Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

PROJECT REPORT

IMAGE PROCESSING USING PYTHON AND SELINIUM

COURSE CODE: ECE3999

TEAM MEMBERS:
K J DEEPAK SOMESH – 18BEC0920
SRIJAN VARMA – 18BEC0906
PALAK TRIPATHI – 18BEC0516
RAMANA ANAND S – 18BEC0907

UNDER THE GUIDANCE OF

PROF. SIVACOUMAR R

ABSTRACT

This project Is based around images. Where you get to read the image and match the images. Its an application based project like a step by step process.

It starts with analyzing the image, representing the images in RGB types, detecting age and gender and then it provides the output of analyzed data.(Extras) Based on the data you have, you going to find its similar match using the other images (like comparing the data with other images and finding its similar match). This process can be done either in offline mode (having your own set of images to compare) or online mode.

In online mode we make our application to access the chrome data and search for a bunch of images(as many as you wish) and predict the similar or related one.

**we are also going automate a particular website for advanced predictions

Literature Study

Now a days images are not just a item of captured scenario. But we can retrieve too much information from it. Images are classified into different types and formats to differentiate between its data types. Today we see images have tagged with gps data of where the image was captured. With this image data, every smart device with image processing features are making life easier for the human beings, the near future is mostly dependent on the image processing. Everything from security systems to virtual meetings are dependent on image processing, AI/ML.

Day by day the pixels of camera and resolutions of the images are getting evolved thus the data of the image is also becoming big. People now take too many pictures and store it. But with this too many pictures people often feel difficult to find the exact picture of the exact date they want. And in expectance of a perfect picture we take too many pictures and waste the storage by creating duplicates.

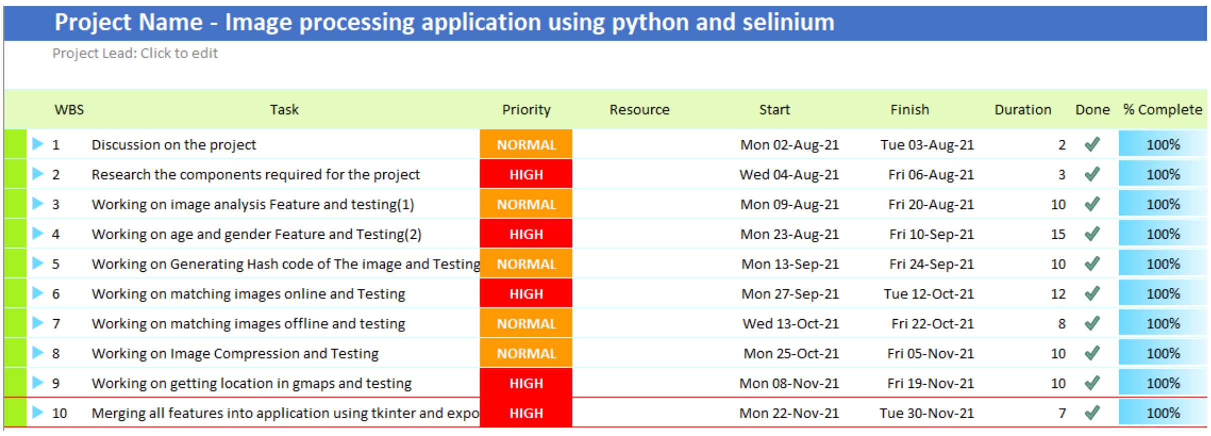
Bhumika Gupta (2017), proposed object detection is a well-known computer technology connected with computer vision and image processing that focuses on detecting objects or its instances of a certain class (such as humans, flowers, animals) in digital images and videos. There are various applications of object detection that have been well researched including face detection, character recognition, and vehicle calculator. Object detection can be used for various purposes including retrieval and surveillance. In this study, various basic concepts used in object detection while making use of OpenCV library of python 2.7, improving the efficiency and accuracy of object detection are presented.

