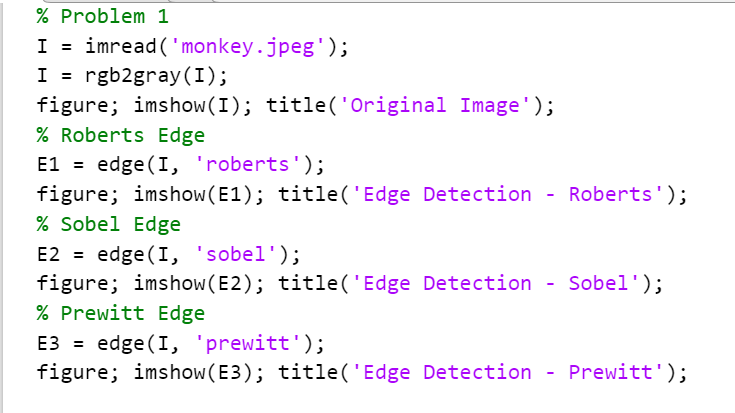
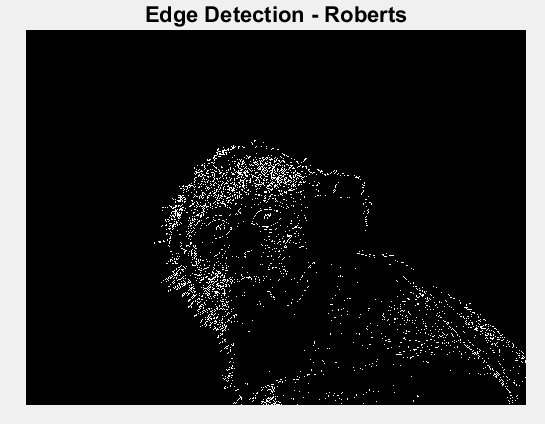
Problem 1:



Code for Part 1 Problem 1

Original Picture

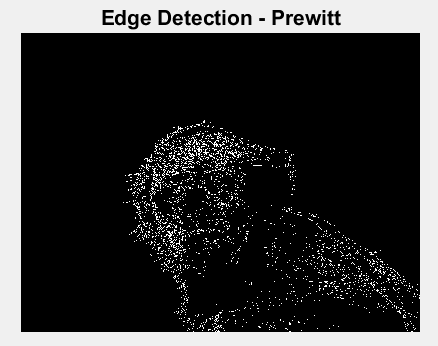

Original Picture



Edge Detection- Roberts



Edge Detection- Sobel



Edge Detection- Prewitt

The Roberts operator is the least successful of the three edge detection methods since it is unable to identify distant land near the horizon. Although the Sobel and Prewitt operators are comparable, the Sobel does appear to capture edges in more detail (for instance, it captures the edges of the several rings on the front bike tire better than the Prewitt operator).

Problem 2:

1st Order Derivative:

2nd Order Derivative:

Image: 6 6 6 6 5 4 3 2 1 1 1 1 1 1 6 6 6 6 6

1st Derivative of the image:

0 0 -1 -1 -1 -1 -1 0 0 0 0 0 5 0 0 0 0

2nd Derivative of the Image:

0 0 -1 0 0 0 0 1 0 0 0 0 5 -5 0 0 0

Problem 3:

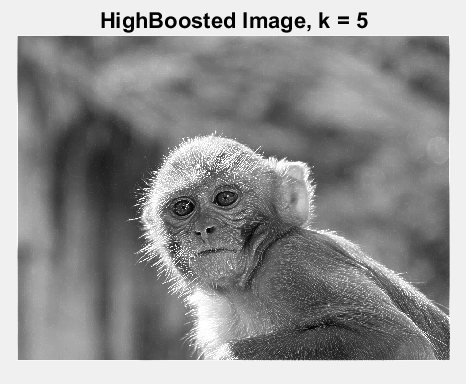
A screenshot of a computer code

Description automatically generated

Code



High Boosted Image when k=1



High Boosted Image when k=5

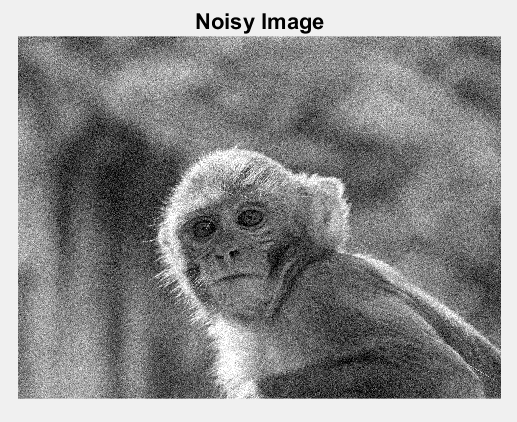
By just applying the unsharp masking to the original image when k=1, we achieve little to no effect. High boost filtering is what happens when k>1, as the second image, "High Boosted Image when k=5", demonstrates. The image has a noticeably improved level of sharpness.

Problem 4:

A computer code with purple text

Description automatically generated

Code



Noisy Image

A close-up of a monkey

Description automatically generated

Average Filter



Gaussian Filter

The gaussian filter removes gaussian noise more effectively, as seen in the Figures. However, one negative effect is that the final denoised image is hazier.

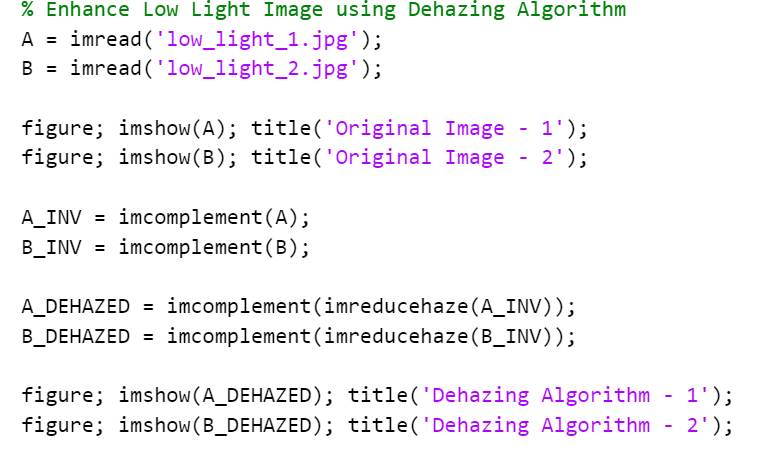
Problem 5:

1:

2:

|  |  |  |
| --- | --- | --- |
| 4 | 6 | 4 |
| 6 | 3 | -8 |
| 1 | -7 | -4 |

Part 2:



Code 1

A screenshot of a computer code

Description automatically generated

Code 2

A screenshot of a computer

Description automatically generated

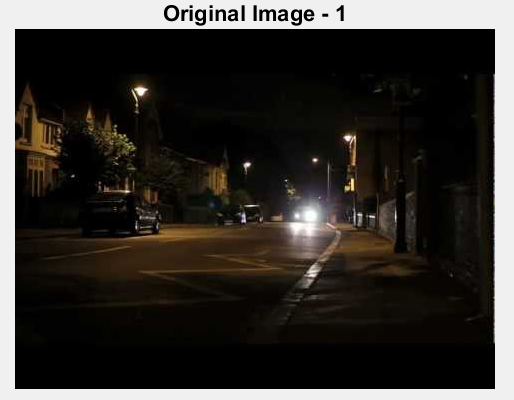
Code 3

A screenshot of a computer program

Description automatically generated

Code 4

1:

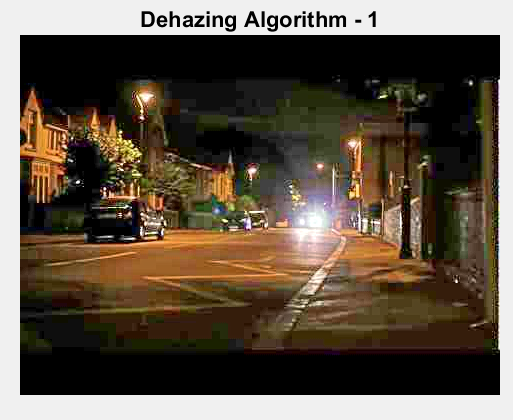


Original Image- 1



Original Image- 2

1:

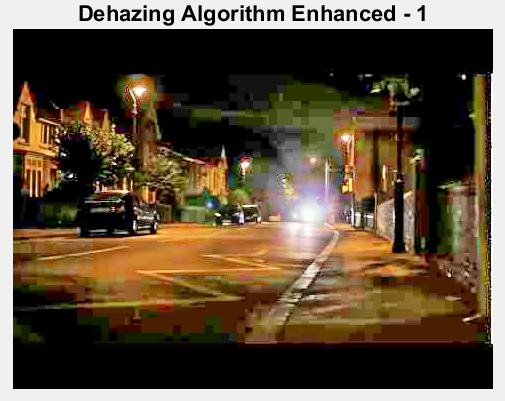


Dehazing Algorithm 1

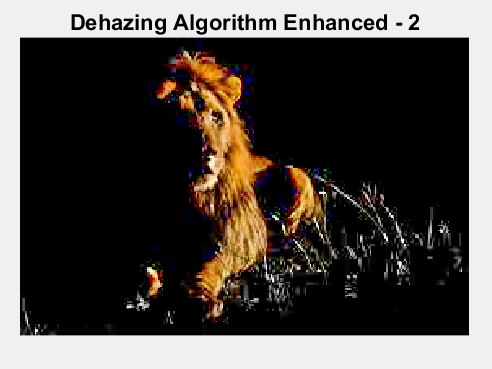


Dehazing Algorithm 2

2:



Dehazing Algorithm Enhanced 1

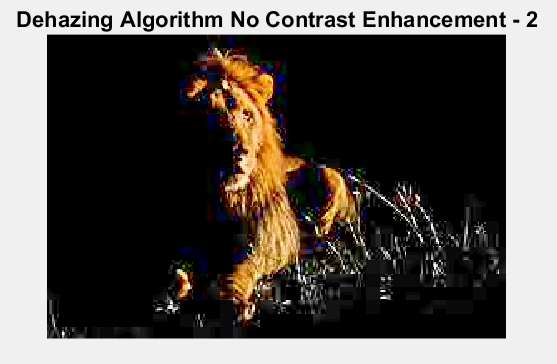


Dehazing Algorithm Enhanced 2

3:



Dehazing Algorithm No Contrast Enhancement 1



Dehazing Algorithm No Contrast Enhancement 2

4:

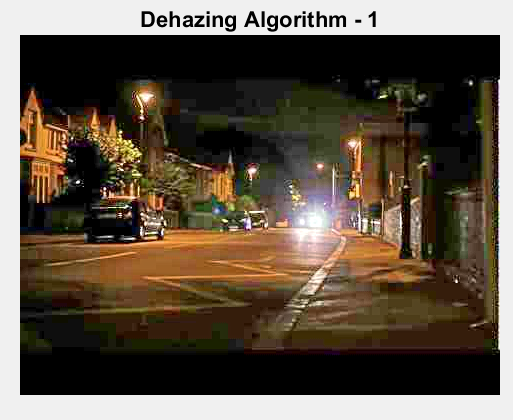


LAB Dehazing 1



LAB Dehazing 2

5:



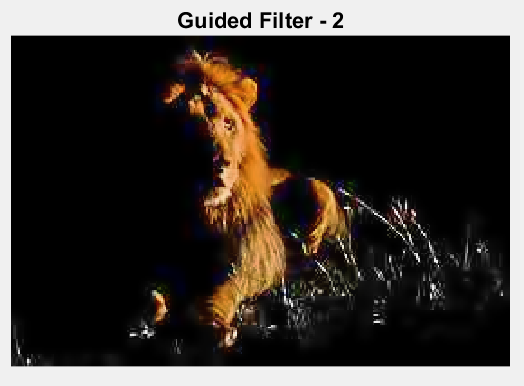
Dehazing Algorithm 1



Guided Filter 1

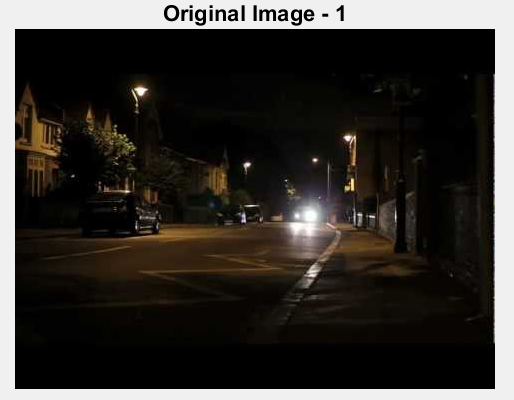


Dehazing Algorithm 2



Guided Filter 2

6:



Original Image 1

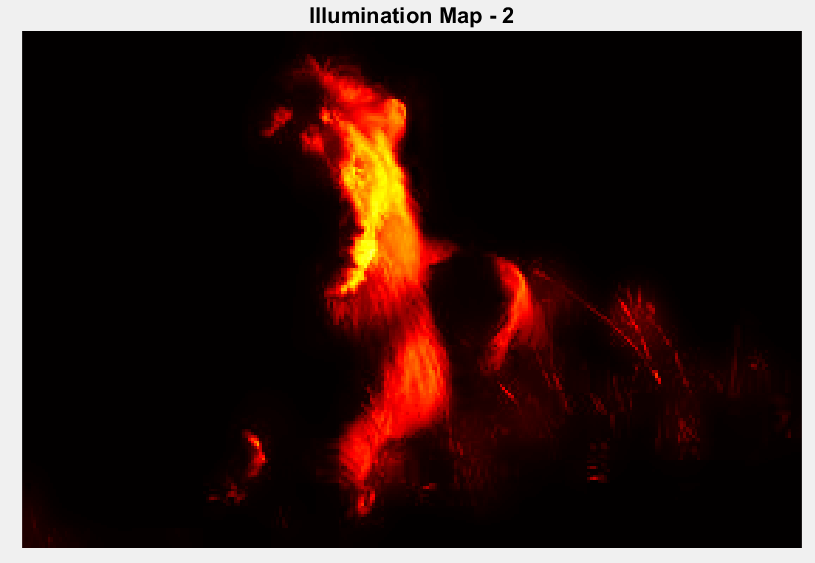


Illumination Map 1

A lion lying down in the dark

Description automatically generated

Original Image 2



Illumination Map 2

An effective approach for improving a low light image is the image enhancement algorithm. It functions in two key steps. The low light image is first inverted, and then it is subjected to an optimised use of an image dehazing method.

The general dehazing algorithm is initially applied to the low light photos. By including certain optional options in the second stage, such as raising the contrast enhancement, we further strengthen this technique. The identical process is used in step three, but without any contrast augmentation, producing a superior overall low-light image. The dehazing method is applied to our image in the fourth step, and then we convert it back to RGB, which regrettably produces subpar results. In the fifth stage, we apply a denoising filter after dehazing because low light photographs typically have a lot of noise because cameras with higher focal lengths do (lower focal length cameras create more clear and less noisy low light images). To see where the majority of the light is coming from, we construct an illumination map from the original photos in the sixth phase, which is shown by brighter colours in the heat map.

Part 3:

The idea I have for the project is Object Tracking Techniques revolving around self driving cars.