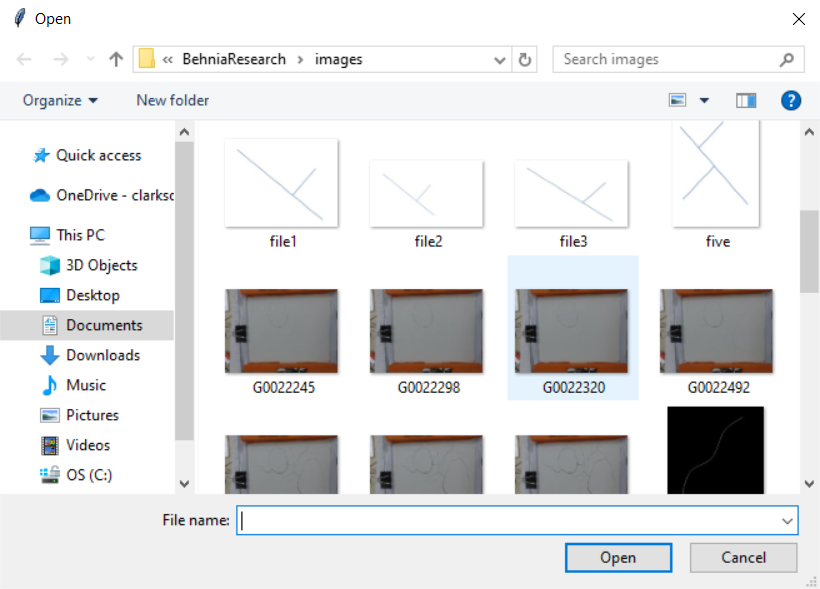
**April Paper Write Up**

Crack Analysis.py

**The Process:**

STEP ONE

Once the program has begun, the user’s file system will pop up, prompting the user to select the image they would like to analyze. This uses the tkinter packages which allows the pop-up file system to open and allow the user to interact with it. The specific package from tkinter is askopenfilename, which allows the user to see the window below.

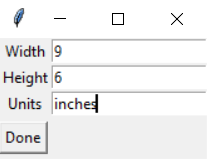


STEP TWO

The image will be resized based on the user’s window size. This will allow very large images to be fully viewed by the user.