Karanbir Chahal (2018) et al. proposed Object detection is the identification of an object in the image along with its localization and classification. It has wide spread applications and is a critical component for vision-based software systems. This paper seeks to perform a rigorous survey of modern object detection algorithms that use deep learning. As part of the survey, the topics explored include various algorithms, quality metrics, speed/size trade-offs and training methodologies. This paper focuses on the two types of object detection algorithms- the SSD class of single step detectors and the Faster R-CNN class of two step detectors. Techniques to construct detectors that are portable and fast on low powered devices are also addressed by exploring new light weight convolutional base architectures. Ultimately, a rigorous review of the strengths and weaknesses of each detector leads us to the present state of the art.

Problem Statement

In this era taking photos and storing it Is not a big deal. Day by day the pixels of camera and resolutions of the images are getting evolved thus the data of the image is also becoming big. People now take too many pictures and store it. But with this too many pictures people often feel difficult to find the exact picture of the exact date they want. And in expectance of a perfect picture we take too many pictures and waste the storage by creating duplicates. So this application provides you with all the features which makes it easy to find duplicates of the image. And this application have set of features at one place which makes it comfortable for people to manipulate the pictures in the gallery however we want. In our Project we had the task of analysing the image and to determine all the components of the captured image. In this project we are going to analyze and detect the age and gender of the people. Then we are going to get a code (Hash Code) of the image and match it to the minimum difference between the code of the image. To achieve all this, we should have a OpenCV library. OpenCV is the base of this project and we are using python coding language to build this project. Firstly OpenCV (Open-Source Computer Vision Library) is an open-source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. In this project we are going to create a separate app using python so that, we can run this project without help of an IDE.

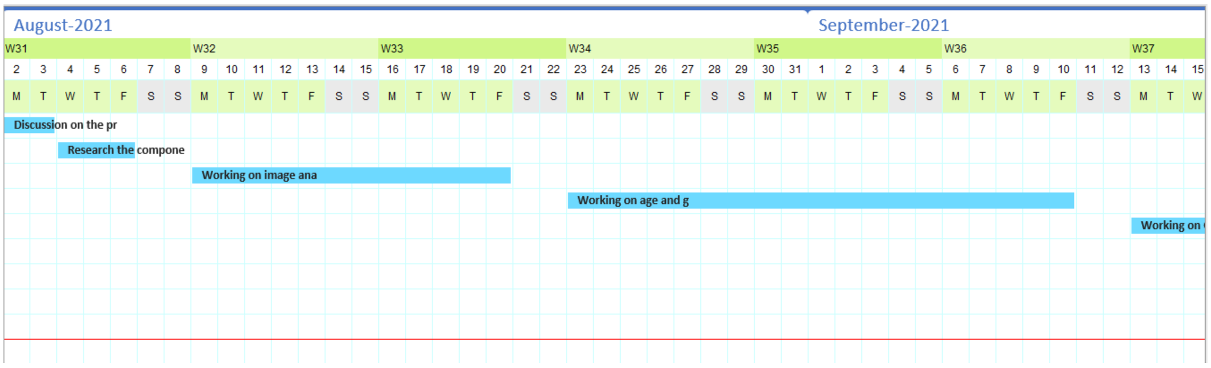
GANT CHART:

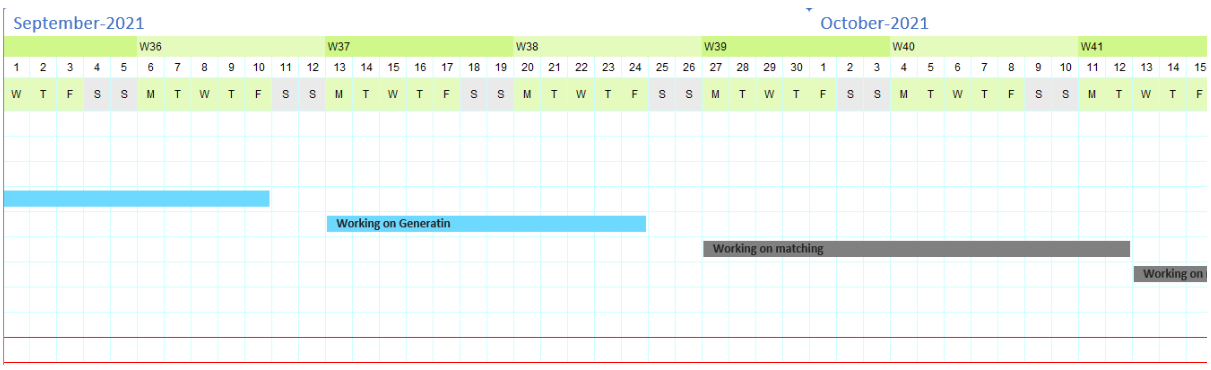


Project Name - Image processing application using python and selinium

Project Lead: Click to edit

WBS	Task	Priority	Resource	Start	Finish	Duration	Done	% Complete
1	Discussion on the project	NORMAL		Mon 02-Aug-21	Tue 03-Aug-21	2	✓	100%
2	Research the components required for the project	HIGH		Wed 04-Aug-21	Fri 06-Aug-21	3	✓	100%
3	Working on image analysis Feature and testing(1)	NORMAL		Mon 09-Aug-21	Fri 20-Aug-21	10	✓	100%
4	Working on age and gender Feature and Testing(2)	HIGH		Mon 23-Aug-21	Fri 10-Sep-21	15	✓	100%
5	Working on Generating Hash code of The image and Testing	NORMAL		Mon 13-Sep-21	Fri 24-Sep-21	10	✓	100%
6	Working on matching images online and Testing	HIGH		Mon 27-Sep-21	Tue 12-Oct-21	12	✓	100%
7	Working on matching images offline and testing	NORMAL		Wed 13-Oct-21	Fri 22-Oct-21	8	✓	100%
8	Working on Image Compression and Testing	NORMAL		Mon 25-Oct-21	Fri 05-Nov-21	10	✓	100%
9	Working on getting location in gmaps and testing	HIGH		Mon 08-Nov-21	Fri 19-Nov-21	10	✓	100%
10	Merging all features into application using tkinter and expo	HIGH		Mon 22-Nov-21	Tue 30-Nov-21	7	✓	100%





Methodology

- As we know this project is around images and the application itself is to analyse and decode the image. so, we use python language for this project and for decoding the image in certain levels we use python libraries.
- Now to get the properties of the image we use PIL library to get properties using the commands. the properties we get is like on the image size, resolution, exit data, format, etc.

- The analysing part is where it comes to know certain views and specs of the image. like we get how the image has a base where it includes RGB colours. so, we get the output in RGB filter so that we can sense the presence of the number of REDS, GREENS and BLUES. to further compile it we use channel filter analysis to highlight the certain parts of the image where RGB is present in the image.

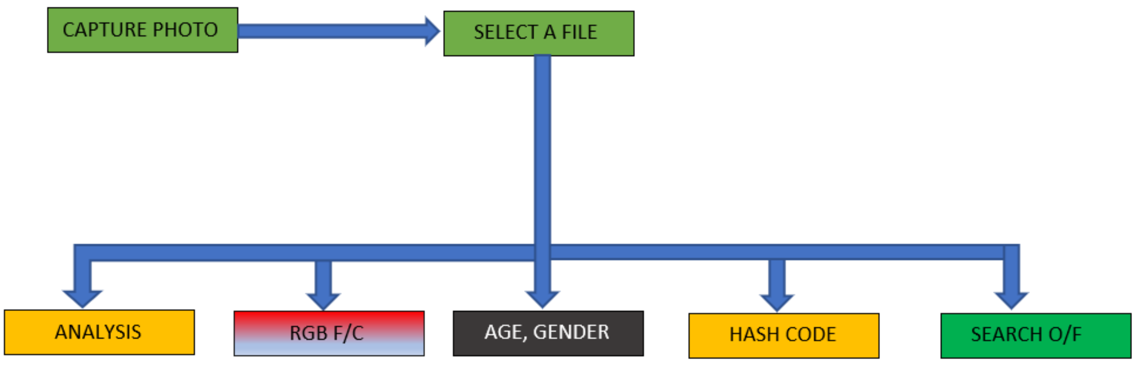
- Now for age and gender detection we use a frame of box to detect a face. so, we use 6 trained models to detect face, age and gender.
  to detect a face, it uses certain image parameters to detect and for age and gender we give some predefined age list and gender difference along with the trained models.

