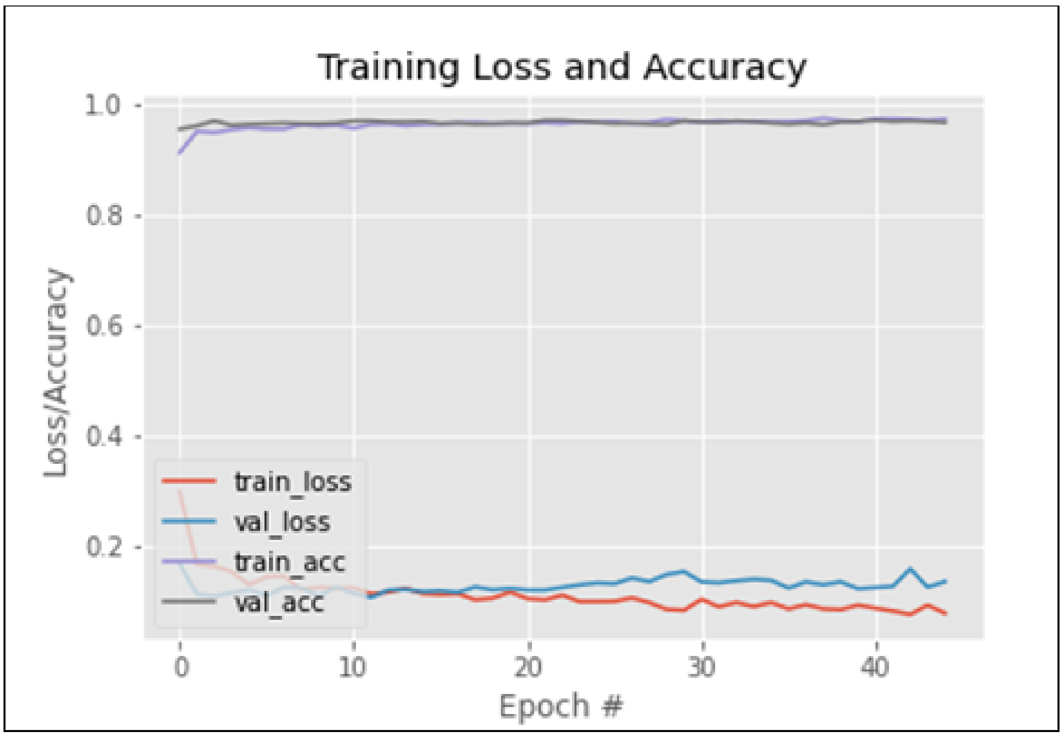
Graph 2: Loss/Accuracy Performance Test to Epoch during DenseNet201 Model Training

4.3 ResNet50

In *"ResNet50" CNN Structure,* the resultant accuracy was `97%` and validation accuracy was `96%` with `epochs` up to `45`. The `f1-score` and training `loss-accuracy` is shown in `Table 3` below and plotted on `Graph 3`. In this model, `95.93%` test accuracy was achieved.

Table 1: Performance of ResNet50 Model for Classification of Masked and Unmasked Face images

	Precision	Recall	F1-Score	Accuracy
with_mask	0.96	0.95	0.95	0.97
without_mask	0.97	0.98	0.97	



