#!/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

# In[1]:

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()