- For getting the hash code of the image we use python library called image hash. so, it works on examining the contents of the image which constructs a unique hash value that uniquely identifies an input image based on the contents of an image. with this hash code in our hands, we can compare the other images and get the highly identical images. which can help with finds the duplicates of the images.
- This process can be done on images which are on the websites. for getting images from the website, we use the selenium library to open a browser and we give certain commands to auto execute the process of getting images. so, we can search image in two ways either using an image itself with help of google image search engine or using pre build search box where we enter the types of object names or a person name etc.

- For image compression we use the PIL library and certain commands on adjusting the resolution values of the image. for locating the location of the image, we use the exit data of the image and this feature opens google maps using selenium and automates the process of locating the image on the google maps.

Implementation-

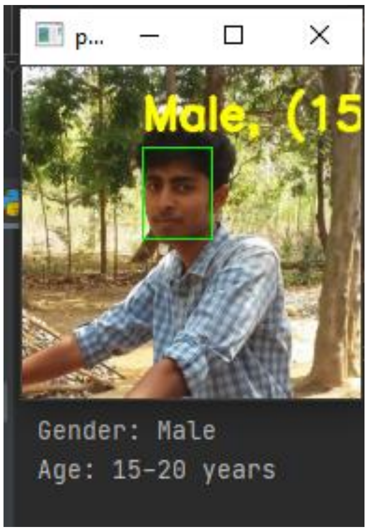
Here's the code we used to achieve our objective



CODE FOR AGE AND GENDER DETECTION:

```
def age_gender_detector(frame):
   t = time.time()
   resultImg,faceBoxes=highlightFace(faceNet,frame)
   for faceBox in faceBoxes:
      face = frame[max(0,faceBox[1]-padding):
              min(faceBox[3]+padding,frame.shape[0]-1),max(0,faceBox[0]-padding)
              :min(faceBox[2]+padding, frame.shape[1]-1)]
      blob = cv2.dnn.blobFromImage(face, 2.0, (227,227), MODEL_MEAN_VALUES, swapRB=False)
      genderNet.setInput(blob)
      genderPreds = genderNet.forward()
      gender = genderList[genderPreds[0].argmax()]
      print(f'Gender: {gender}')
```

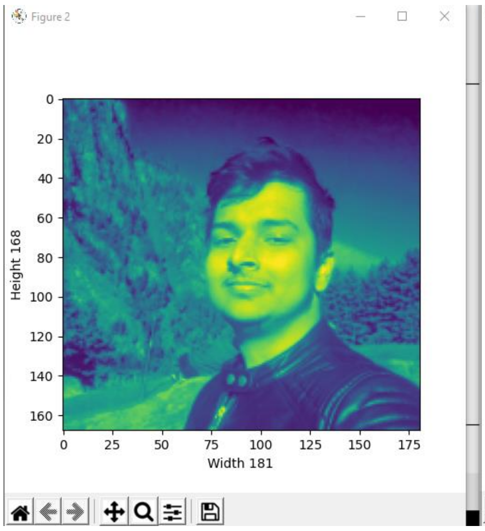
```
        ageNet.setInput(blob)


        agePreds = ageNet.forward()

        age = ageList[agePreds[0].argmax()]

        print(f'Age: {age[1:-1]} years')

        cv2.putText(resultImg, f'{gender}, {age}', (faceBox[0], faceBox[1]-10),
cv2.FONT_HERSHEY_SIMPLEX, 0.47, (0,255,255), 1, cv2.LINE_AA)


    return resultImg
```

