

# **A Project Report**

## **ON**

### **Face Detection in Images :**



#### **Submitted by:**

Fiza ( 2k18 - ELE- 31)

Ubaid-ur-Rehman ( 2k18- ELE -115)

(Faculty of Engineering and Technology)  
Univeristy OF Sindh , Jamshoro  
Bachelor of Electronics 4<sup>rd</sup> year

Subject : Computer Vision and Image Processing

Teacher : Dr. Sundar Ali Khuwaja

Github Repository : <https://github.com/FizaShakeel/CV-Assignment>

## Contents:

1. Introduction
2. System Requirments
3. Code
4. Implementations
5. Results and Discussions

## Introduction:

Face detection in Images is to describe a collection of related computer vision tasks that involve activities like identifying objects like faces in images such as predicting the class of one object like face in an image. It is a technique that identifies or locates human faces in an image.

So, we can distinguish these Computer vision tasks with examples :

**classification** : This is done by predicting the type of class of an object in an image.

**Input** : we are given a single image.

**Output** : one or more images are detected

## System Requirements :

1. Python
2. Open CV
3. Numpy
4. Json
5. Matplotlib

## Code:

```
#!/usr/bin/env
python

# coding: utf-8
# Let's add the libraries where they are really needed, not all of them at
the first line
address = '../input/face-detection-in-images/face_detection.json'
# In[2]:
import json
import codecs
# In[3]:
# get links and stuff from json
jsonData = []
```

```
with codecs.open(address, 'rU', 'utf-8') as js:
    for line in js:
        jsonData.append(json.loads(line))
print(f"{len(jsonData)} image found!")
print("Sample row:")
jsonData[0]
# In[4]:
import numpy as np
import requests
from tqdm import tqdm
from PIL import Image
from io import BytesIO
# In[5]:
# load images from url and save into images
images = []
for data in tqdm(jsonData):
    response = requests.get(data['content'])
    img = np.asarray(Image.open(BytesIO(response.content)))
    images.append([img, data["annotation"]])
# In[6]:
get_ipython().system('mkdir face-detection-images')
# In[7]:
import cv2
import time
# In[8]:
count = 1
totalfaces = 0
start = time.time()
for image in images:
    img = image[0]
    metadata = image[1]
    for data in metadata:
        height = data['imageHeight']
        width = data['imageWidth']
        points = data['points']
        if 'Face' in data['label']:
            x1 = round(width*points[0]['x'])
            y1 = round(height*points[0]['y'])
            x2 = round(width*points[1]['x'])
            y2 = round(height*points[1]['y'])
```

```
        cv2.rectangle(img, (x1, y1), (x2, y2), (0, 0, 255), 1)
        totalfaces += 1
    cv2.imwrite('./face-detection-
images/face_image_{}.jpg'.format(count),img)
    count += 1

end = time.time()
print("Total test images with faces : {}".format(len(images)))
print("Sucessfully tested {} images".format(count-1))
print("Execution time in seconds {}".format(end-start))
print("Total Faces Detected {}".format(totalfaces))
# In[9]:
import matplotlib.pyplot as plt
# In[10]:
face1 = cv2.imread("./face-detection-images/face_image_64.jpg")
# In[11]:
plt.figure(figsize=(20,25))
plt.imshow(face1)
plt.show()
# In[12]:
plt.figure(figsize=(18,15))
plt.imshow(cv2.cvtColor(face1, cv2.COLOR_BGR2RGB))
# In[13]:
face2 = cv2.imread("./face-detection-images/face_image_400.jpg")
# In[14]:
plt.figure(figsize=(20,25))
plt.imshow(face2)
plt.show()
```

## Implementations:

Application	Implementations
Face detection in Images	This is code for object detection and I am just detection the images through this code . And other thing is convert BGR to RGB (color).

## Results and Discussions:

Simple face detection in image with bounding box:



Convert RGB to BGR :



Another Detection of faces in image :



