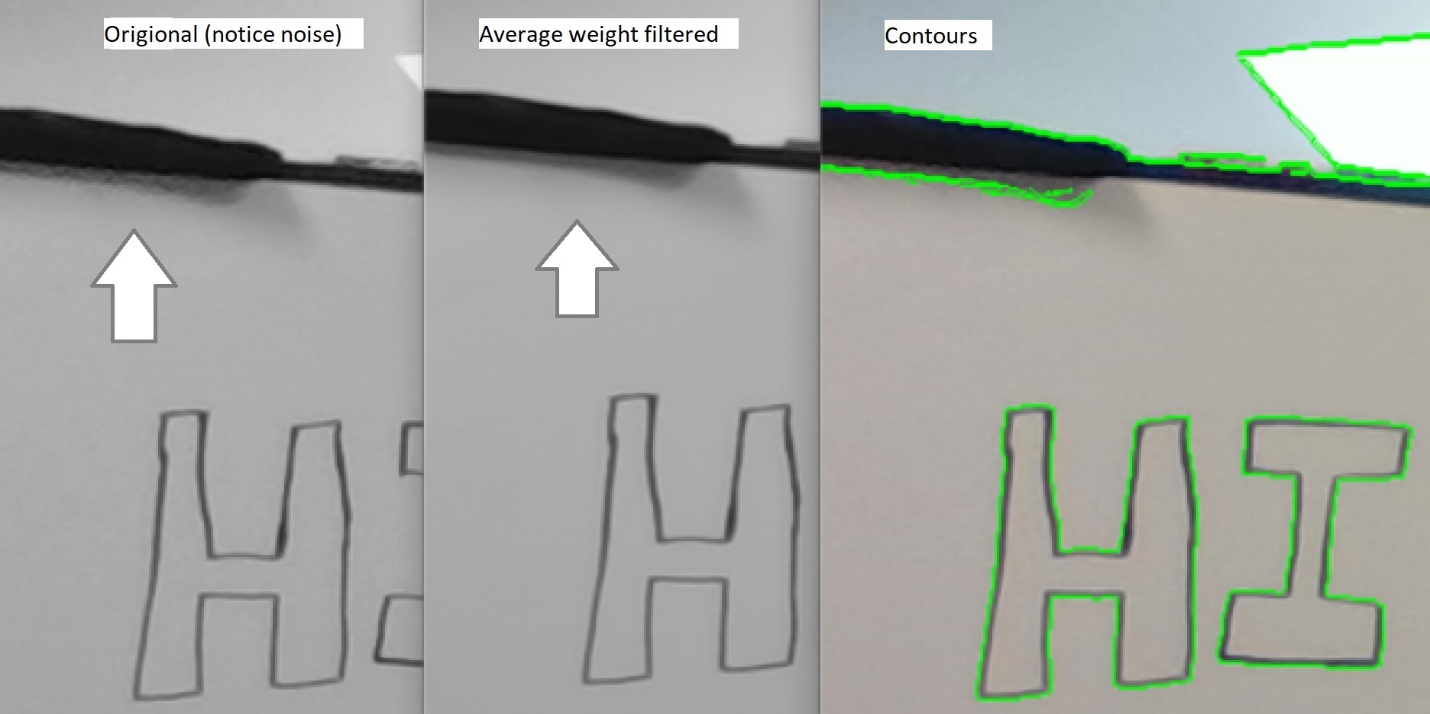
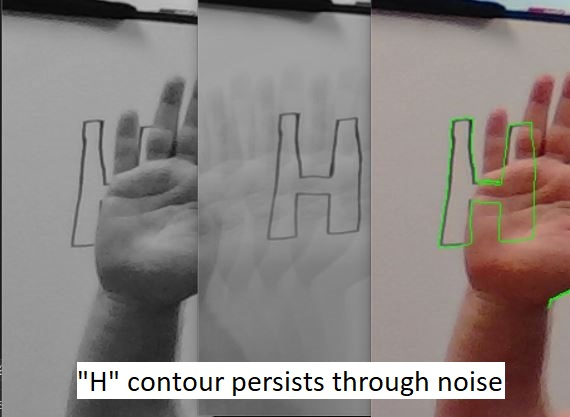
Program was created to demonstrate proof of concept when n images are taken and the “per pixel” average is calculated, the image will be better suited for finding contours using OpenCV. The GPU is used to perform some of the calculations using CUDA directly. The pre-built cv::gpu:: functions were not used since the purpose of the project is to explore and get a very low level understanding on how OpenCV and CUDA work.

In short the program will take n images and will continue taking images replacing the oldest image. This way there are always n images to process. The n images are averaged per pixel to produce a single image. This image is very clear and most noise is eliminated.



Once the averaged image is produced, The image is blurred and filtered to find contours. The contours are more stable than in a single image.



This is useful if the contours that are being found are static in an image feed. For example, an Optical Character Reader (OCR) that takes in a live camera feed.

To produce an image at a reasonable frame rate the GPU had to be used. The program takes n images as stated above. So there are n \* Width \* Height pixels which can quickly add up as n increases. Therefore, each of the pixels produced on the final image is ran in its thread.

The GPU is also used to filter the contours that to big or to small than the desired range. The there are 2 sliders that are used to adjust the range at run time.