

FACE MASK DETECTOR

Object: Face mask detector with Computer Vision in Python. Project for safety against COVID-19.

In this project, we will build a model to detect whether a person is wearing a mask in real-time. We will do this using the concepts of computer vision using the OpenCV library and Keras.

Programming Language: Python

Tools and Libraries: OpenCV, TensorFlow, Numpy, Keras

IDE: Visual Studio Code

Prerequisites: Python, Deep Learning, Machine Learning

BEFORE STARTING INSTALL THE FOLLOWING

- Python 3.9
- TensorFlow
- Numpy
- OpenCV
- imutils

CODE

- train.py

With this code we train the Neural Network.

Build the Neural Network.

```
model = Sequential([
    Conv2D(100, (3,3), activation='relu', input_shape=(150, 150, 3)),
    MaxPooling2D(2,2),

    Conv2D(100, (3,3), activation='relu'),
    MaxPooling2D(2,2),

    Flatten(),
    Dropout(0.5),
    Dense(50, activation='relu'),
    Dense(2, activation='softmax')
])
model.compile(optimizer='adam', loss='binary_crossentropy',
metrics=['acc'])
```

This convolution network consists of two pairs of Conv and MaxPool layers to extract features from the dataset. Which is then followed by a Flatten and Dropout layer to convert the data in 1D and ensure overfitting.

And then two Dense layers for classification.

Image Data Generation/Augmentation.

```
TRAINING_DIR = "Dataset/train"
train_datagen = ImageDataGenerator(rescale = 1.0 / 255,
                                   rotation_range = 40,
                                   width_shift_range = 0.2,
                                   height_shift_range = 0.2,
                                   shear_range = 0.2,
                                   zoom_range = 0.2,
                                   horizontal_flip = True,
                                   fill_mode = 'nearest')

train_generator = train_datagen.flow_from_directory(TRAINING_DIR,
                                                    batch_size = 100,
                                                    target_size = (150, 150))
```

Initialize a callback checkpoint to keep saving best model after each epoch while training.

```
checkpoint = ModelCheckpoint('model2-{epoch:03d}.model', monitor='val_loss', verbose=0, save_best_only=True, mode='auto')
```

Train the model.

```
history = model.fit_generator(train_generator,
                              epochs = 30,
                              validation_data=validation_generator,
                              callbacks=[checkpoint])
```

- test.py

With this code we test our Neural Network.

Once the program is run, you can press ESC to exit.

Load the last model.

```
model = keras.models.load_model("./model-010.h5")
```

Define the camera.

```
webcam = cv2.VideoCapture(0) # Use camera 0
```

Load the .xml file where the frontal face is defined.

```
classifier =  
cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
```

Starting live detection.

```
while True:
    (rval, im) = webcam.read()
    # Resize the image to speed up detection
    mini = cv2.resize(im, (im.shape[1] // size, im.shape[0] //
size))

    # detect MultiScale / faces
    faces = classifier.detectMultiScale(mini)
```

After detecting the face, draw a rectangle around it and consider only that piece of image to detect the face mask.