

# IMAGE ANALYSIS AND PROCESSING USING PYTHON AND SELENIUM

REVIEW - 2

BY:

18BEC0920 - K J DEEPAK SOMESH

18BEC0906 - SRIJAN VARMA

18BEC0516- PALAK TRIPATHI

18BEC0907- RAMANA ANAND S

# ABSTRACT

A

This project is based around images. Where you get to read the image and match the images. It's 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 go 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 to automate a particular website for advanced predictions



# INTRODUCTION

I

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.

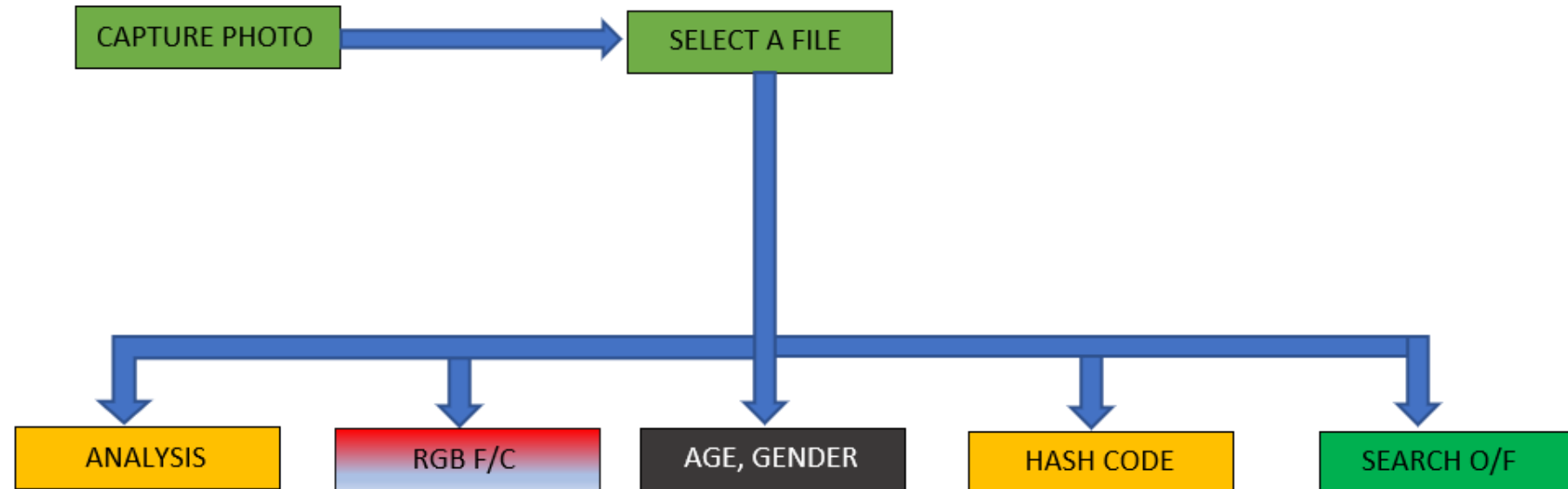
In this project we are going to create an separate app using python so that, we can run this project without help of an IDE.

# What is an Open CV?

- OpenCV is a **huge open-source library for computer vision, machine learning, and image processing**. OpenCV supports a wide variety of programming languages like Python, C++, Java, etc. It can process images and videos to identify objects, faces, or even the handwriting of a human.
- In our project we use Python as our programming language.
- **How does an OpenCV work?**
- OpenCV is a cross-platform library using which we can develop real-time computer vision applications. It mainly focuses on image processing, video capture and analysis including features like face detection and object detection.