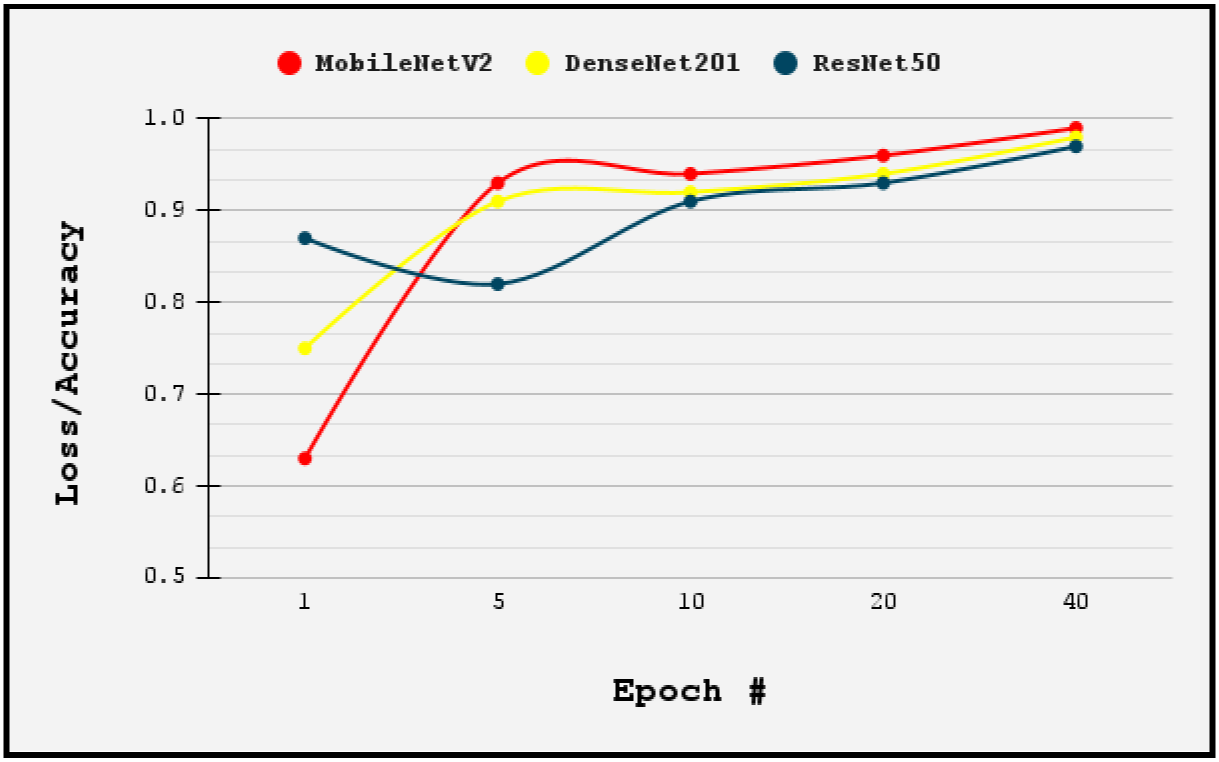
Graph 3: Loss/Accuracy Performance Test to Epoch during ResNet50 Model Training

4.4 Comparison

At least It is clearly visible that the *“MobileNetV2” CNN model* is best among these three models. This model was used in the real time face detection as well.

Table 4: Comparing Accuracy Results of the 3 Models

# Epochs	Accuracy Value	MobileNetV2	DenseNet201	ResNet50
1	Train Accuracy	0.63	0.75	0.87
	Test Accuracy	0.84	0.90	0.95
5	Train Accuracy	0.93	0.91	0.82
	Test Accuracy	0.90	0.90	0.94
10	Train Accuracy	0.94	0.92	0.91
	Test Accuracy	0.96	0.91	0.94
20	Train Accuracy	0.96	0.94	0.93
	Test Accuracy	0.98	0.95	0.95
40	Train Accuracy	0.99	0.98	0.97
	Test Accuracy	0.99	0.98	0.97



Graph 4: Graphically Comparing Accuracy Results of the 3 Models

Thus, the testing section of the dataset is tested with MobileNetV2. In the below given `Figure 6`, the confusion matrix is shown. Here, we can see that the testing data is included with 387 no masked images and 384 masked images. And the final model predicts 388 no masked images and 383 masked images. For predicting 388 no masked images, the model is 383 times correct and 5 times wrong. Similarly, for predicting 383 masked images, 379 times it's correct and 4 times it's wrong.

In the end, we can say that our system can detect faces in real time now and these are our limitations with our limited resources. We'll work hard in future to make it more perfect.