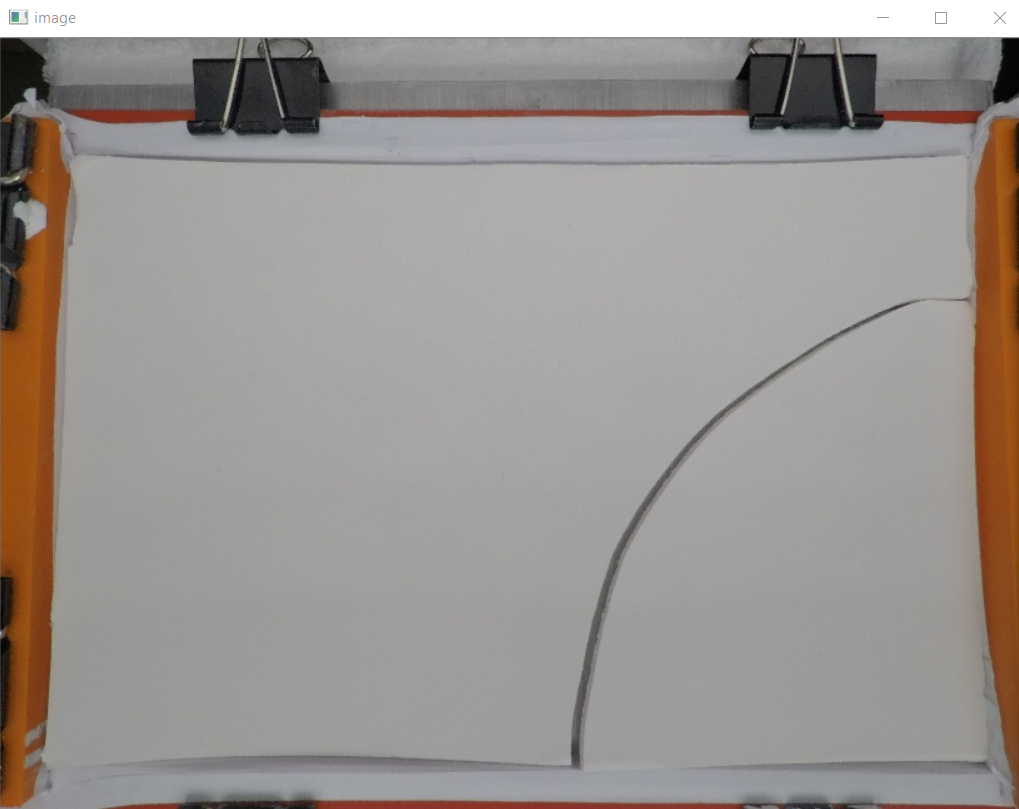
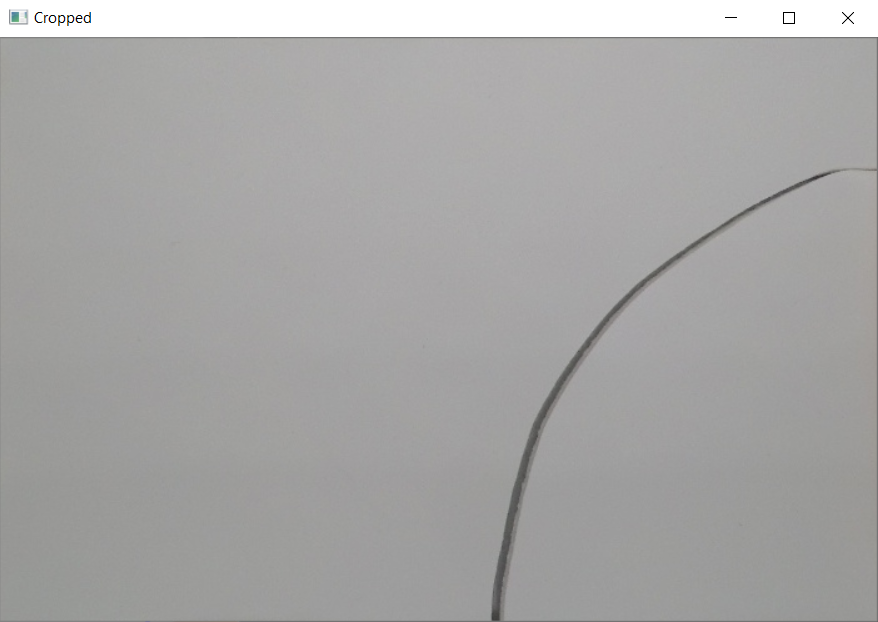
STEP THREE

Using tkinter being the package used for GUI user input for python, a pop-up window will prompt the user to enter in the width, height and units used for the image. This will allow the program to calculate the length of the line and the width accurately. These numbers will allow the program to measure each pixel in terms of the units provided.