# IMPLEMENTATION

B





# CAPTURING PHOTO

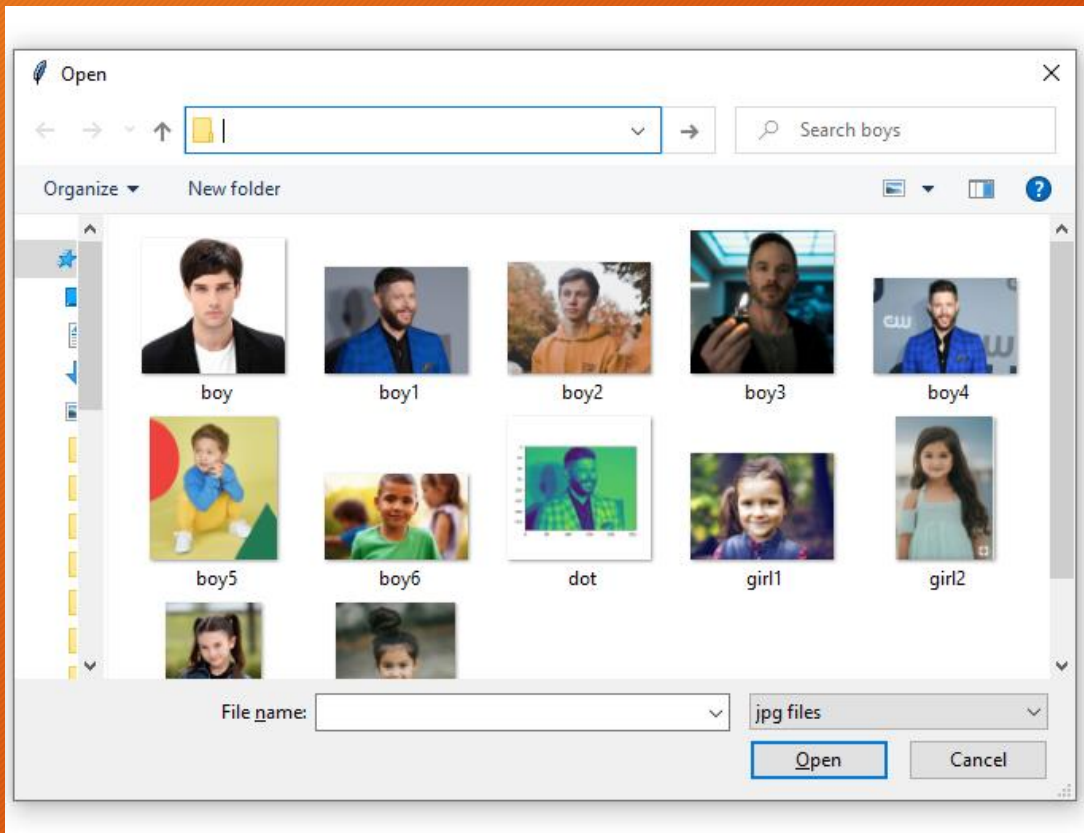
1

When we run the application we get a pop of image capturing window to take a snap to analyze the photo. Now this works on OpenCV module. Using the basic command we can wake this capturing window. In this step I am going to capture a picture of me and save it in a file. We assign two function keys to operate this window, esc to close and space button for capturing and saving.

We use this command “`cam = VideoCapture(0, cv2.CAP_DSHOW)`”  
To call the cv2 video capture function.

# SELECTING THE FILE

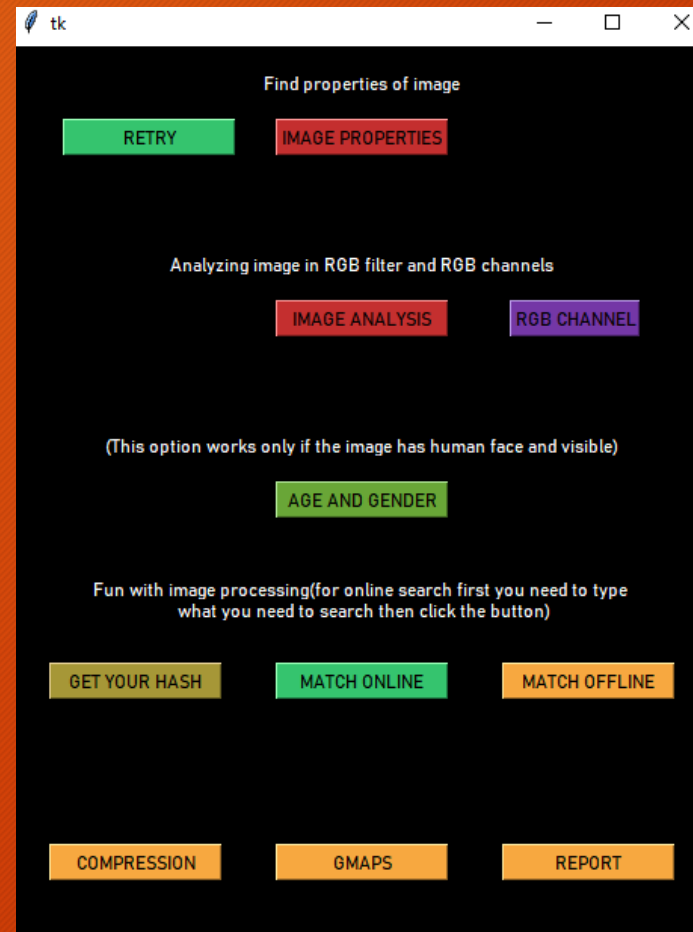
2



This window pops up soon after the capture window is closed, to select the file that you saved or we can also choose other image files as we wish. So after selecting the file click open to write this file inside our application and then we will process this file in our next steps. This feature is created by using tkinter powered by python. So creating this type of window saves time to process image than we manually replace the text in the code.

# APPLICATION

3





# SECTION-1 (IMAGE PROPERTIES)

S1



- Image properties:- After selection of the image we will go and find the image properties by clicking the “IMAGE PROPERTIES” button.
- In this section we will display the properties of the image in a text file. As soon as we press this button we will get a detailed properties of an image.
- Retry:- This button returns to the starting so that you can select different image to process.