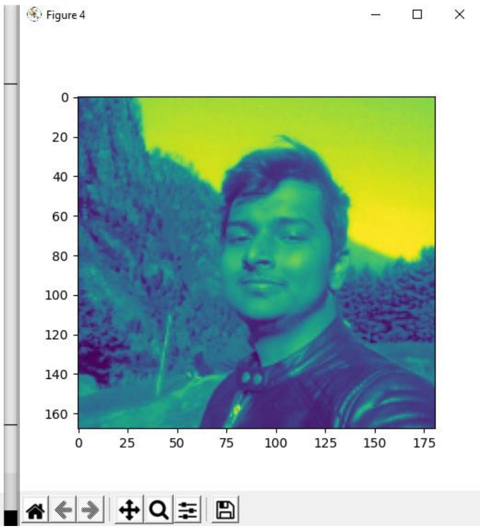


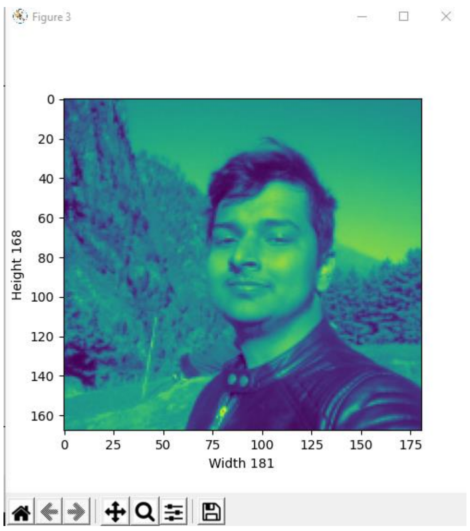
```
Gender: Male
Age: 15-20 years
```

CODE FOR IMAGE ANALYSIS:

```
def dotty():
  filo = open('imageproperties.txt', "w")
    text = ('Type of the image : \n', type(pic))
    text1 = ('Shape of the image : {} \n'.format(pic.shape))
    text2 = ('Image Height {} \n'.format(pic.shape[0]))
    text3 = ('Image Width {} \n'.format(pic.shape[1]))
    text4 = ('Dimension of Image {} \n'.format(pic.ndim))
    text5 = ('Image size {} \n'.format(pic.size))
    text6 = ('Maximum RGB value in this image {} \n'.format(pic.max()))
    text7 = ('Minimum RGB value in this image {} \n'.format(pic.min()))
    filo.writelines(str(text))
    filo.writelines(str(text1))
    filo.writelines(str(text2))
    filo.writelines(str(text3))
    filo.writelines(str(text4))
    filo.writelines(str(text5))
    filo.writelines(str(text6))
    filo.writelines(str(text7))

    webbrowser.open("imageproperties.txt")

    image = Image.open(file)
    exifdata = image.getexif()

    for tagid in exifdata:
        # getting the tag name instead of tag id
        tagname = TAGS.get(tagid, tagid)

        # passing the tagid to get its respective value
        value = exifdata.get(tagid)

        # printing the final result
        text8 = f"{tagname:25}: {value}"

        filo.writelines(str(text8))



def image_analysis():
    #print('IMAGE ANALYSIS?-y/n')
    #if 'y' in input():

    plt.title('Your Image')
    plt.figure(figsize=(5, 5))
    plt.imshow(pic)
    #plt.show()

    for c in zip(range(3)):
        split_img = np.zeros(pic.shape, dtype="uint8")
        split_img[0:255, :, c] = pic[:, :, c]
        plt.figure(figsize=(5, 5))
        plt.imshow(split_img)

    #plt.show()

    return plt.show()

def rgb_channel():

    for c in zip(range(3)):
        split_img = pic[:, :, c]
        plt.ylabel('Height {}'.format(pic.shape[0]))
        plt.xlabel('Width {}'.format(pic.shape[1]))
        plt.figure(figsize=(5, 5))
        plt.imshow(split_img)
    return plt.show()
```



