KENDRIYA VIDYALAYA
MALLESHWARAM
Bangalore – 55

Project Name:

Computer Vision

ACADEMIC YEAR : 2021-22

Project Report on Computer Vision

Group Members

Class 11 B	
Name	**Roll no.**
Pranav A S	11233
Venkat Narayan T L	11251
Yadhu Krishna	11253
R S Vijaysuraj	11237
Pratyush Govind K P	11235

Project Guide -

Bhavana **PGT(CS)**

Signature

TABLE OF CONTENTS

INTRODUCTION

What Does Computer Vision Mean?

Computer vision (CV) is the subcategory of artificial intelligence (AI) that focuses on building and using digital systems to process, analyze and interpret visual data. The goal of computer vision is to enable computing devices to correctly identify an object or person in a digital image and take appropriate action.

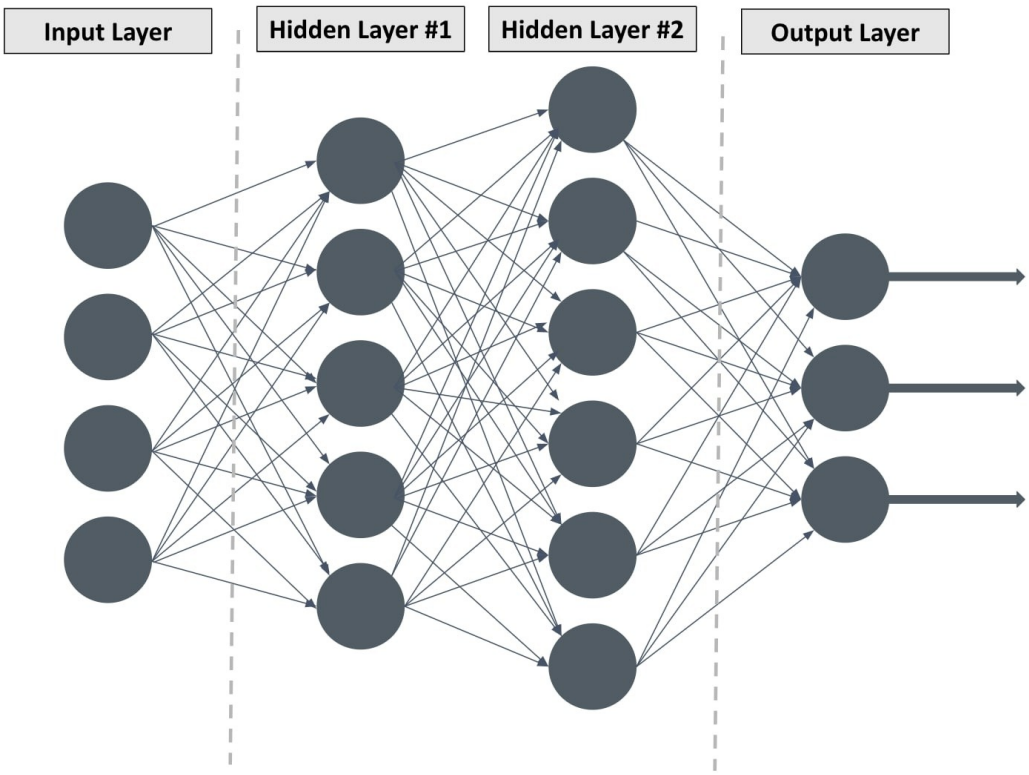
Computer vision's aim is to enable computers to perform the same kind of tasks as humans with the same efficiency.

Real Time Face Mask Detection With OpenCV:

In this pandemic situation, we all should take some safety precautions. Wearing a mask is one of them. But many people don't want to wear a mask. So today we are going to build a face mask detection project that tells if a person is wearing a mask or not. Using Opencv python and Deep learning we can build a face mask detector.

What is Deep Learning?

Deep learning is a type of machine learning and a subset of artificial neural networks. Deep learning is inspired by the human brain, and it learns from a large amount of data. It is used to solve complex problems that are very hard to solve by using typical machine learning algorithms.

