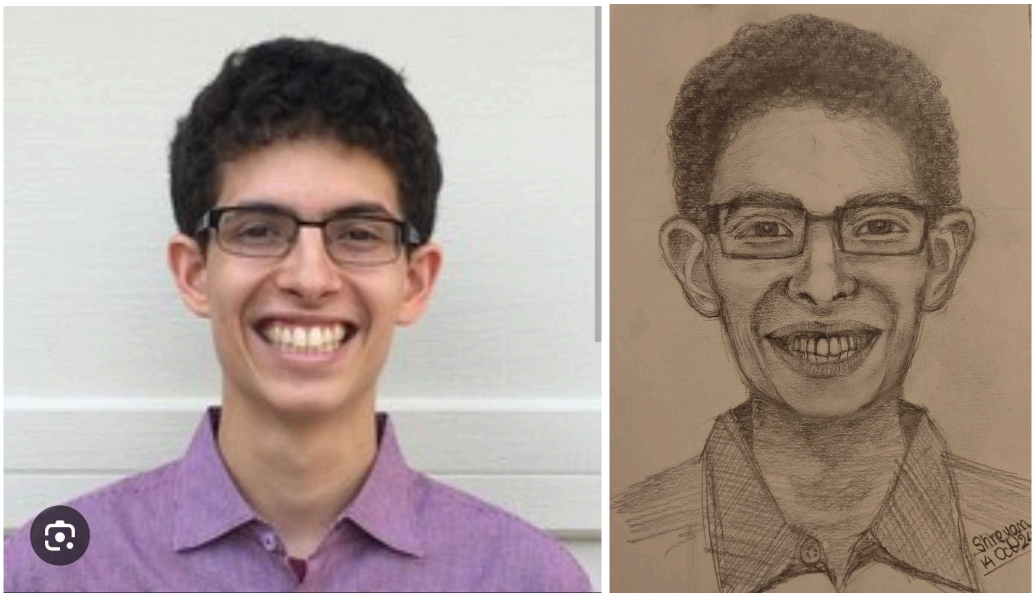
PortrART: Drawing Portraits Made Easy

Motivation: As an artist who loves drawing portraits, I find it hard to find the perfect art software which is catered towards just helping the user work on portraits. Also, there is currently no software on the web which can compare a drawing and a reference picture and tell me how well I drew it (i.e., my accuracy) or give me a drawing of the main features of the portrait.

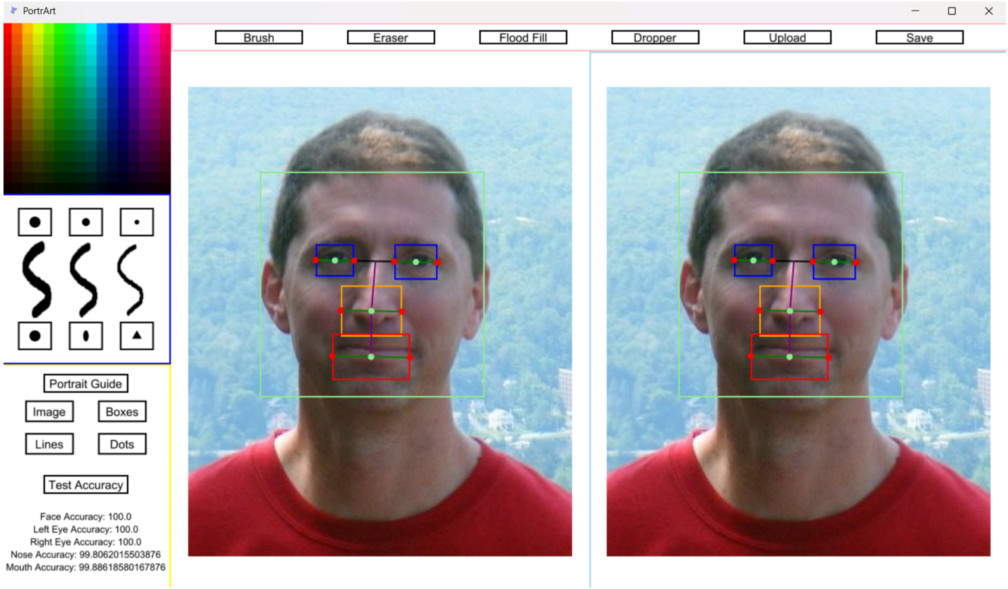
Description: My project is based on creating a completely functional digital art software. This software would be complete with its own tools like different brushes, eraser, flood fill, blur and selection.

Along with being a digital art software, the application will have a special focus on portrait art and try to optimize tools to give the best experience to portrait artists.