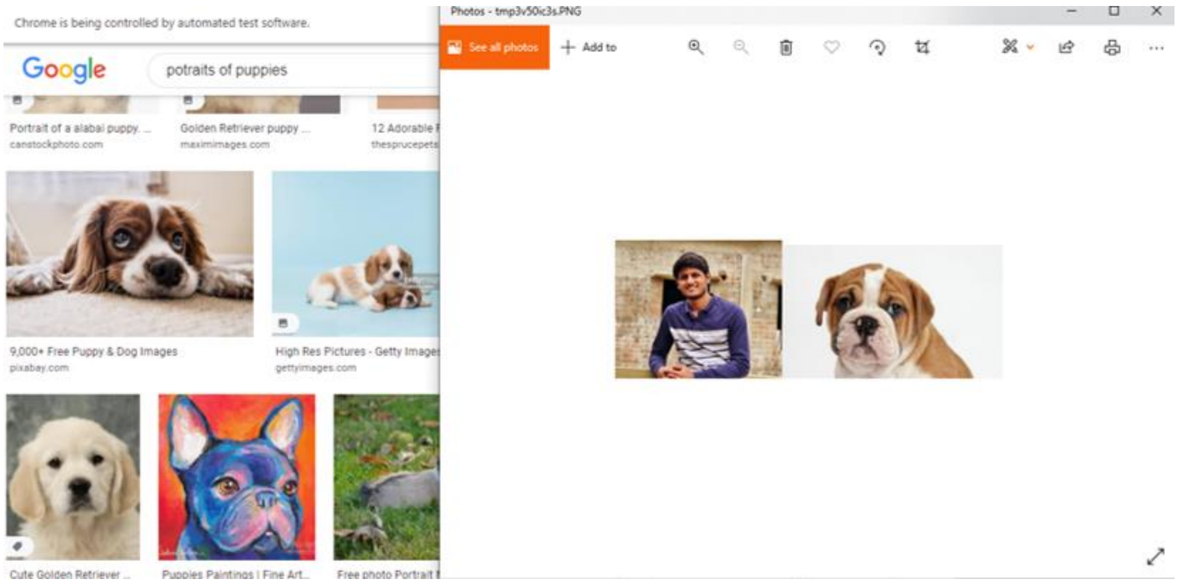




CODE TO MATCH ONLINE:

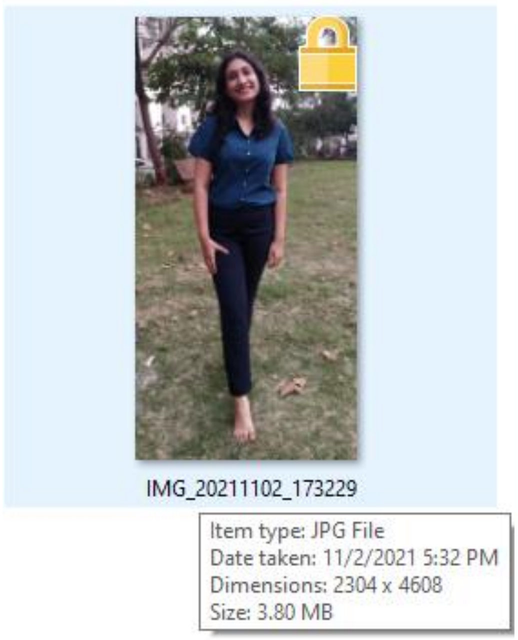
```
def age_gender_detector(frame):
   t = time.time()
   resultImg,faceBoxes=highlightFace(faceNet,frame)


   for faceBox in faceBoxes:
      face = frame[max(0,faceBox[1]-padding):
              min(faceBox[3]+padding,frame.shape[0]-1),max(0,faceBox[0]-padding)
              :min(faceBox[2]+padding, frame.shape[1]-1)]


      blob = cv2.dnn.blobFromImage(face, 2.0, (227,227), MODEL_MEAN_VALUES, swapRB=False)
      genderNet.setInput(blob)
      genderPreds = genderNet.forward()
```

```
        gender = genderList[genderPreds[0].argmax()]

        print(f'Gender: {gender}')

        ageNet.setInput(blob)

        agePreds = ageNet.forward()

        age = ageList[agePreds[0].argmax()]

        print(f'Age: {age[1:-1]} years')

        cv2.putText(resultImg, f'{gender}, {age}', (faceBox[0], faceBox[1]-10),
cv2.FONT_HERSHEY_SIMPLEX, 0.47, (0,255,255), 1, cv2.LINE_AA)

return resultImg
```