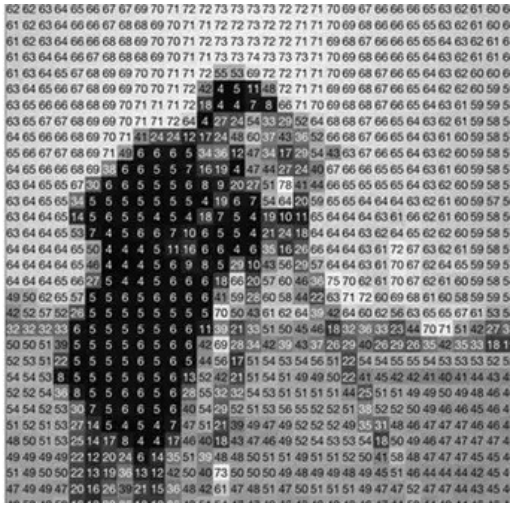


WHAT IS IMAGE PROCESSING?

Before implementing face mask detection problem, first we need to understand that how to handle images. Images are simply a collection of colors in red, green and blue format. As a human we see an image with some object or shape in it, but for computer it is just an array with color values range from 0 to 255.The way computer sees anything is different from the way human see an image. But that's the good news for us because if we got an array of the image than it becomes simple for us to implement any algorithm on that array.

Steps to Perform Image Processing :

- Load images using Python or any other programming you are working on.
- Convert images into array
- And finally apply some algorithm on that array



OpenCV

- OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library.
- The library has more than 2500 optimized algorithms.
- It has C++, Python, Java and MATLAB interfaces and supports Windows, Linux,Android and MacOS.
- Will help us to load images in Python and convert them into array.
- Each index of array represents (red, green, blue) color pixel which ranges from 0 to 255.

Features of OpenCV:

- Face Detection
- Geometric Transformations
- Image Thresholding
- Smoothing Images
- Background Removals
- Image Segmentation

Installing OpenCV:

- Open Command Prompt
- Type -

```
pip install opencv-python
```

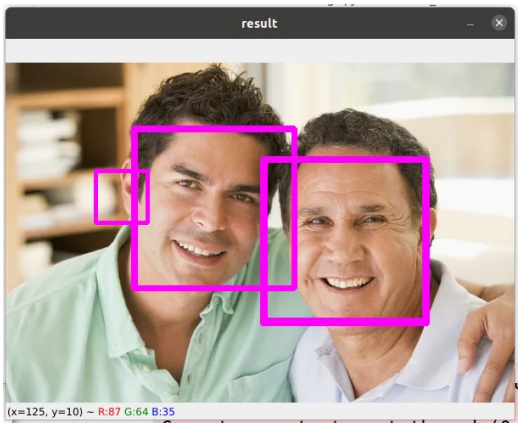
FACE DETECTION USING OpenCV

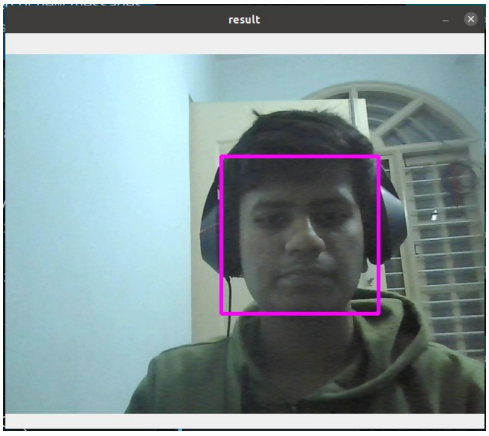
- To detect faces in OpenCV we are going to use the Voila-Jones Object Detection Framework.
- We are going to download **haarcascade_frontalface_default.xml**
- Code to detect face from an **Image**:

```
import cv2

haar_data =
cv2.CascadeClassifier('/home/pranav/Desktop/Python
Project/data.xml')

img = cv2.imread(‘/home/pranav/Desktop/Python
Project/tom.png’)

while True:

   faces = haar_data.detectMultiScale(img)

   for x,y,w,h in faces:

      cv2.rectangle(img,(x,y),(x+w,y+h), (255,0,255),4)

   cv2.imshow('result',img)

   #27- ASCII of Escape

   if cv2.waitKey(2)== 27:

      break

cv2.destroyAllWindows()
```

- Code to detect face from a **Webcam:**

```
import cv2

haar_data = cv2.CascadeClassifier('/home/pranav/Desktop/Python
Project/data.xml')

capture = cv2.VideoCapture(0)

while True:

   flag, img = capture.read()

   if flag:

      faces = haar_data.detectMultiScale(img)

      for x,y,w,h in faces:

         cv2.rectangle(img,(x,y),(x+w,y+h),(255,0,255),4)

      cv2.imshow('result',img)

      #27 - ASCII of Escape

      if cv2.waitKey(2)==27:

         break

capture.release()

cv2.destroyAllWindows()
```





