

**APPLIED ARTIFICIAL INTELLIGENCE**

**FACE MASK DETECTION**

Project Report Author:

Under the guidance of: **Balamurugan Govindaraj**

**Prof. Simon Ziegler**

# Table of Contents

[Table of Contents 2](#_Toc78017744)

[1. Project Description 3](#_Toc78017745)

[2. Packages Used 3](#_Toc78017746)

[3. Pre-processing and Training on Dataset 3](#_Toc78017747)

[4. Deployment of Face mask detector 4](#_Toc78017748)

[5. MobileNetV2 for image classification 4](#_Toc78017749)

[6. Open-CV DNN for Face Detection 4](#_Toc78017750)

[7. Result 5](#_Toc78017751)

[8. References 6](#_Toc78017752)

**Figures**

[Figure 1 Screenshot without Mask 5](#_Toc78017734)

[Figure 2 Screenshot with Mask 6](#_Toc78017735)

**List of tables**

[Table 1 List of Packages used 3](#_Toc77923463)

# Project Description

Under the Corona restriction, wearing masks in shopping malls and shops is mandatory. It helps stop the spread of the Corona virus, though the government kept stressing the importance of wearing one. The goal of this project is to detect and predict people without masks using Artificial Intelligence.

Using Image Processing and Computer Vision, one can detect the face mask. In this project, Face mask detector has been build using Tensorflow, Keras, MobileNet and OpenCV.

# Packages Used

|  |  |
| --- | --- |
| Packages | Explanation |
| tensorflow>=1.15.2 | Open source AI library to build models |
| keras>=2.3.1 | Neural network Library |
| imutils>=0.5.3 | A python library for simple image processing functions like translation, rotation, resizing and skeletonization |
| opencv-python>=4.2.0.\* | OpenCV is a Python library which is designed to solve computer vision problems. |
| matplotlib>=3.2.1 | Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. |
| scikit-learn>=0.24.2 | Scikit-learn is a library in Python that provides many unsupervised and supervised learning algorithms. |

Table List of Packages used

# Pre-processing and Training on Dataset

The following steps are performed in pre-processing and training on datasets.

1. Load images from the source with the target size using load\_img() into PIL format. Convert into 3d Numpy array using img\_to\_array() method. The preprocess\_input() function is meant to adequate your image to the format the model requires. Append this image into a list called data and append its category into a list called labels.
2. Perform one-hot encoding on the labels so that the categorical data can be more expressive as numbers. It can be encoded directly and rescaled where needed. This will be done by calling the functions LabelBinarizer(), fit\_transform() and to\_categorical().
3. Split the data using train\_test\_split() to 80% training data and 20% testing data.
4. Augment the data using ImageDataGenerator(). It generates batches of tensor image data with real-time data augmentation.
5. Construct baseModel using mobilenetv2() using imagenet as image classifier. Define headModel then flatten() it and add two Dense() layers, one with ReLU and another with softmax. Construct the model with baseModel as input and headModel as output.
6. Compile the model with adam optimizer. Train the model with model.fit() function.
7. Save the model in h5 format for future use.

# Deployment of Face mask detector

1. Load serialized face detection model using cv2.dnn.readNet as FaceNet. Load our saved mask detection model from disk as MaskNet.
2. Load real-time feed from video stream and read the feed frame by frame.
3. Resize the frame using imutils.resize(). Get the BLOB from the frame using cv2.dnn.blobfromImage()
4. Pass the blob through network and obtain face detection using setInput() and faceNet.forward()
5. If the probability of face 0.5 % get the bounding co-ordinates for the object. If at least one face detected get the predictions using maskNet.predictions().
6. Display the image with red rectangular box if the prediction without mask or display the image with green rectangular box if the prediction with mask.
7. Exit the video stream on pressing the Esc key.

# MobileNetV2 for image classification

MobileNetV2 is a deep learning model based on Convolutional Neural Network. Deep Neural Networks have been deployed for the classification problem in MobileNetV2. TensorFlow loaded pretrained weights from ImageNet. To avoid impairing already learned features, the base layers are frozen. Thereafter, new trainable layers are added, and these layers are trained on the collected dataset to determine the features to classify a face wearing a mask from a face not wearing one. Then the model is fine-tuned, and then the weights are saved. Using pre-trained models helps avoid unnecessary computational costs and helps in taking advantage of already biased weights without losing already learned features.

# Open-CV DNN for Face Detection

The system is built on a single shot multi-box detector (SSD) using the ResNet-10 architecture as its base model. Like the YOLO technique, a single shot is all it takes to detect multiple objects in an image using the Single Shot Multi-Box Detector. It is a much faster, high-accuracy object detection algorithm than others.

# Result

Screen shot without Mask

A person wearing a hat

Description automatically generated with medium confidence

Figure Screenshot without Mask

Screen shot with Mask

A picture containing text, person, indoor, green

Description automatically generated

Figure Screenshot with Mask

# References

[www.machinelearningmastery.com](http://www.machinelearningmastery.com) Date Accessed: 06-Jul-2021

<https://www.analyticsvidhya.com/blog/2021/07/convolution-neural-network-better-understanding/> Date Accessed: 13-Jul-2021

<https://keras.io/api/preprocessing/> Date Accessed: 13-Jul-2021

[www.kaggle.com](http://www.kaggle.com) Date Accessed: 13-Jul-2021