Chrome is being controlled by automated test software.
Google
potraits of puppies
Portrait of a alabai puppy. ...
canstockphoto.com
Golden Retriever puppy ...
maximimages.com
12 Adorable
thesprucepets
9,000+ Free Puppy & Dog Images
pixabay.com
High Res Pictures - Getty Images
gettyimages.com
Cute Golden Retriever ...
Puppies Paintings | Fine Art...
Free photo Portrait
Photos - tmp3v50ic3s.PNG
See all photos
Add to

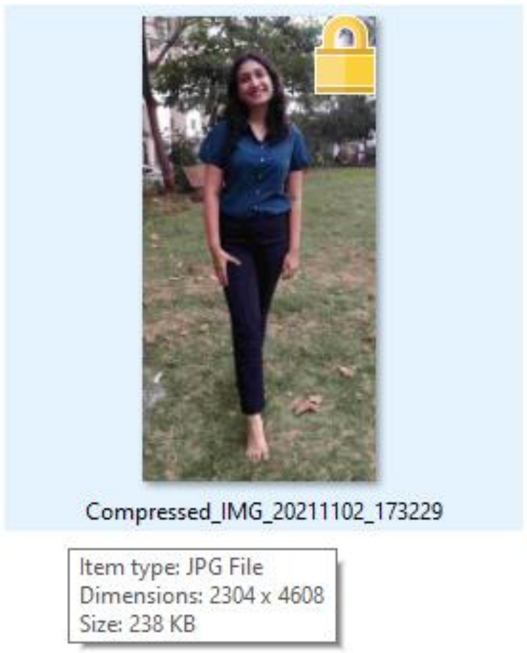
CODE FOR COMPRESSION:

```
def compross():


    verbose = False


    if(len(sys.argv)>1):

        if(sys.argv[1].lower() == "-v"):

            verbose = True

    cwd = './boys/'

    formats = ('.jpg', '.jpeg')


    for file in os.listdir(cwd):
        if os.path.splitext(file)[1].lower() in formats:
            print('compressing', file)
            compressMe(file, verbose)


    print("Done")
    for file in os.listdir(cwd):
        os.remove(cwd+file)
    return file
```



IMG_20211102_173229

Item type: JPG File
Date taken: 11/2/2021 5:32 PM
Dimensions: 2304 x 4608
Size: 3.80 MB