Training an Algorithm to Detect Facemasks

- We first need to take pictures **without mask**. Which can be done by modifying the webcam code as:

```
import cv2

import numpy as np

haar_data = cv2.CascadeClassifier('/home/pranav/Desktop/Python
Project/data.xml')

capture = cv2.VideoCapture(0)

data = []

while True:

    flag, img = capture.read()

    if flag:

        faces = haar_data.detectMultiScale(img)

        for x,y,w,h in faces:

            cv2.rectangle(img,(x,y),(x+w,y+h),(255,0,255),4)

            face = img[y:y+h,x:x+w, : ]

            face = cv2.resize(face,(50,50))

            if len(data) < 400:

                data.append(face)

        cv2.imshow('result',img)

        #27 - ASCII of Escape
```

```
    if cv2.waitKey(2)==27 or len(data) >= 200:
            break
capture.release()
cv2.destroyAllWindows()
np.save('without_mask.npy',data)
```

- Then we take pictures while wearing a mask

```
import cv2
import numpy as np
haar_data = cv2.CascadeClassifier('/home/pranav/Desktop/Python
Project/data.xml')
capture = cv2.VideoCapture(0)
data = []
while True:
    flag, img = capture.read()
    if flag:
        faces = haar_data.detectMultiScale(img)
        for x,y,w,h in faces:
            cv2.rectangle(img,(x,y),(x+w,y+h),(255,0,255),4)
            face = img[y:y+h,x:x+w, : ]
            face = cv2.resize(face,(50,50))
            if len(data) < 400:
                data.append(face)
        cv2.imshow('result',img)
```

```
        #27 - ASCII of Escape
        if cv2.waitKey(2)==27 or len(data) >= 200:
            break
capture.release()
cv2.destroyAllWindows()
np.save('with_mask.npy',data)
```

- Now we write a code to teach the Machine to differentiate:

```
import numpy as np
import cv2
with_mask = np.load('with_mask.npy')
without_mask = np.load('without_mask.npy')
with_mask.shape
without_mask.shape
with_mask = with_mask.reshape(200,50*50*3)
without_mask = without_mask.reshape(200,50*50*3)
without_mask.shape
with_mask.shape
X = np.r_[with_mask,without_mask]
X.shape
labels = np.zeros(X.shape[0])
labels[200:] = 1.0
names = {0: 'Mask', 1:'No Mask'}
```