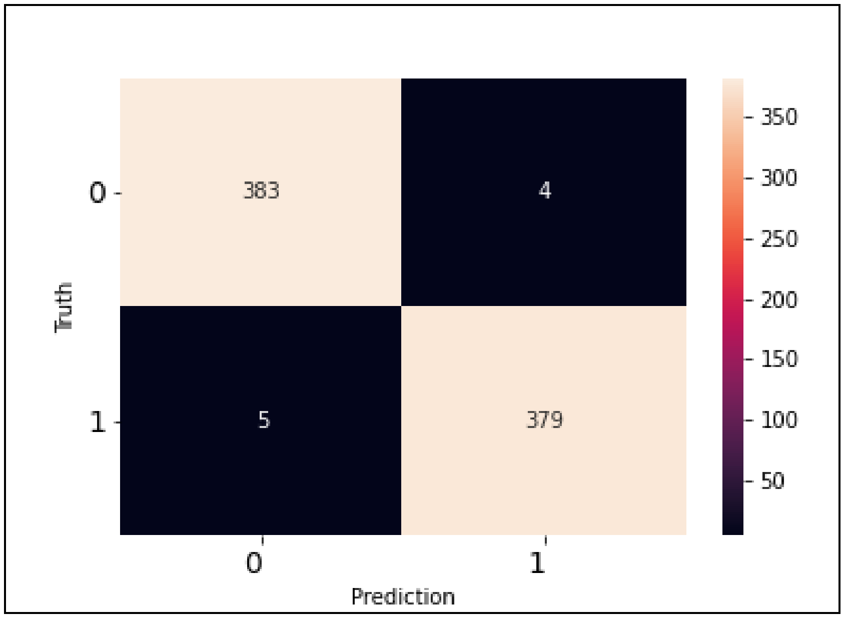


Figure 6: Confusion Matrix

The below given `Figure 7`, shows the clear result after a mask is detected by the system. Cent percent accuracy rate is achieved after using our real time face-mask detection model as it classifies this image as a masked image.