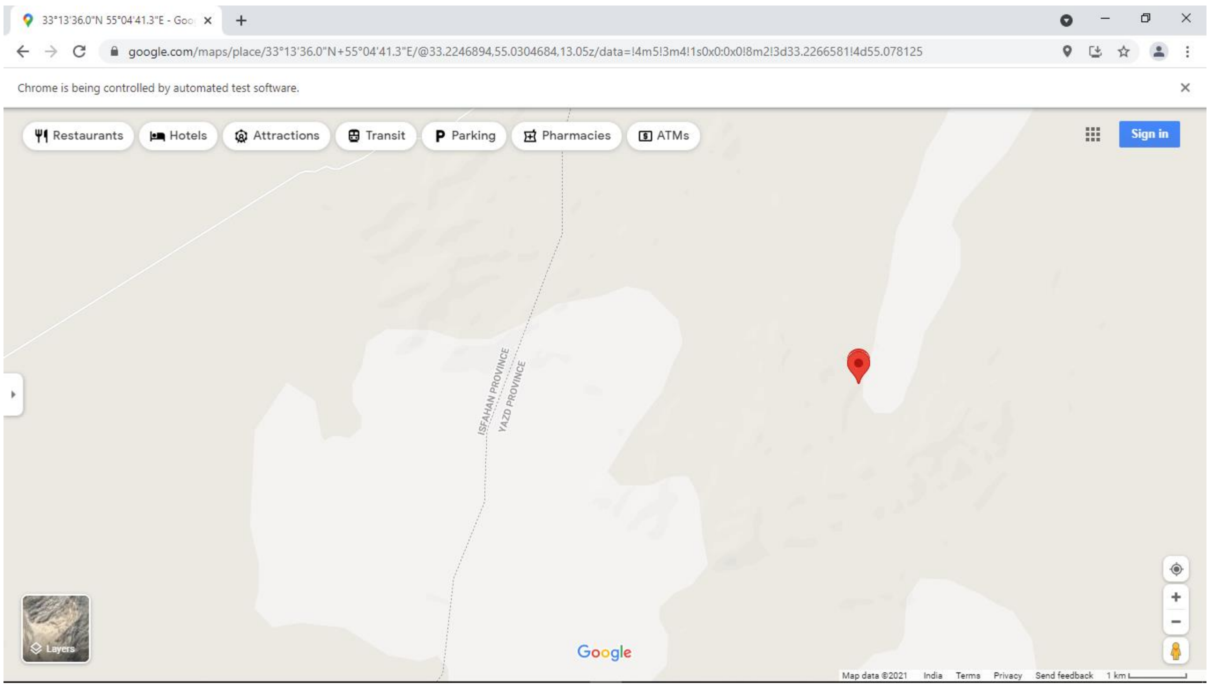


Compressed_IMG_20211102_173229

Item type: JPG File
Dimensions: 2304 x 4608
Size: 238 KB

FOR MAPS –

```
def MyMap():
    Tk().withdraw()
    file = tk.StringVar()
    file = askopenfilename(initialdir=file, filetypes=[("jpg files", "*.jpg")])
    img = exifread.process_file(open(str(file), 'rb'))
    chrome_options = webdriver.ChromeOptions()
    chrome_options.add_argument("headless")
    driver = webdriver.Chrome(executable_path="C:/chromedriver.exe")
    driver.get('https://www.google.com/maps/@13.6066213,79.4141105,15z')
    latitude = format_lati_long(str(img['GPS GPSLatitude']))
    jet = str(latitude)
    # print(latitude)
    longitude = format_lati_long(str(img['GPS GPSLongitude']))
    det = str(longitude)
    entr = (jet+', '+det)
    ser = driver.find_element_by_xpath('//*[@id="searchboxinput"]')
    ser.send_keys(entr)
    ser.send_keys(Keys.ENTER)
    time.sleep(9)
    pas =
driver.find_element_by_xpath('//*[@id="pane"]/div/div[1]/div/div/div[7]/div/div[1]/span[3]')
    location = pas.location
    size = pas.size
    driver.save_screenshot('Mapscr.png')
    tex = get_text(location, size)
    print(tex)
    exi = driver.find_element_by_xpath('//*[@id="sb_cb50"]')
    exi.click()
    ser.send_keys('Temples near'+tex)
    #time.sleep(280)
  return driver
```


33°13'36.0"N 55°04'41.3"E - Goo
google.com/maps/place/33°13'36.0"N+55°04'41.3"E/@33.2246894,55.0304684,13.05z/data=!4m5!3m4!1s0x0:0x0!8m2!3d33.2266581!4d55.078125
Chrome is being controlled by automated test software.
Restaurants
Hotels
Attractions
Transit
Parking
Pharmacies
ATMs
Sign in
ISFAHAN PROVINCE
YAZD PROVINCE
Layers
Google
Map data ©2021
India
Terms
Privacy
Send feedback
1 km

RESULTS –

We successfully processed images and used it to perform analysis, find RGB elements, found age and gender, got the HASH code, and made its location searchable online and offline, Matched the photos online and offline and we even compressed the image using this application. So we conclude that this projects is application with bundle of image processing features

References

- https://pypi.org/project/opencv-python/
- https://www.geeksforgeeks.org/selenium-python-tutorial/#:~:text=Selenium%20is%20a%20powerful%20tool,will%20be%20working%20with%20Python.
- https://docs.python.org/3/library/tkinter.html#:~:text=The%20tkinter%20package%20(%E2%80%9CTk%20interface,well%20as%20on%20Windows%20systems.