

Face Mask Detection using OpenCv

Submitted by- Akash Jyoti Borah

INTRODUCTION

Face mask detection is an application developed in Python to detect face mask. The application is a real time application which can help in pandemic situation like we witnessed during Covid-19 to ensure safety guidelines.

The technical requirements for the projects are:

1. Programming Language: Python
2. OpenCv
3. Tensorflow/ Keras framework
4. Convolutional Neural Network(transfer learning from MobileNet-v2)
5. Libraries(numpy, imutils, matplotlib, scipy)
6. Google Colab environment

OBJECTIVES:

1. If no person is in the video, it should alert "No Person".
2. If the person walks in, detect the person
3. Detect the person wearing a mask or not with alerts like "Mask Detected" and "No Mask Detected".
4. If a new person walks in, repeat steps 2 and 3 for him/her.
5. If the two persons are standing close enough, it should alert "Maintain Social Distancing".

APPROACH

Step 1: Installing the dependencies

Step 2: Preparing the dataset

The dataset was collected from Kaggle. Performed basic data preprocessing steps and arrange the images in the form of two arrays data and labels.

Step 3: Perform one-hot encoding on the labels

LabelBinarizer was used to perform one hot encoding of labels

Step 4: Data Augmentation

Performed data augmentation to increase the diversity of images in our dataset so that the model perform better in new data.

Step 5: Loading the Pre-trained Model

The pertained model MobileNetV2 was loaded as base model and the head of the model was constructed

Step 6: Compiling the model

Adam optimizer was used with learning rate $1e-4$ and batch size 32. Loss was measured in Binary cross entropy loss.

Step 7: Training the model

The model was trained for 20 epoch and generated an accuracy of 0.98 and validation accuracy of 0.99. Loss was found to be 0.04 and validation loss 0.03 by the end of training.

Step 8: Generating Predictions and printing the Classification Report

Predictions were generated for the test set and when printing the classification report it showed that the model was working fine and ready to be deployed.

Step 9: Plotting the results

The training and loss accuracy was plotted and it showed great results.

Step 10: Saving the model

Using Open CV for face detection

Step 11: Loading the face detector model of OpenCv and the mask detector model we developed.

Step 12: Importing extra functions to access Video feed on Colab notebook.

Step 13: Preparing the primary functions to detect and predict mask which takes arguments the frames(i.e. images, face detector and mask detector).

Step 14: Face mask detection is performed to capture images from webcam and it captures frames in live video feed. Face mask is detected and showcase the mentioned functionality.

Step 15: The stream of images are written to a file which are processed to make a live video.