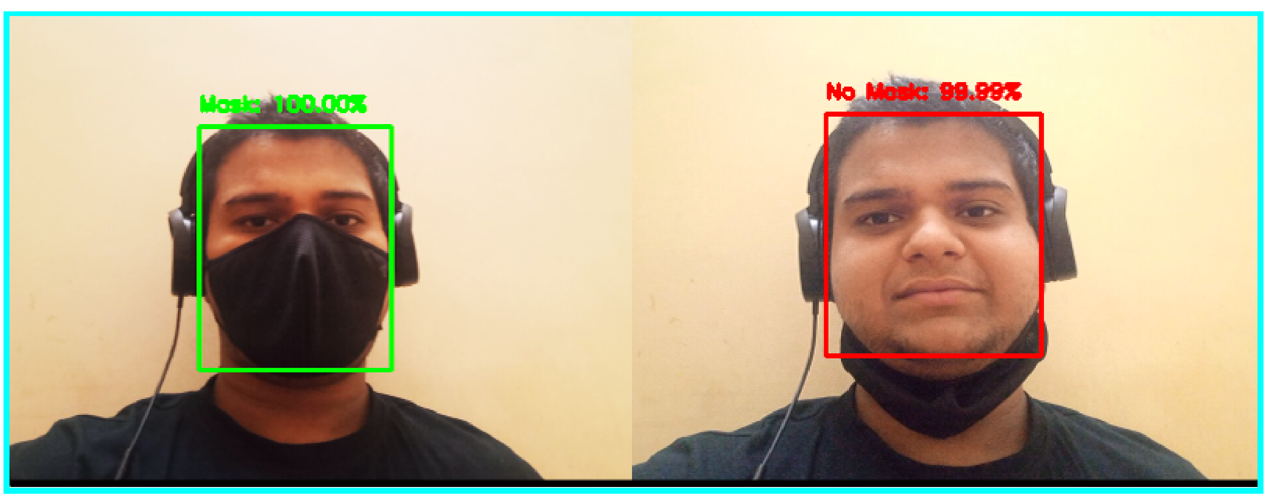


Figure 7: Real-Time Application Results

5 Conclusion

The facemask detection system that we have developed in our collaboration has a considerable success rate and can be implemented to detect masked and unmasked faces. With the number of *Covid-19 cases* still on the rise, wearing facemasks helps to limit the danger escalation, thus providing an effective way to dip the curve. With a source of live feed from closed-circuit television cameras, it can be used by the government to take actions against mask-less citizens in crowded areas like office complexes, markets, railway stations and airports. It can also lead to research in future prospects such as detection of face shields and maintaining social distancing in areas having dense populations, suffering under the grasps of the pandemic.

As more and more research in the field of AI gets conducted, we come up with new and more efficient models which can also run in embedded systems. One such version is *"MobileNetV3",* which was presented at *ICCV in Seoul, South Korea* [X]. Our algorithms can be later built and improved upon, using the same CNN over *MobileNetV3* for higher levels of accuracy and lower latency. These metrics will not only improve the network but also the overall structure with a lower computational expense, thus aiding not only our topic, but the world's technological prospects and research for further betterment of humanity, as a whole.

References

[1] Maliszewska, M., Mattoo, A. & van der Mensbrugghe, D., 2020. *The Potential Impact of COVID-19 on GDP and Trade: A Preliminary Assessment*. https://doi.org/10.1596/1813-9450-9211.

[2] Sahu P (April 04, 2020) *Closure of Universities Due to Coronavirus Disease 2019 (COVID-19): Impact on Education and Mental Health of Students and Academic Staff*. https://doi.org/10.7759/cureus.7541

[3] Stadnytskyi, V., Bax, C. E., Bax, A., & Anfinrud, P. (2020). *The airborne lifetime of small speech droplets and their potential importance in SARS-CoV-2 transmission*. http://doi.org/10.1073/pnas.2006874117.

[4] Hcp, H. P. (2019). *Interim infection prevention and control recommendations for patients with suspected or confirmed coronavirus disease 2019 (COVID-19) in healthcare settings*. https://stacks.cdc.gov/view/cdc/86043.

[5] Feng, S., Shen, C., Xia, N., Song, W., Fan, M., & Cowling, B. J. (2020). *Rational use of face masks in the COVID-19 pandemic*. https://doi.org/10.1016/s2213-2600(20)30134-x.

[6] Iftimie S, López-Azcona AF, Vallverdú I, Hernández-Flix S, de Febrer G, et al. (2021) *First and second waves of coronavirus disease-19: A comparative study in hospitalized patients in Reus, Spain*. https://doi.org/10.1371/journal.pone.0248029.

[7] Y. LeCun, K. Kavukcuoglu and C. Farabet; *Convolutional networks and applications in vision*; https://doi.org/10.1109/ISCAS.2010.5537907

[8] Y. Bengio, A. Courville and P. Vincent; *Representation Learning*: A *Review and New Perspectives*; https://doi.org/10.1109/TPAMI.2013.50.

[9] *Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2017. ImageNet classification with deep convolutional neural networks*. https://doi.org/10.1145/3065386.

[10] K. Zhang, Z. Zhang, Z. Li and Y. Qiao, *Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks*; https://doi.org/10.1109/LSP.2016.2603342.

[11] Boué, L. (2018). *Deep learning for pedestrians: backpropagation in CNNs. arXiv preprint* https://arxiv.org/abs/1811.11987.

[12] Cifuentes-Alcobendas, G., Domínguez-Rodrigo, M. *Deep learning and taphonomy: high accuracy in the classification of cut marks made on fleshed and defleshed bones using convolutional neural networks*. https://doi.org/10.1038/s41598-019-55439-6.