# SECTION-2(RGB FILTER AND CHANNEL)

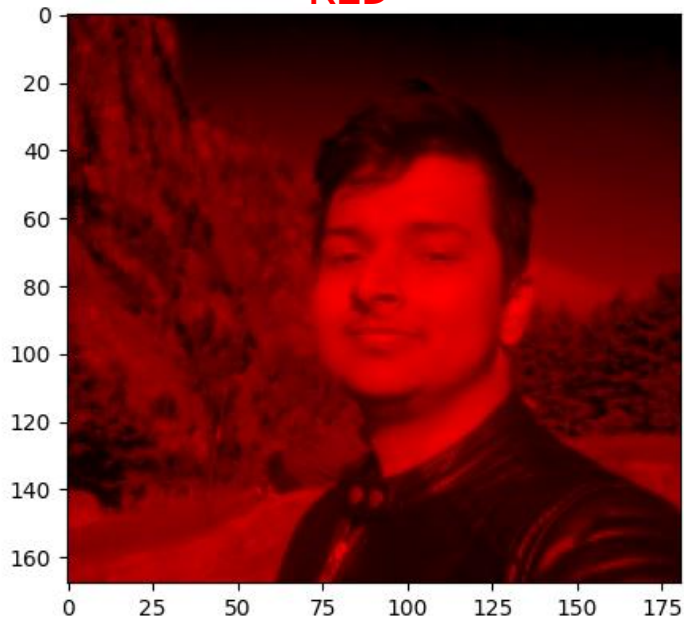
S2

Analyzing image in RGB filter and RGB channels

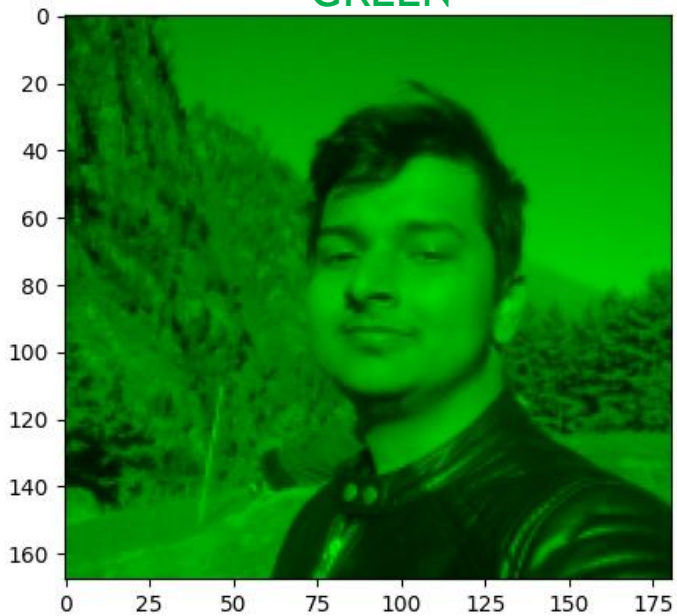
IMAGE ANALYSIS

RGB CHANNEL

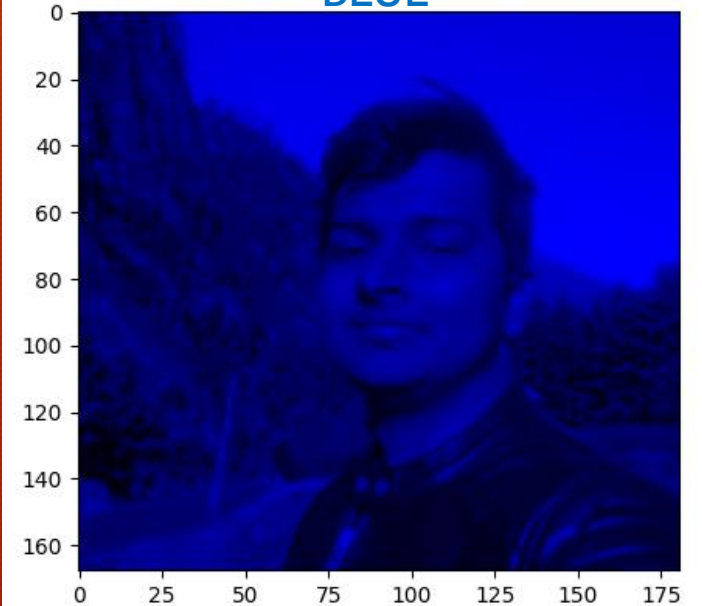
RED



GREEN



BLUE





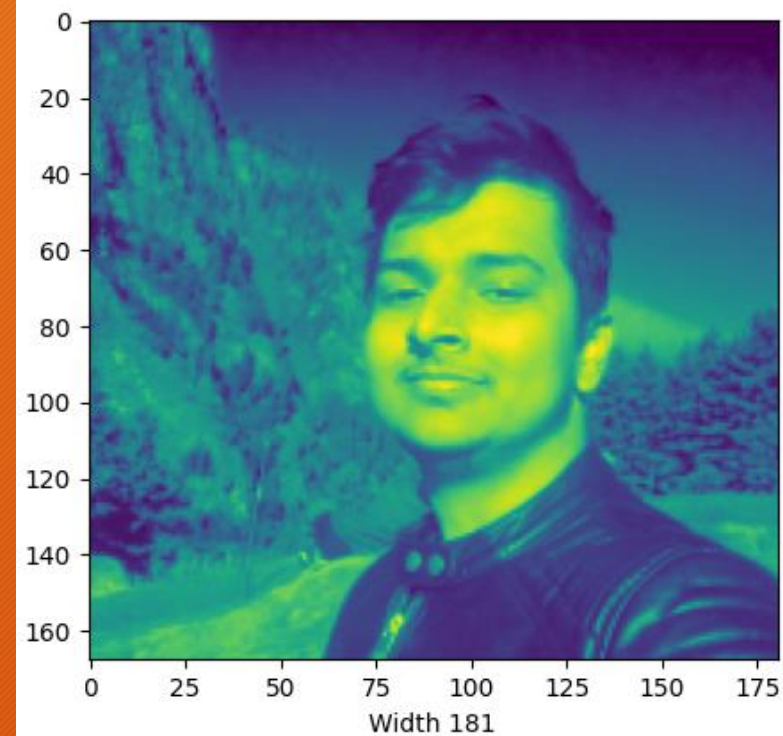
- RGB Filter Analysis:- In this section the image is converted to RGB color model by applying RGB filters(Red, Green, Blue) to it.
- The main purpose of the RGB color model is for the sensing, representation, and display of images in electronic systems, such as televisions and computers, though it has also been used in conventional photography. Before the electronic age, the RGB color model already had a solid theory behind it, based in human perception of colors.
- In this model, every color is defined by three values, one for **red**, one for **green**, and one for **blue**. Usually, these values are 8-bit unsigned integers (a range of 0-255), this is called the **color depth**. The color (0, 0, 0) is **black**, (0, 255, 0) is **green**, (127, 0, 127) is **purple**.
- The color of each pixel is determined by the combination of the red, green, and blue intensities stored in each color plane at the pixel's location. Graphics file formats store RGB images as 24-bit images, where the red, green, and blue components are 8 bits each. This yields a potential of 16 million colors. The precision with which a real-life image can be replicated has led to the commonly used term TrueColor image