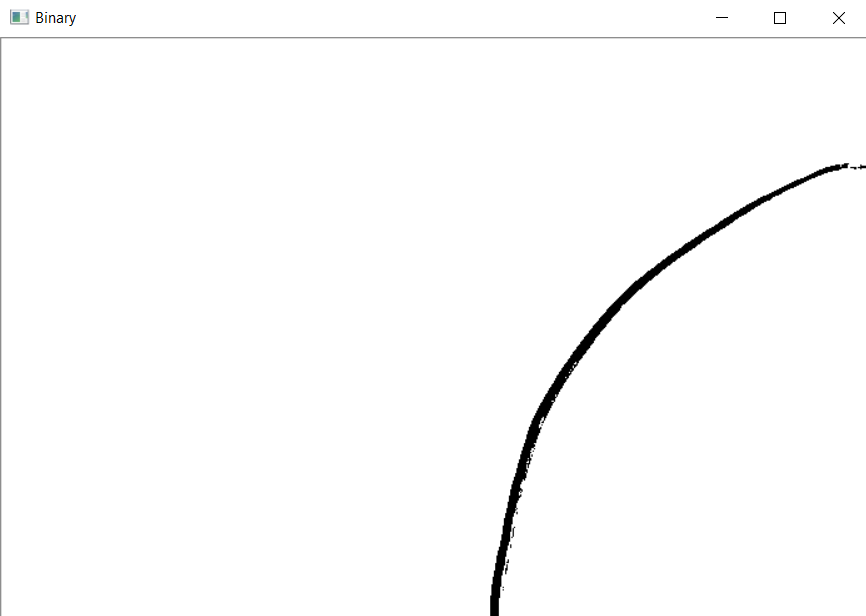
STEP FOUR

The image will be shown to the user. The user will then have the ability to crop the image and remove unnecessary parts of the image by clicking the top left corner and dragging to the bottom right hand corner. When the box has ben created and the user lets go of their mouse, the cropped image will appear on the screen. The program uses the event LBUTTONDOWN to see when the pixel the user originally selects. Once the second pixel is found, these are used as the corners that roi uses the crop the image. If the user is happy with the image not cropped and how is it in its original state, the user will press the Esc button. This button will be used throughout the process to proceed to the next step.

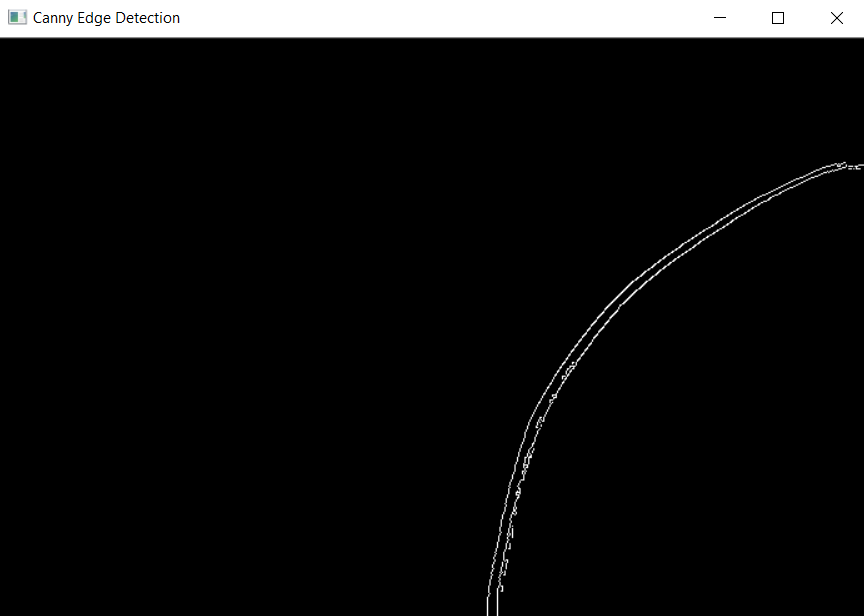
STEP FIVE

The next step in the process is taking the process is the creation of the binary image. This is when the image the user selects is colored either black or white. To get the best results possible for the binary image, a median filter is used to get rid of any extra points in the image not needed. The binary image is then calculated with an adaptive gaussian threshold which will determine what should be colored black and white based on the values in smaller areas of the image.



