|  |  |
| --- | --- |
| Name: | Jason Rich Darmawan |
| Student ID: | 2024280038 |

# Introduction

Edges in an image appear where there is a sudden change in pixel intensity. To explain this, consider a grayscale image with two dimensions—rows and columns. An RGB image, on the other hand, has an extra third dimension for color channels. In this example, we analyze an image of a dog to observe the difference in pixel intensity between the dog and the background (see Figure 1.1). As shown in the figure, the intensity changes quickly between the background and the dog, but it is still difficult to pinpoint the exact edges.

A collage of images and graphs

Description automatically generated

To detect edges, we calculate the derivative of the image. The derivative shows how quickly the pixel values are changing, and edges occur where the change is the greatest (see Figure 1.2). One simple way to do this is by using the first difference or central difference, which measure changes in pixel values by comparing neighboring pixels in horizontal (. This equation computes the difference between the intensity of the pixel at ) or vertical directions.

|  |  |  |
| --- | --- | --- |
|  |  | (1.1) |

|  |  |  |
| --- | --- | --- |
|  |  | (1.2) |

|  |  |  |
| --- | --- | --- |
|  |  | (1.3) |

|  |  |  |
| --- | --- | --- |
|  |  | (1.4) |

Horizontal (x) direction:

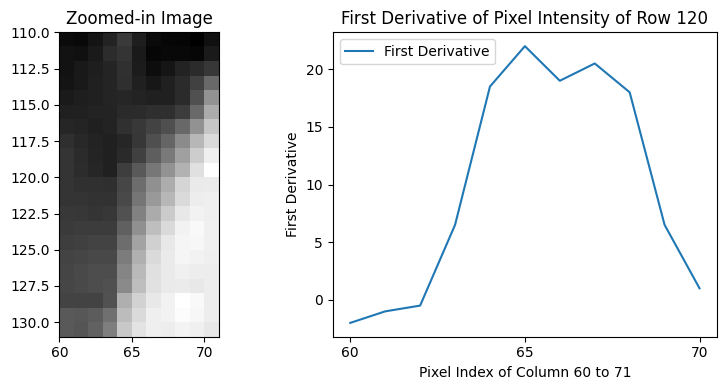
Vertical (y) direction:

Central difference:

Horizontal (x) direction:

Vertical (y) direction:

By combining the gradients from both directions, we can detect the magnitude of changes in intensity, which highlights the edges. In images with low noise, such as our example of a dog in an open field, these simple methods can work well and produce results similar to the Sobel operator. However, the Sobel operator includes a smoothing effect, which makes it more robust in images with more noise or complex textures, where small random variations in pixel intensity could interfere with edge detection.



To improve edge detection, we apply the Sobel operator, which is specifically designed to detect edges more reliably. The Sobel operator combines two key ideas: it approximates the derivative by measuring intensity changes, but it also smooths out noise by using a weighted 3x3 kernel. Additionally, it uses two separate kernels for detecting edges in both the horizontal and vertical directions. This combination of smoothing and directional sensitivity makes the Sobel operator more effective at detecting edges, even in noisy images (see Figure 1.3).

# Python Scripting with Explanation

AAAAAAAAAAAAAAAAAAAA

# Results and Discussion

AAAAAAAAAAAAA

# Conclusion

AAAAAAAAAAAAAAAAA