[13] Malakar, S., Ghosh, M., Bhowmik, S. *et al. A GA based hierarchical feature selection approach for handwritten word recognition. Neural Comput & Applic 32, 2533–2552 (2020)*. https://doi.org/10.1007/s00521-018-3937-8

[14] Kisilevsky, B. S., Hains, S. M. J., Lee, K., Xie, X., Huang, H., Ye, H. H., Zhang, K., & Wang, Z. (2003). *Effects of Experience on Fetal Voice Recognition. Psychological Science, 14(3), 220–224*. https://doi.org/10.1111/1467-9280.02435

[15] Jane A. Ou, Stephen H. Penman, *Financial statement analysis and the prediction of stock returns, Journal of Accounting and Economics, Volume 11, Issue 4, 1989, Pages 295-329,* https://doi.org/10.1016/0165-4101(89)90017-7

[16] Chih-Ping Wei, I-Tang Chiu, *Turning telecommunications call details to churn prediction: a data mining approach, Expert Systems with Applications, Volume 23, Issue 2, 2002, Pages 103-112*, https://doi.org/10.1016/S0957-4174(02)00030-1

[17] Shaohua Wan, Yan Liang, Yin Zhang, *Deep convolutional neural networks for diabetic retinopathy detection by image classification, Computers & Electrical Engineering, Volume 72, 2018, Pages 274-282*, https://doi.org/10.1016/j.compeleceng.2018.07.042
[18] Singh, S., Ahuja, U., Kumar, M. et al. *Face mask detection using YOLOv3 and faster R-CNN models: COVID-19 environment*, https://doi.org/10.1007/s11042-021-10711-8
[19] G. Yang *et al; Face Mask Recognition System with YOLOV5 Based on Image Recognition, 2020;* https://doi.org/10.1109/ICCC51575.2020.9345042
[20] J. Ieamsaard, S. N. Charoensook and S. Yammen; *Deep Learning-based Face Mask Detection Using YoloV5;* https://doi.org/10.1109/iEECON51072.2021.9440346
[21] Jignesh Chowdary G., Punn N.S., Sonbhadra S.K., Agarwal S. (2020) *Face Mask Detection Using Transfer Learning of InceptionV3.* In: Bellatreche L., Goyal V., Fujita H., Mondal A., Reddy P.K. (eds) *Big Data Analytics. BDA 2020. Lecture Notes in Computer Science, vol 12581. Springer, Cham*. https://doi.org/10.1007/978-3-030-66665-1_6
[22] Loey, M., Manogaran, G., Taha, M. H. N., & Khalifa, N. E. M. (2021). *A hybrid deep transfer learning model with machine learning methods for face mask detection in the era of the COVID-19 pandemic. Measurement*. https://doi.org/10.1016/j.measurement.2020.108288
[23] Qin B, Li D. *Identifying Facemask-Wearing Condition Using Image Super-Resolution with Classification Network to Prevent COVID-19. Sensors. 2020*; https://doi.org/10.3390/s20185236
[24] Basha, C. M. A. K., Pravallika, B. N., & Shankar, E. B. (2021). *An Efficient Face Mask Detector with PyTorch and Deep Learning. EAI Endorsed Transactions on Pervasive Health and Technology*. https://eudl.eu/doi/10.4108/eai.8-1-2021.167843
[25] S. V. Militante and N. V. Dionisio. *Real-Time Facemask Recognition with Alarm System using Deep Learning*. https://doi.org/10.1109/ICSGRC49013.2020.9232610.
[26] S. V. Militante and N. V. Dionisio, *Deep Learning Implementation of Facemask and Physical Distancing Detection with Alarm Systems*, https://doi.org/10.1109/ICVEE50212.2020.9243183.
[27] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, Quoc V. Le, Hartwig Adam; *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2019,* https://openaccess.thecvf.com/content_ICCV_2019/html/Luo_SegEQA_Video_Segmentation_Based_Visual_Attention_for_Embodied_Question_Answering_ICCV_2019_paper.html