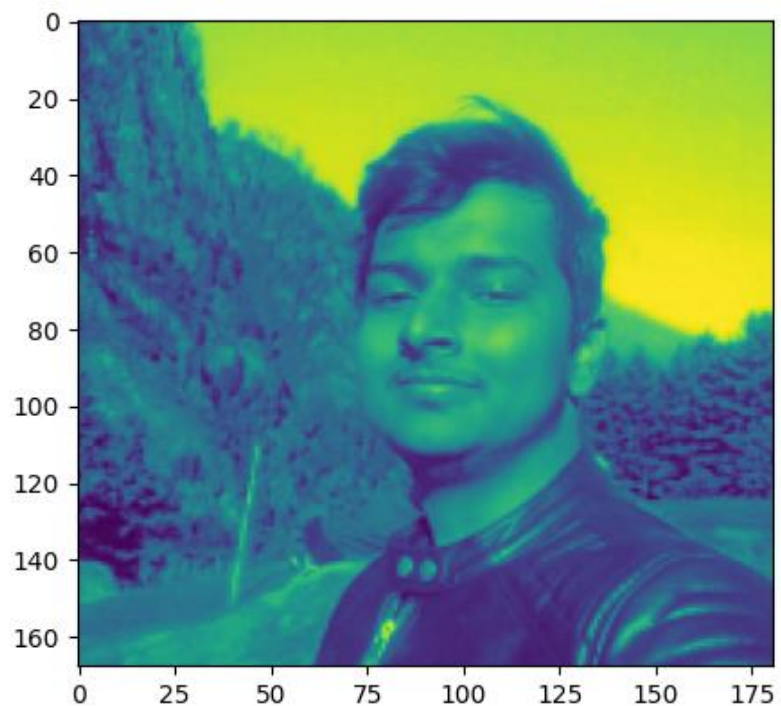
# RGB CHANNEL ANALYSIS

S2

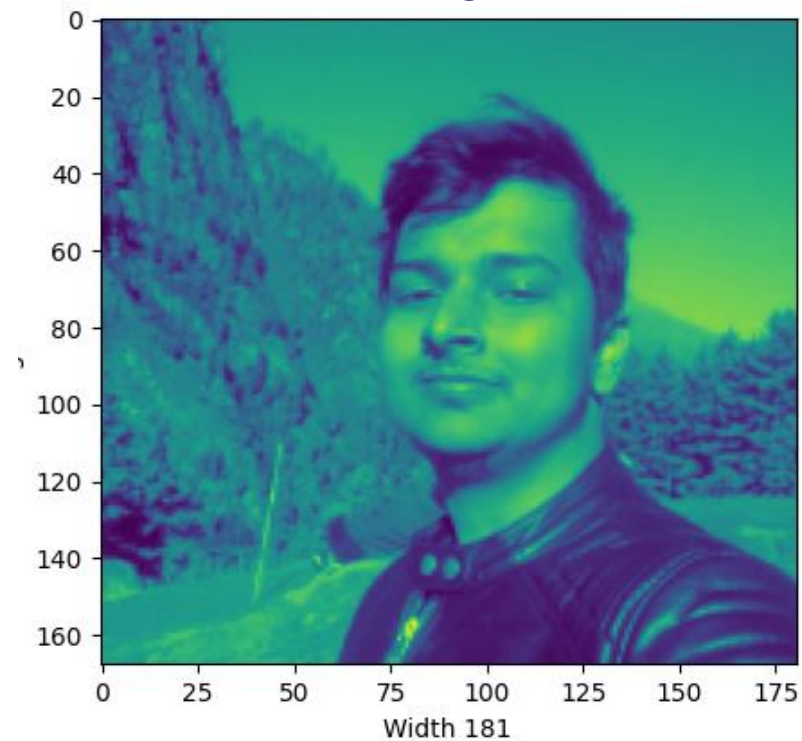
RED



GREEN



BLUE



- RGB Channel Analysis :- An RGB image has three channels: red, green, and blue. RGB channels roughly follow the color receptors in the human eye, and are used in computer displays and image scanners. If the RGB image is 24-bit (the industry standard as of 2005), each channel has 8 bits, for red, green, and blue—in other words, the image is composed of three images (one for each channel), where each image can store discrete pixels with conventional brightness intensities between 0 and 255. If the RGB image is 48-bit (very high color-depth), each channel is made of 16-bit images.
- Notice how the black and grey sides have similar brightness in all channels, the red part of the picture is much brighter in the red channel than in the other two, and how the green part of the picture is shown much brighter in the green channel.



# SECTION-3(AGE AND GENDER)

S3

(This option works only if the image has human face and visible)

AGE AND GENDER



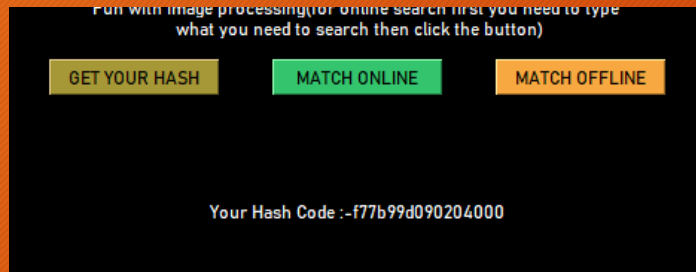
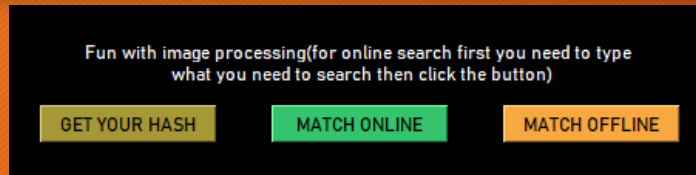
- Age and Gender Detection :- In this section we are going to detect the age and gender of person(if included in image). To implement this we need pre-trained models to detect the face and predict the age and gender.
- The pre trained models we used are