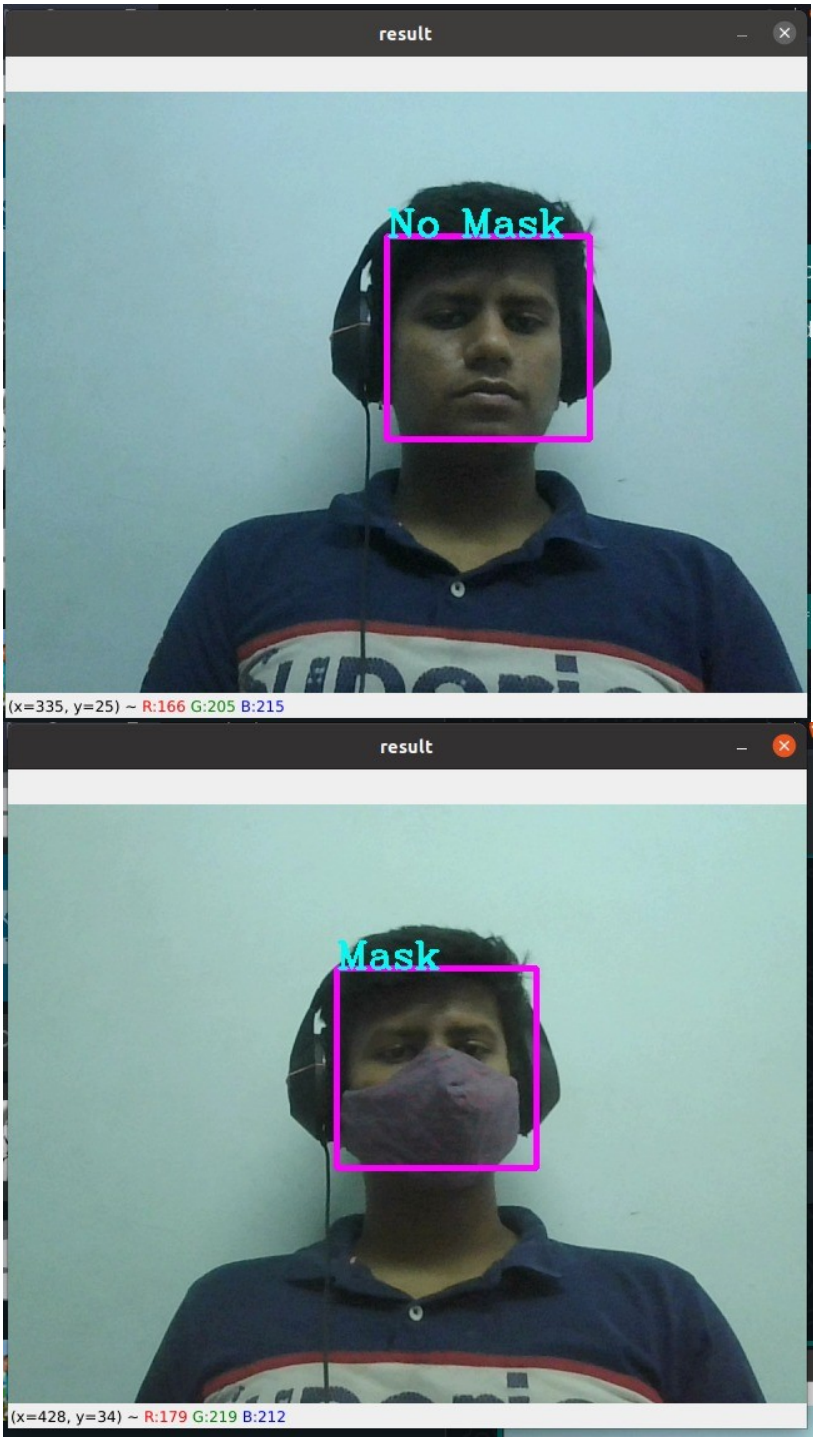
```
     #python - sklearn - scikit-learn

     # svm - Support Vector Machine

     #SVC - Support Vector Classification

     from sklearn.svm import SVC

     from sklearn.metrics import accuracy_score

     from sklearn.model_selection import train_test_split

     x_train,x_test,y_train,y_test =
train_test_split(X,labels,test_size=0.25 )

     x_train.shape

     # PCA - Princippal Component Analysis

     from sklearn.decomposition import PCA

     pca = PCA(n_components = 3)

     x_train= pca.fit_transform(x_train)

     x_train[0]

     x_train.shape

     x_train,x_test,y_train,y_test =
train_test_split(X,labels,test_size=0.25 )

     svm = SVC()

     svm.fit(x_train,y_train)

     #x_test = pca.transform(x_test)

     y_pred =svm.predict(x_test)

     accuracy_score(y_test, y_pred)
```

- Now the Final Code:

```
import cv2

import numpy as np

haar_data = cv2.CascadeClassifier('/home/pranav/Desktop/Python
Project/data.xml')

capture = cv2.VideoCapture(0)

font = cv2.FONT_HERSHEY_COMPLEX

while True:

    flag, img = capture.read()

    if flag:

        faces = haar_data.detectMultiScale(img)

        for x,y,w,h in faces:

            cv2.rectangle(img,(x,y),(x+w,y+h),(255,0,255),4)

            face = img[y:y+h,x:x+w, :]

            face = cv2.resize(face,(50,50))

            face = face.reshape(1,-1)

            #face = pca.transform(face)

            pred = svm.predict(face)

            n = names[int(pred)]

            cv2.putText(img, n, (x,y), font, 1, (244, 250, 0), 2)

            print(n)

        cv2.imshow('result',img)

        #27 - ASCII of Escape
```

```
            if cv2.waitKey(2)==27:
                break
    capture.release()
    cv2.destroyAllWindows()
```

OUTPUT


result
No Mask
(x=335, y=25) ~ R:166 G:205 B:215
result
Mask
(x=428, y=34) ~ R:179 G:219 B:212

REQUIREMENTS

- **Hardware**

  Webcam

- **Software**

  Python, OpenCV, Anaconda

BIBLIOGRAPHY

- https://opencv.org/
- https://github.com/opencv/opencv/tree/master/data/haarcascades
- https://en.wikipedia.org/wiki/Viola%E2%80%93Jones_object_detection_framework
- https://brain-mentors.com/face-mask-detection-using-opencv-in-python/
- https://www.youtube.com/watch?v=csYkGeJ5bCM
- https://www.youtube.com/watch?v=dQw4w9WgXcQ