

Tesseract Text Detection

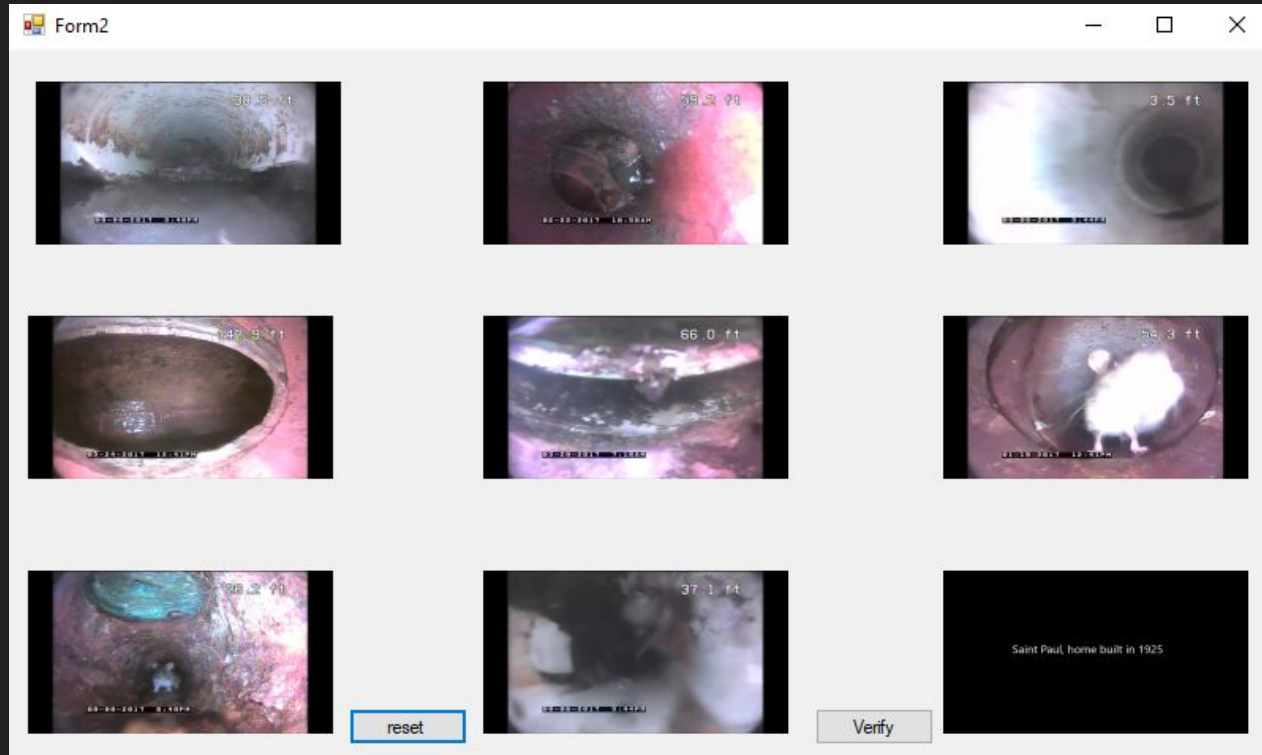
UI PROCESS

- We load the video in the C# interface
- We randomly choose 10 different frames from the video and display it in the interface
- We mark the area where we want to run our detection algorithm
- We verify if the marked area is correct or not
- We do some image processing on the marked area for the 10 randomly generated frames to clean up the images
- Then we run the tesseract code on the processed image to get the output

Loading the Video



Generating 9 Random Frames



Marking the Area



Verification

We generate 9 more random frames with the marked area to verify if its correct or not.



Image Processing

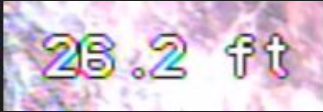
After we have all the marked areas, we run some image processing through them to get better results from our algorithm.

- Denoising of the Image
- Grayscale of the Image
- Image contrasting to get a black and white Image depending on the threshold
- Blur the Image for better edge detection
- Trying sobel edge detection
- Canny edge detection

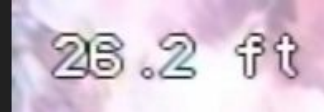
Image Processing

Denoising-

Original Image



Denoised Image



Stats- (Out of 100 images)

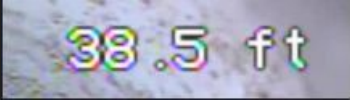
Accuracy- 61%

Images with darker background tend to give correct results but the ones with a lot of noise or a brighter background tend to give wrong results.

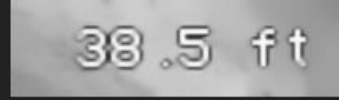
Image Processing

Grayscale

Original Image



Grayscale Image



Stats-

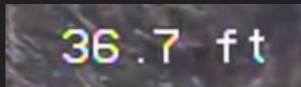
Accuracy- 59%

Although it looks better than just denoising it gives more error.

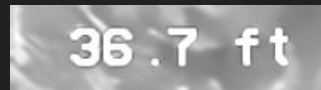
Image Processing

Image Contrasting

Original Image



Contrasting Image



Then Changing it to black and white



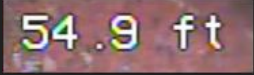
NOTE: The hyper parameters will be different for different backgrounds, there is no universal number. Here we are using the one that was most commonly used.

Low Accuracy

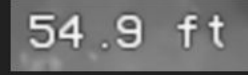
Image Processing

Gaussian Blur-

Original Image



Blurred Image



Stats-

Accuracy- 64%

Images with darker background tend to give correct results but the ones with a lot of noise or a brighter background tend to give wrong/no results.

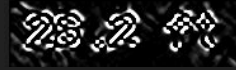
Image Processing

Sobel Edge Detection

Original Image



Sobel on blurred Image



Sobel on contrasted Image



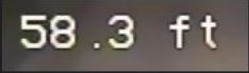
NOTE: The hyper parameters will be different for different backgrounds, there is no universal number. Here we are using the one that was most commonly used.

Low Accuracy

Image Processing

Canny-

Original Image



58.3 ft

Canny on Blurred Image



58.3 ft

NOTE: The hyper parameters will be different for different backgrounds, there is no universal number.

Could have a high Accuracy if we could use different hyper parameters for different backgrounds and fill up the numbers. Currently tesseract cannot detect this.