- For age:- “age\_deploy.prototxt”, “age\_net.caffemodel”.
- For gender:- “gender\_deploy.prototxt”, “gender\_net.caffemodel”.
- For face:- “opencv\_face\_detector.pbtxt”, “opencv\_face\_detector\_uint8.pb”.
- We insert this files and set a constant resize values for image. Then we print and display the age and gender detected by the use of face frame which is created.



- Ideally, Age Prediction should be approached as a Regression problem since we are expecting a real number as the output. However, estimating age accurately using regression is challenging.
- The Audience dataset has 8 classes divided into the following age groups [(0 - 2), (4 - 6), (8 - 12), (15 - 20), (25 - 32), (38 - 43), (48 - 53), (60 - 100)]. Thus, the age prediction network has 8 nodes in the final SoftMax layer indicating the mentioned age ranges.
- It should be kept in mind that Age prediction from a single image is not a very easy problem to solve as the **perceived age** depends on a lot of factors and people of the same age may look pretty different in various parts of the world. *Also, people try very hard to hide their real age!*

# SECTION-4(HASH CODE)

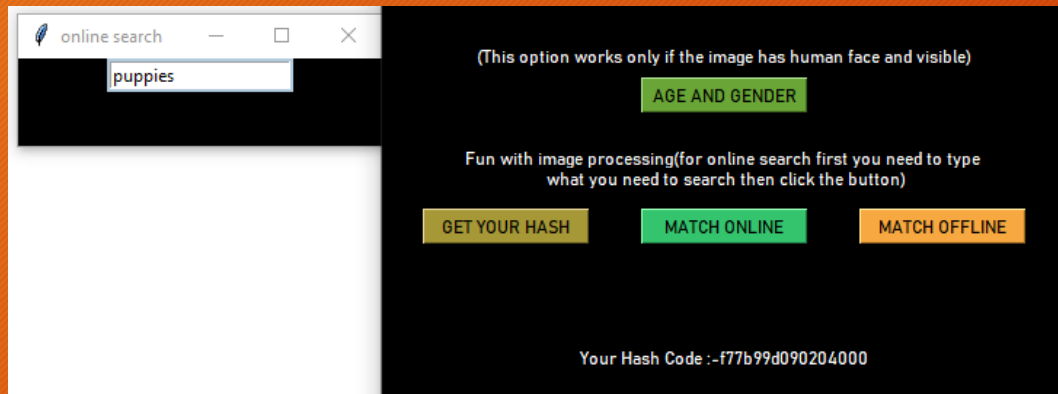
S4



- Generating Hash code:- In this section we use ImageHash library to generate the hash code of the inserted image.
- We first insert our image and then we will generate the hash code for our image through Imagehash.