It would have features such as getting the loomis method sketch of the portrait by just uploading the reference image. This would be immensely helpful to portrait artists because this would ensure that the proportions of the portrait are perfectly matched but they can add their own personality to the drawing. Another feature would be feature comparison. This would compare the accuracy of the portrait drawn by the artist to the reference image. It would also show which features should be improved to improve the overall accuracy of their drawing.



This is one of the portraits I drew of Professor Schick. Although this was a quick sketch, you can see that not all the proportions of the face are perfect. It would be a lot more perfect if I could somehow get the rough positioning of all the facial features. This is what my art program hopes to achieve.



```
Face Accuracy: 100.0
Left Eye Accuracy: 100.0
Right Eye Accuracy: 100.0
Nose Accuracy: 99.8062015503876
Mouth Accuracy: 99.88618580167876
```

Program complete with all features!

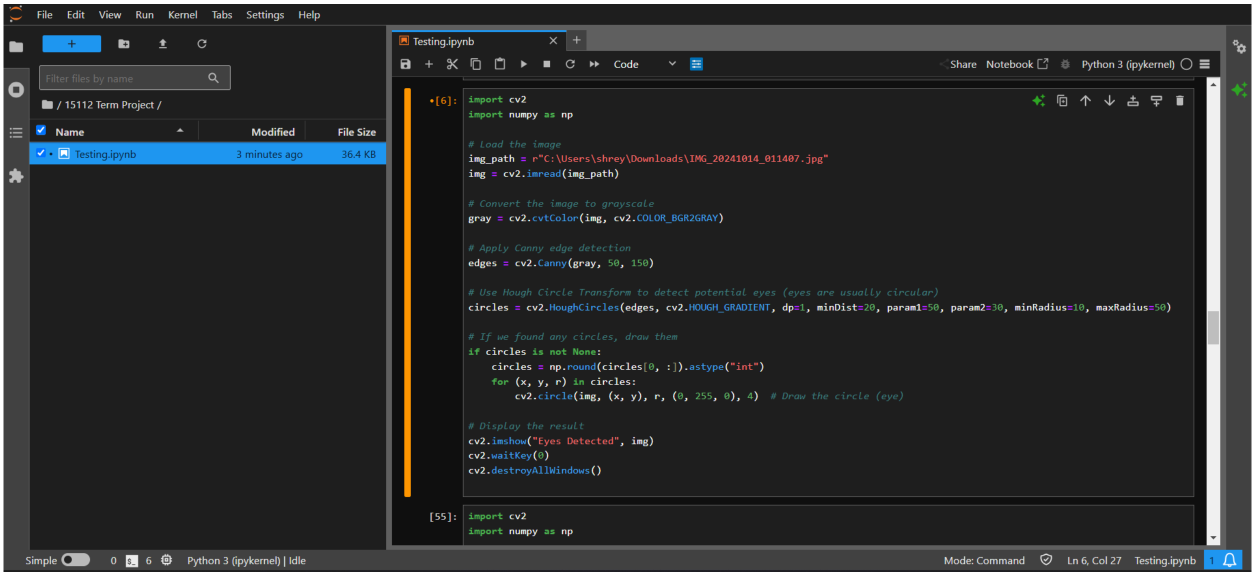
Similar Projects:
A major source of inspiration for me was the digital art softwares I have use, such as:

1. Ibis Paint X
2. Autodesk Sketchbook
3. Rebelle 5

My software will showcase many similar features that these industry level drawing softwares have, such as brushes, flood fill, etc.

Another major portion of the project is facial recognition, and extracting all the facial features. This is similar to how face recognition in general works, and is similar to the project TFaceID, in that it will also require a face detecting algorithm.

Version Control / Backup Plan: I am backing all my work on Anaconda-Jupyter, which stores everything on the cloud, and can be accessed from different devices. I am also regularly saving all my code on a google drive.