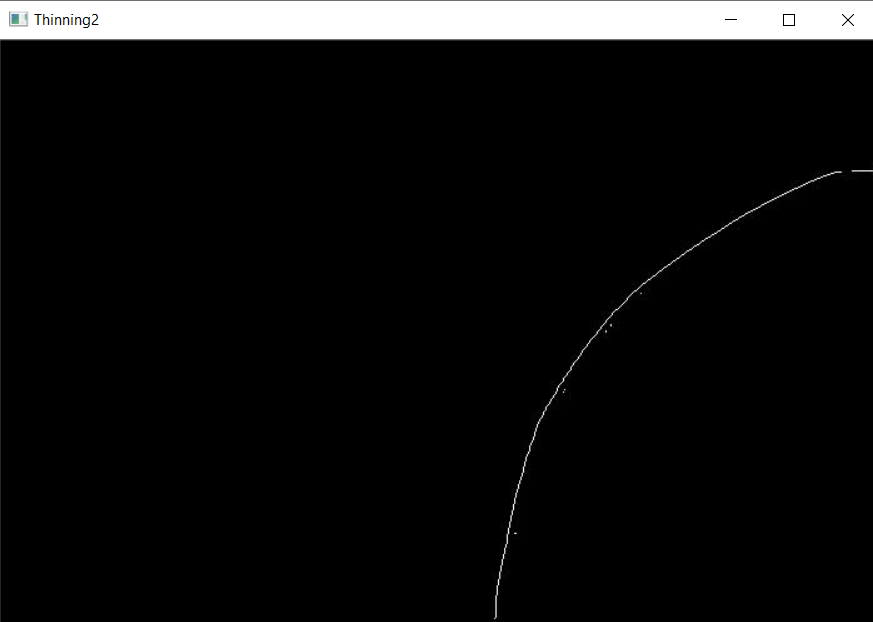
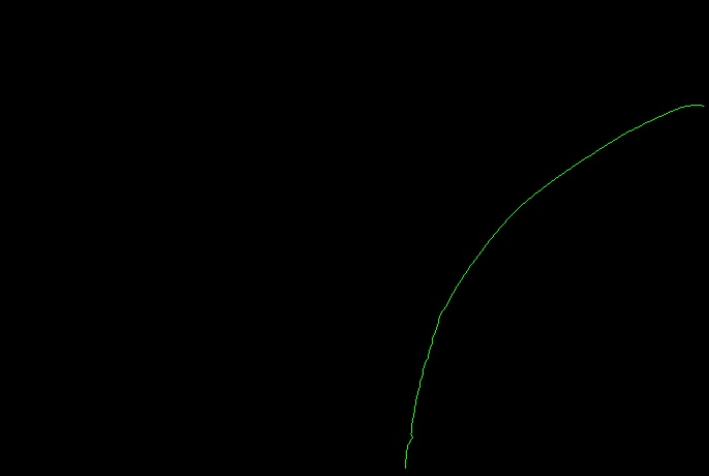
STEP SIX

Canny edge detection is used to find the edges of the photo. This will be important when calculating the width of the image.



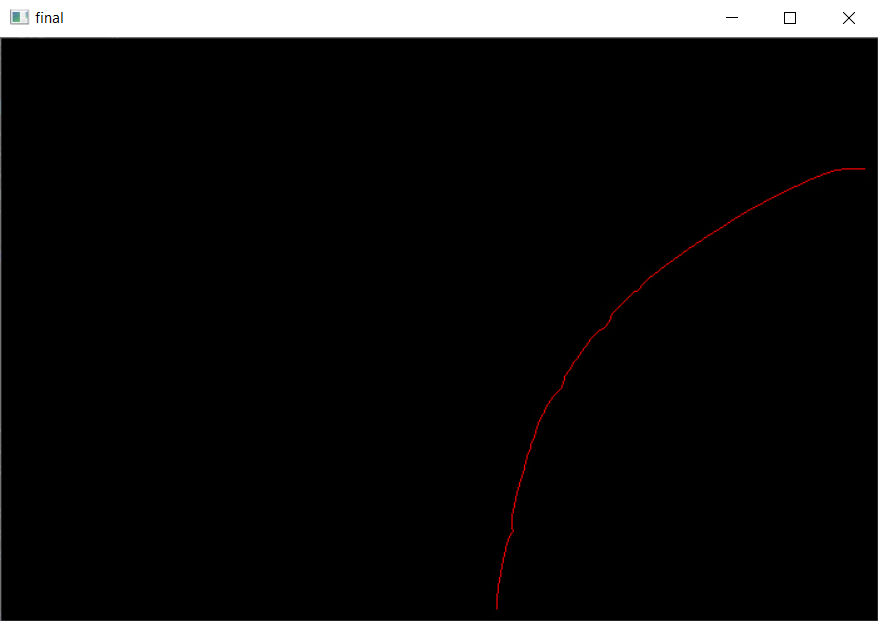
STEP SEVEN

The thinning function takes the binary image and returns a picture of the midline of the image. This skeletonized image will be used for finding the length of the cracking. When the thinned image is created, the complete function will be used to fill in any gaps. This is important because the length calculation only works if the skeleton line is continuous.