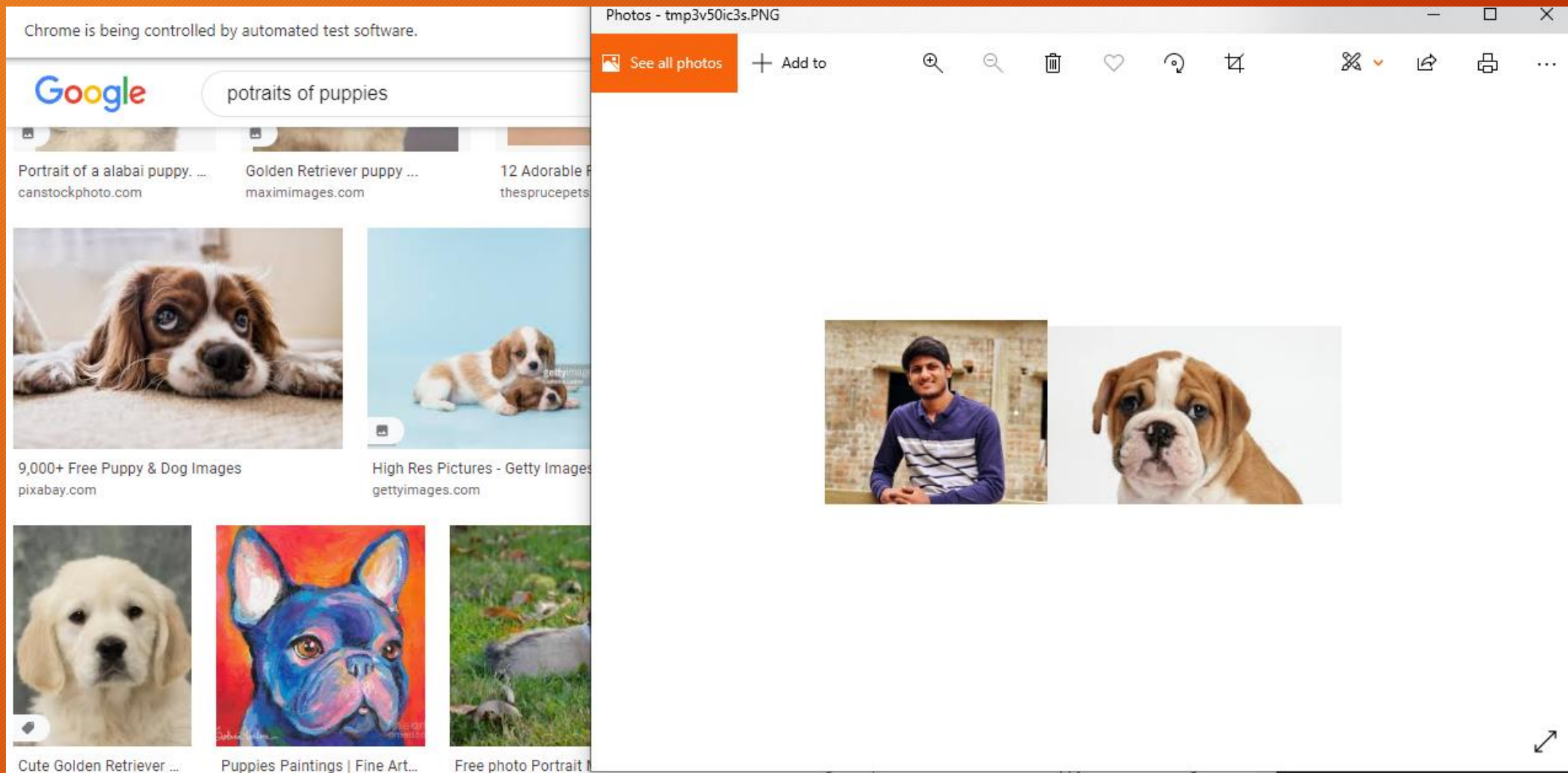
# MATCH ONLINE/OFFLINE

S5



- Matching Online:- After getting hash code for our image, we can match the hash codes of other images to our hash code. Then we have created a search box for online search. We first enter the object name or objects which we want to match with.
- For example we have typed “puppies”. After this we press the “Match online” button then it directs to chrome.





