

Face mask and social distancing detection

Team

- Ayush N Patel (190907662)
- Kotla Ujwal Reddy (190907664)
- Yash Agarwal (190907580)
- Aravind (190907304)

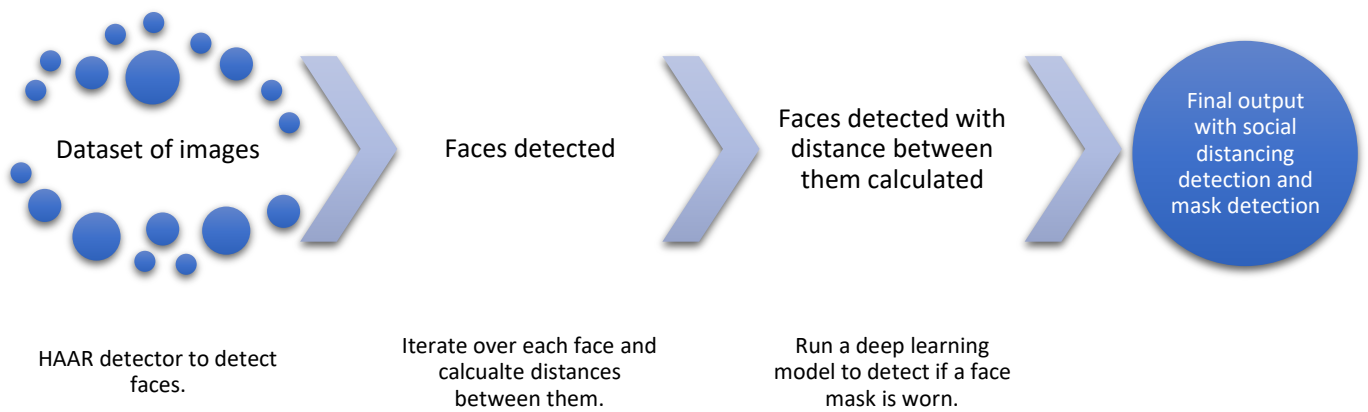
Objective

To design a deep learning model that can detect if a person is wearing a face mask and if two adjacent people are following social distancing norms.

Methodology

- Detect faces
- Calculate distance between each face
- Mask detection on detected faces

Process Flow



Process Flow

1. Three datasets are to be used, they are:
 - a. Haar^[1] cascade database^[2] for face detection
 - b. A training and validation database^[3] for mask detection
 - c. A test database^[4] for testing mask detection and social distancing violation
2. First, the database is run through a haar^[1] cascade model to detect faces.
3. If more than one face is detected, the faces in those images are iterated over and the distances^[5] between them are calculated.
4. If social distancing violations are detected (<6ft), a visual alert is thrown.
5. The faces detected and the distances calculated are appended to a dictionary to be referenced later.
6. Post successful face detection and distance calculation, the images along with the dictionary are sent through to a mask detection model that is to be trained on an existing Kaggle database.
7. Final output image returns an image with the detected faces, distances between each face and if they are wearing a mask, overlayed on the original image.

Code

- Importing necessary libraries

```
import numpy as np    #linear algebra
import pandas as pd    #data processing, CSV file I/O (e.g.
pd.read_csv)
import cv2
from scipy.spatial import distance
```

- Importing haar cascade file for frontal face detection

```
#loading haarcascade_frontalface_default.xml
face_model = cv2.CascadeClassifier('../input/haar-cascades-for-face-
detection/haarcascade_frontalface_default.xml')
```

- Plotting a box around the detected faces and trying the computation on a sample image

```
import matplotlib.pyplot as plt

#trying it out on a sample image
img = cv2.imread('../input/face-mask-
detection/images/maksssksksss249.png')

img = cv2.cvtColor(img, cv2.IMREAD_GRAYSCALE)

faces = face_model.detectMultiScale(img) #returns a list of
(x,y,w,h) tuples

out_img = cv2.cvtColor(img, cv2.COLOR_RGB2BGR) #colored output image

#plotting
for (x,y,w,h) in faces:
    cv2.rectangle(out_img,(x,y),(x+w,y+h),(0,0,255),2)
plt.figure(figsize=(10,10))
plt.imshow(out_img)
```