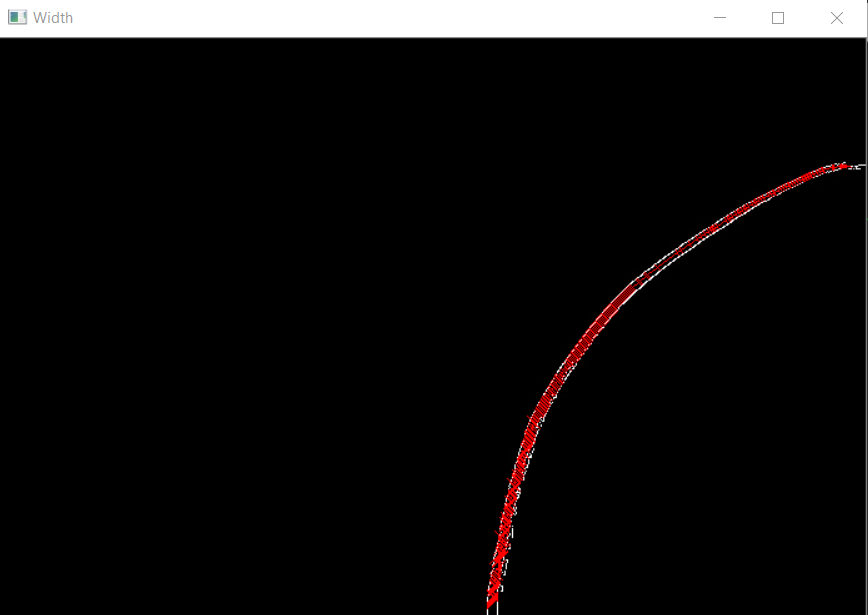
STEP EIGHT

The length function is then called. There are various parts to getting the length. The first step is the program loops through every pixel in the image and will get the red, green and blue values of the pixel. With these values, an if statement can be used to figure out if each pixel is part of the skeleton of the cracking. For each pixel that is part of the skeleton of the cracking, the getColor function is then called. This function takes in the x and y coordinates of the pixel and then will return a function that will include the pixels directly above, below, to the side or diagonal to the selected pixel, along with the number of pixels that fit those conditions. If a pixel, only has one pixel next to it, then it is an endpoint. This pixel will be colored blue. If a pixel has two pixels next to it, it falls in the middle of the line so it will be colored green. Lastly, if there are more than two pixels next to the selected one, it is categorized as a splitting point. These pixels will be colored blue. The rest of the length function is used to get measurements of length for each segment of the cracking. Looping through the array of endpoints, if the pixel is part of the middle of the line, the length is updated. Once the pixels get to an endpoint, the total length is submitted. If the program reaches a splitting point, it is treated as an endpoint, and the splitting point is added as an endpoint. The length is outputted and then the program moves to the next endpoint. Once all of the lengths are measured, the output will look like the one to the left.



STEP NINE

As for the width, the function uses all the slopes calculated from the three to five consecutive pixels in the skeleton. For each point, a width is calculated and then shown on the image. This is calculated by getting the slope of each point and then getting the perpendicular slope of each point. Then, the slope is made into a fraction. The x coordinate is added to both sides of the x direction and the y coordinate is added to both sides of the y direction. This is looped through until the width has hit the edge. This is shown in the images below.



STEP TEN

The length and width is outputted in the command window and put into the text file output.txt.

