James Coleman

CS 3150

Homework 2 part 1

September 18th, 2020

To begin I brought in python’s cv2 and numpy packages for creating and transforming images. I also imported randrange from random and pyplot from matplotlip. pyplot will be helpful for displaying my images and I’ll use randrange for my own random pixel filter.

I set up a helper method called show() that displays my original image along with any number of images with in a list of images. The list of images is actually a list of tuples where the first element of the tuple is the image and the second element is the name of the image.

After importing my image, converting it from BGR to RGB (I’m not positive that this step was necessary. Maybe I could have gone straight from BGR to gray) and then to gray, I’m ready to start applying filters.

* Average filters

For averaging filters, I began with a 3 by 3 box but after applying it and viewing it the change was barely perceptible. I then tried a 5 by 5 box but still the change was hard to notice. So finally, I jumped the dimensions up to 11 by 11 to try and force a significant blur and that did the trick:

A group of people posing for a photo

Description automatically generated

The change is still a bit hard to see but it is most evident in the leaves on the left-hand side of the image.

* Sobel filters

Sobel filters have a far more obvious effect compared to the average filters and I implemented both a vertical and a horizontal version. To combine them into a gradient filter I first tried to average the two images that I got from the vertical and horizontal filters together. This approach worked ok but what it produced was more of an image of where the two filters where overlapping. Using numpy’s maximum function ended up being a better way of acquiring the gradient image:

A screenshot of a cell phone

Description automatically generated

The difference between vertical and horizontal is most pronounced in the building’s siding. The vertical filter does not pick up the outline of the boards while the horizontal one does easily. The gradient image on the other hand really highlights the leaves in the background and also gives more definition to the contours of my shirt.

* Laplacian filters

I implemented two Laplacian filters a four neighbor and an eight neighbor. The four-neighbor filter turned out very dark leaving only faint out lines. However, those outlines that can be seen are very informative as they seem to be the most essential info about the shapes in this image. The eight-neighborhood image has more edges present and also has brighter edges:

A person with collar shirt

Description automatically generated

* Median filter

For my median filter I chose to take the median of 25 surrounding pixels. The resulting image is the most subtly different of all my filtered images. Overall the image just appears to be slightly grayer. The spot that jumps out to me as being different is right under my teeth. That dark line seems to be a bit thinner in the median filter image:

A person posing for the camera

Description automatically generated

* Gaussian filters

The Gaussian filters where also very subtle. I tried three different masks from the slides and tried some adjustments on them. I was sure to divide the filters by the sum of the weights and still I am not sure I can tell the difference in these images:

A screenshot of a social media post

Description automatically generated

* Prewitt filters

For my first optional filter I wanted to try the Prewitt filter and see how it compares with the Sobel filters. My prediction is that they would be *very* similar. So, using the same process that I used for the Sobel filters but with slightly different boxes I constructed a vertical, horizontal and gradient filter along with accompanying images. Then to get an idea of the difference I mounted each one next to the corresponding Sobel image:

A person with collar shirt

Description automatically generated

A person with collar shirt

Description automatically generated

A person with collar shirt

Description automatically generated

My prediction was correct.

* Random pixel filter

For my last filter I decided to come up with my own. I basically created a filter that would choose a random pixel from a 11 by 11 box around the target pixel and replace it. The result is a pixelated blur that I think looks pretty interesting. It has a grainy blur to it as opposed to the average and gaussian filters that have very smooth blurs.

A person posing for the camera

Description automatically generated