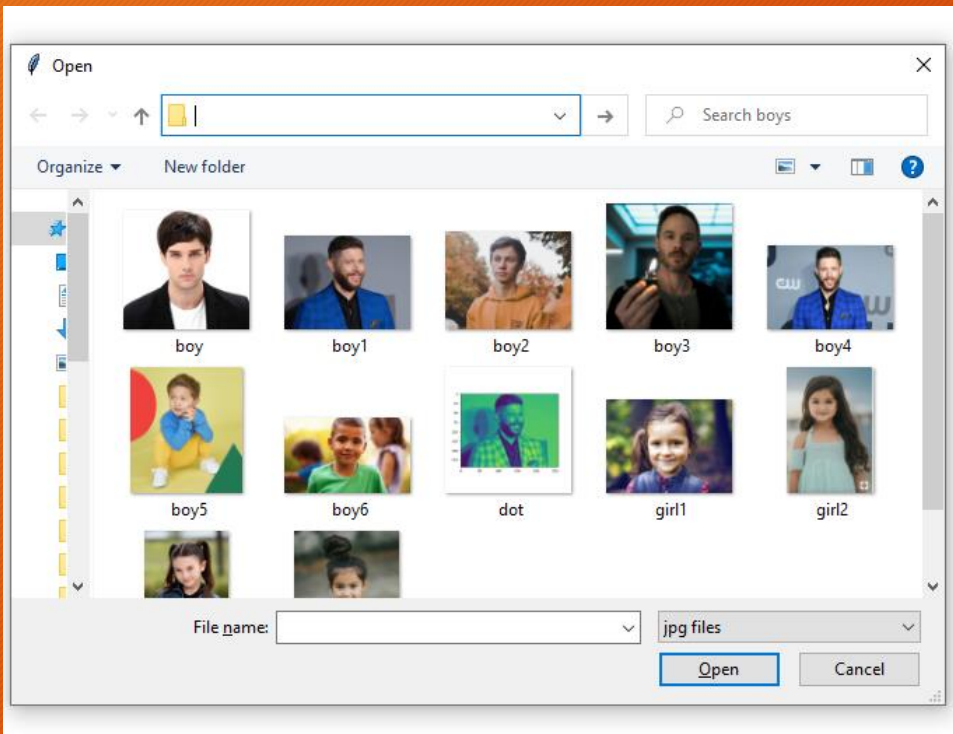
S5

- In this project we have automated the chrome using selenium library. By using the selenium chrome web driver, we search the object in google and it gives a bunch of images of the object that you typed in the search box of the IA application and then it automatically downloads the images until the end of the page and stores it in a accessible folder.
- To automate the chrome we need the Xpaths to control the chrome. So we inspected the chrome search box location and copied the path of it in xpath format. Similarly for the scroll of the page.
- After storing the images in particular folder, we give access to application to read and generate the hash code.
- Then we calculate the difference between each image hash code. The difference between the inserted image and the other image should be less. If that condition satisfies then we get a merged image of inserted and predicted image side by side.



# OFFLINE

S5

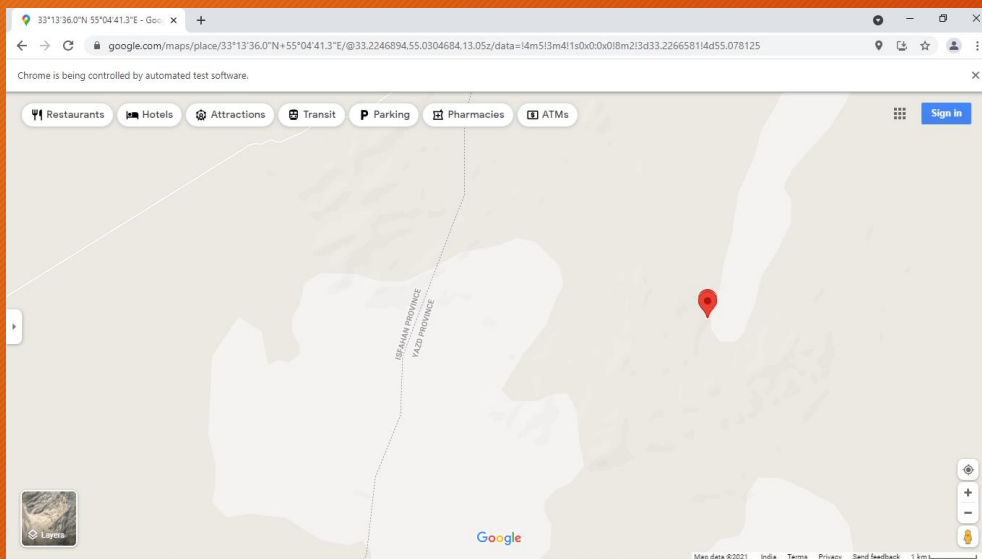


- Match Offline:- After getting hash code for our image, we can match the hash codes of other images to our hash code by selecting the directory folder of images which you want to match.
- Then we calculate the difference between each image hash code with our hash code. The difference between the inserted image and the other image should be less. If that condition satisfies then we get a merged image of inserted and predicted image side by side.



# LOCATING THE IMAGE

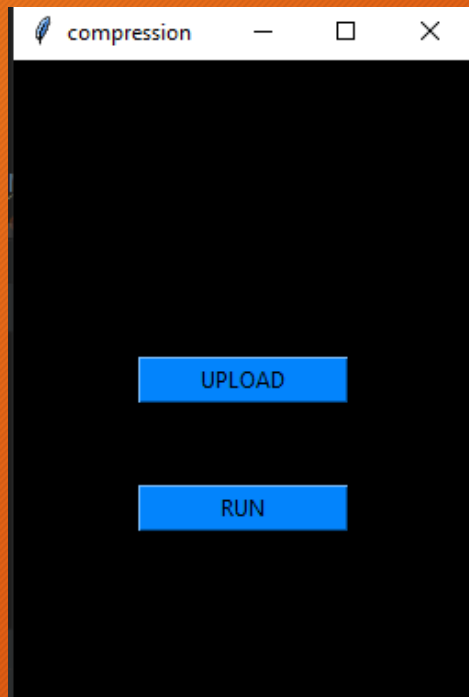
S6



- We take images in different places. After taking too many photos, you could not remember where exactly the you shot.
- Some do not notice that the gps data is stored in images. So our application extracts the gps data in terms of co-ordinates.
- Using those coordinates we use selenium to pin point the location in the maps.

