Project 2

Introduction:

For my individual term project, I picked the clear view and poor weather separation option in project 2. To complete this project, I had to preform image manipulation, image filtering, edge detection, edge data retrieval, and image sorting. In total, 1169 dashcam images were used in the project which consisted of various locations, weather, and time of day. The result of this project was to display a histogram for every image and then display histograms of the sorted clear and poor weather average values. Clear weather consisted of sunny, cloudy, and night while poor weather consisted of foggy, snowing, and raining.

Method:

To begin this project, I first had to retrieve the images that I would be processing. The 1169 dashcam images were downloaded from the CDVL website ([CDVL - View File](https://www.cdvl.org/members-section/view-file/?id=3027)). Once I had the images, I used photoshop to convert and resize them using batch processing. This action consisted of making the images grayscale, resizing them to ¼ width and height, and then saving them as raw files for processing.

Now that I had all of my images the right size and ready to be processed, I started to work on my program. I used C++ for my implementation and tested everything in Linux to make sure that it worked once I submitted it. In my program I defined image width to be 648 and the height to be 486 for all to the images. To cycle through all 1000+ images I used “int argc, char \*argv[]” in my main function so that I can simply specify the image directory in the command line when executing the code to get each image file. Then one by one all the images have a 3x3 and 5x5 differential filter applied to the central 1/3 width and height of the image. These filters were then used to acquire the 3x3 and 5x5 gradient of the images. While computing the gradient then maximum values was of the 3x3 and 5x5 were stored for later use in determining the weather condition.

For each image a histogram must be displayed so to do that I created two arrays of size 16 with each array space representing a 16-value bucket from 0 to 256. Then for each image gradient value on both the 3x3 and 5x5 filters I checked which bucket they would fall under then added one to their respective bucket. At the end of each images cycle their 3x3 and 5x5 histograms are then printed showing the user what values each image gives on its own.

The last thing to do was to separate the images and display their average histogram values. This was first done by creating four more arrays, two for poor weather and two for clear. Then using the max 3x3 and 5x5 gradient values previously found, set a tolerance level of what images are considered poor to separate the images. For the histograms the same process as the individual images is done except that then values keep adding up until all images are processed. Once all the images are processed each bucket in all four histograms are divided by the number of poor or clear images to get their averages. Then the histograms are displayed along with the number of poor weather and clear weather images in each category.

Report:

The result of the program is a 3x3 and 5x5 gradient value histogram for each image and four summary histograms showing the average 3x3 and 5x5 gradient values of all the poor weather and clear weather images. The number of images in each category is also displayed to see the separation of the 1169 images. My analysis of the results is that clear weather images have a higher gradient maximum value than poor weather images.