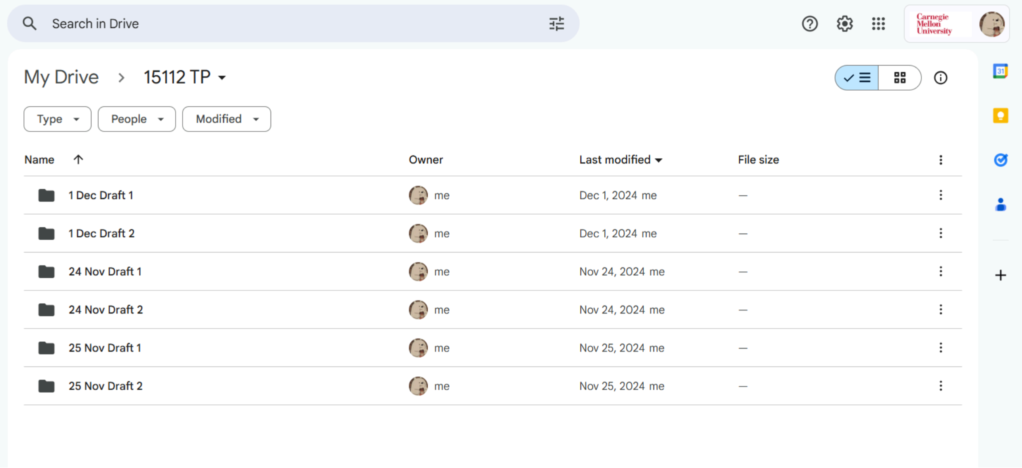
•[6]:

```
import cv2
import numpy as np

# Load the image
img_path = r"C:\Users\shrey\Downloads\IMG_20241014_011407.jpg"
img = cv2.imread(img_path)

# Convert the image to grayscale
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

# Apply Canny edge detection
edges = cv2.Canny(gray, 50, 150)

# Use Hough Circle Transform to detect potential eyes (eyes are usually circular)
circles = cv2.HoughCircles(edges, cv2.HOUGH_GRADIENT, dp=1, minDist=20, param1=50, param2=30, minRadius=10, maxRadius=50)

# If we found any circles, draw them
if circles is not None:
    circles = np.round(circles[0, :]).astype("int")
    for (x, y, r) in circles:
        cv2.circle(img, (x, y), r, (0, 255, 0), 4)  # Draw the circle (eye)

# Display the result
cv2.imshow("Eyes Detected", img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

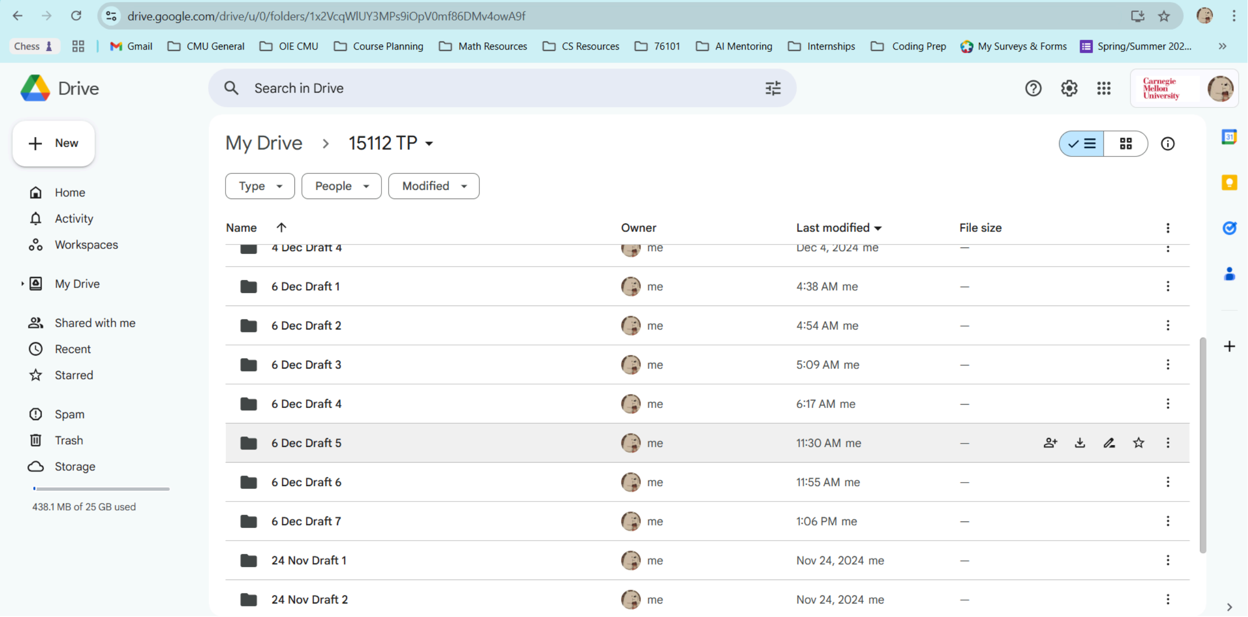
[55]:

```
import cv2
import numpy as np
```

Code saved on Jupyter Notebook



Name	Owner	Last modified	File size
1 Dec Draft 1	me	Dec 1, 2024 me	—
1 Dec Draft 2	me	Dec 1, 2024 me	—
24 Nov Draft 1	me	Nov 24, 2024 me	—
24 Nov Draft 2	me	Nov 24, 2024 me	—
25 Nov Draft 1	me	Nov 25, 2024 me	—
25 Nov Draft 2	me	Nov 25, 2024 me	—


drive.google.com/drive/u/0/folders/1x2VcqWlUY3MPs9iOpV0mf86DMv4owA9f

Name	Owner	Last modified	File size
4 Dec Draft 4	me	Dec 4, 2024 me	—
6 Dec Draft 1	me	4:38 AM me	—
6 Dec Draft 2	me	4:54 AM me	—
6 Dec Draft 3	me	5:09 AM me	—
6 Dec Draft 4	me	6:17 AM me	—
6 Dec Draft 5	me	11:30 AM me	—
6 Dec Draft 6	me	11:55 AM me	—
6 Dec Draft 7	me	1:06 PM me	—
24 Nov Draft 1	me	Nov 24, 2024 me	—
24 Nov Draft 2	me	Nov 24, 2024 me	—

Code saved on Google Drive daily

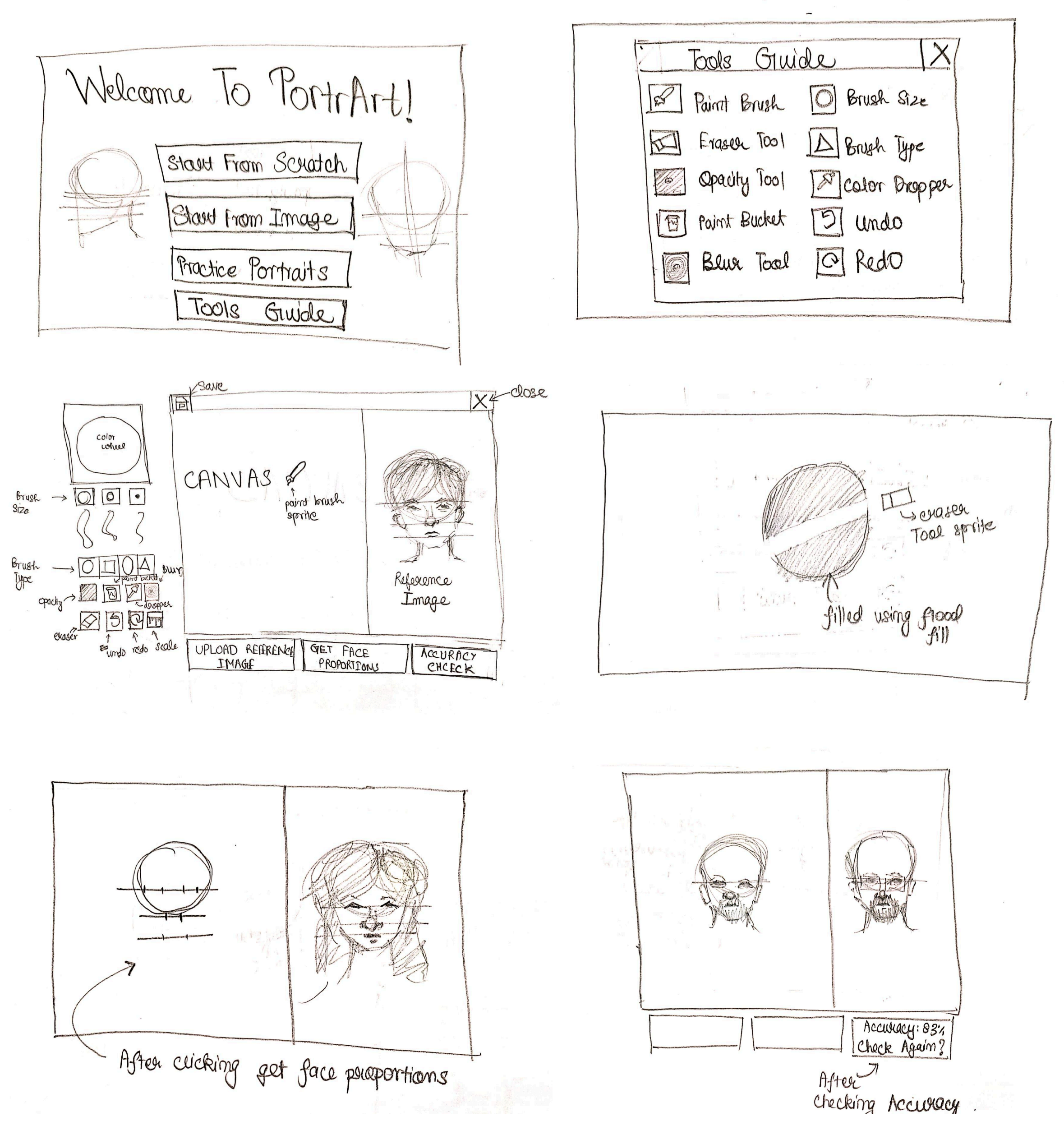
Tech List:

I am using mainly two external libraries:

1. OpenCV- specifically Haar-Cascades
2. Python Pillow (PIL)

In addition, I am also using some external cascades from https://github.com/sightmachine

Storyboard



Left to Right, Top to Bottom: *Starting screen, Tools guide, Drawing Program Layout, Eraser & Paint Bucket Tools, Getting face proportions, Accuracy checker*