# IMAGE COMPRESSION

S7



- In this part we use PIL library in python and numpy to compress image.
- Now a days images that ranges between 1mb to 6mb file size is pretty common and usage of images also increased time to time. So to reduce the storage and the processing time to upload or access the image can be reduced using image compression.
- In this application you can just upload the image and run, it does the compression for you.



# COMPRESSED IMAGE



IMG\_20211102\_173229

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

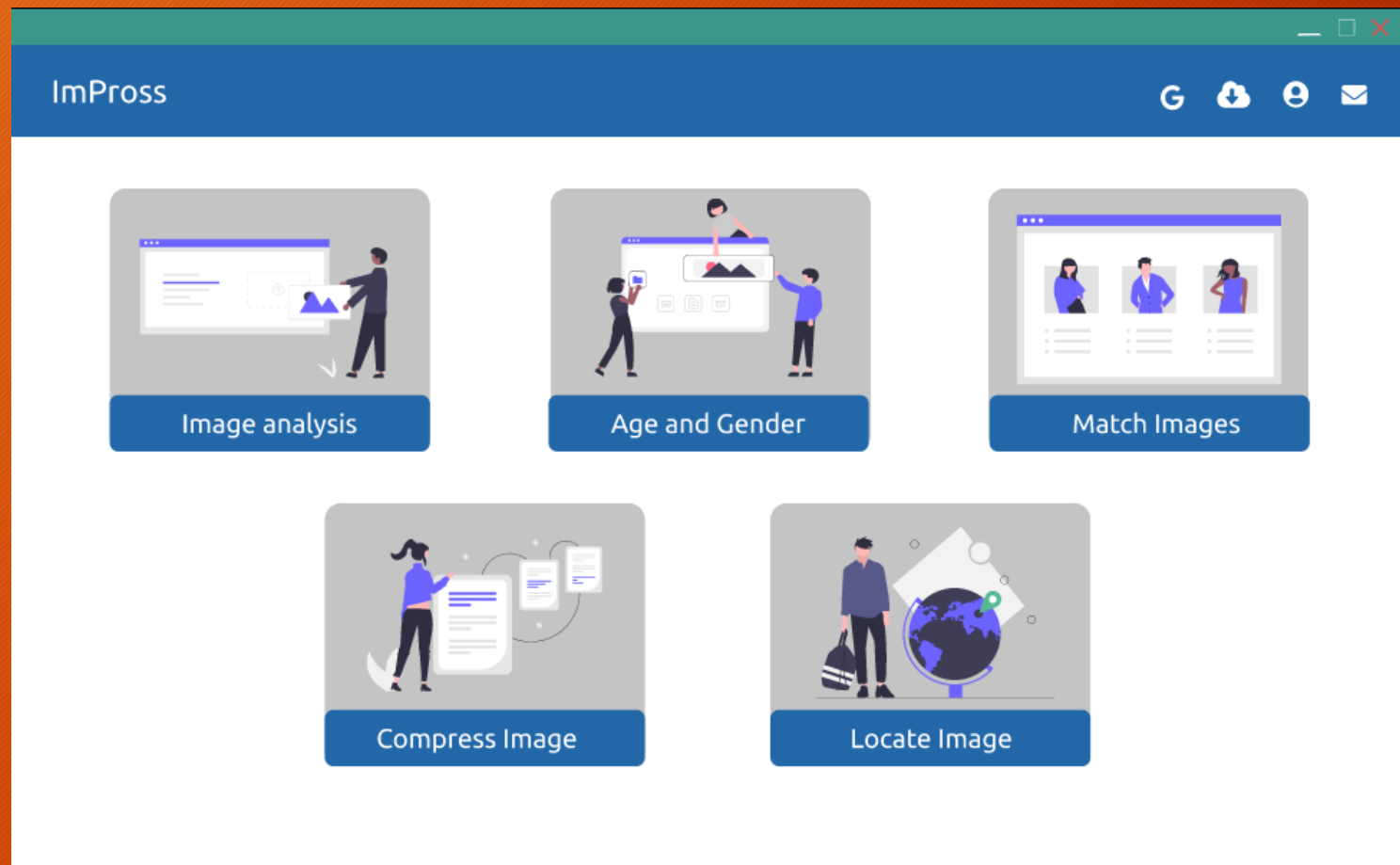


Compressed\_IMG\_20211102\_173229

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

# APPLICATION INTERFACE

UI





# CONCLUSION

C

In this project we have analyzed and processed image in such a way that gives an base data of an image and processing this image into finding the age-gender and finding similar images using hash code.

We have also located the image location and also compressed the image.

For further analysis of the image we can automate the google cloud vision API. which detects objects and more. So, we conclude that this project is a tool to analyze and process images.

# REFERENCES

R

- <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.](https://docs.python.org/3/library/tkinter.html#:~:text=The%20tkinter%20package%20(%E2%80%9CTk%20interface,well%20as%20on%20Windows%20systems.)