- Checking for social distancing violations

```

MIN_DISTANCE = 130
if len(faces)>=2:
    label = [0 for i in range(len(faces))]
    for i in range(len(faces)-1):
        for j in range(i+1, len(faces)):
            dist = distance.euclidean(faces[i][:2],faces[j][:2])
            if dist<MIN_DISTANCE:
                label[i] = 1
                label[j] = 1
    new_img = cv2.cvtColor(img, cv2.COLOR_RGB2BGR) #colored output
    image
    for i in range(len(faces)):
        (x,y,w,h) = faces[i]
        if label[i]==1:
            cv2.rectangle(new_img,(x,y),(x+w,y+h),(255,0,0),2)
        else:
            cv2.rectangle(new_img,(x,y),(x+w,y+h),(0,255,0),2)
    plt.figure(figsize=(10,10))
    plt.imshow(new_img)

else:
    print("No. of faces detected is less than 2")

```

Sample Outputs

- Face detection



Figure 1.1 – Sample image 257



Figure 1.2 – Sample image 269

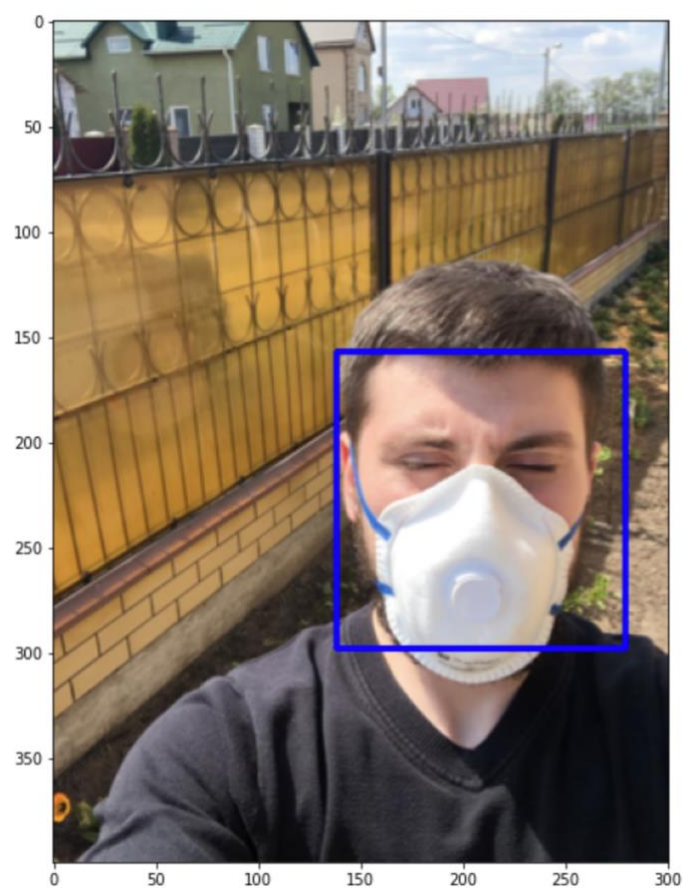


Figure 1.3 – Sample image 262

- Social distancing detection
(Step2, post face detection)
(Green indicates no violation of social distancing; Red indicates a violation of social distancing)



Figure 2.1 – Sample image 257



Figure 2.2 – Sample image 269

No. of faces detected is less than 2

Figure 2.3 – Sample image 262

Roadmap

To search and implement a suitable model for detection of face mask to work in conjunction with the haar and distance detector.

References

- [1] P. Viola; M. Jones "Rapid object detection using a boosted cascade of simple features" in Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, 15 April 2003
- [2] "Haar Cascades for Face Detection" kaggle.com
<https://www.kaggle.com/datasets/gpreda/haar-cascades-for-face-detection> (accessed Sept. 20, 2022)
- [3] "Face Mask Detection ~12K Images Dataset" kaggle.com
<https://www.kaggle.com/datasets/ashishjangra27/face-mask-12k-images-dataset> (accessed Sept. 29, 2022)
- [4] "Face Mask Detection" kaggle.com
<https://www.kaggle.com/datasets/andrewmvd/face-mask-detection> (accessed Sept. 29, 2022)
- [5] "OpenCV Social Distancing Detector" pyimagesearch.com
<https://pyimagesearch.com/2020/06/01/opencv-social-distancing-detector/> (accessed Oct. 1, 2022)