EE655: Computer Vision and Deep Learning

HOMEWORK - I

Name: Vaibhav Itauriya

Roll Number: 231115

Branch: ME

Date of submission: 31/01/2025

GitHub Repository of this Course:

Contents

1 Introduction

In this assignment, we explore the concept of Integral Image and its application in filtering techniques. The goal is to compute the response of various filters on a given image using the Integral Image method. Instead of processing individual pixels, this technique allows for efficient computation of filter responses by leveraging summed-area tables. The GitHub repository shared here contains the implemented code.

2 Implementation Details

The implementation follows these key steps:

- **1. Reading the Image:** The input image is loaded using the OpenCV library; it has a dimension of (168,299).
- **2. Computing the Integral Image:** The integral image is computed to facilitate rapid filtering. We used the cv.integral() function and removed the first row and column for filter implication.
- **3. Applying Filters:** Various filters are applied over the image using the integral image representation for efficiency, as given in the assignment.
- **4. Visualising Results:** The output responses for each filter are generated and displayed. The output generated have shape (165, 296).

3 Results

The following images present the results after applying all six filters to the original grayscale image. Each row consists of the original image, the filter used, and the output visualisation in grayscale and heatmap representations.

3.1 Filter 1

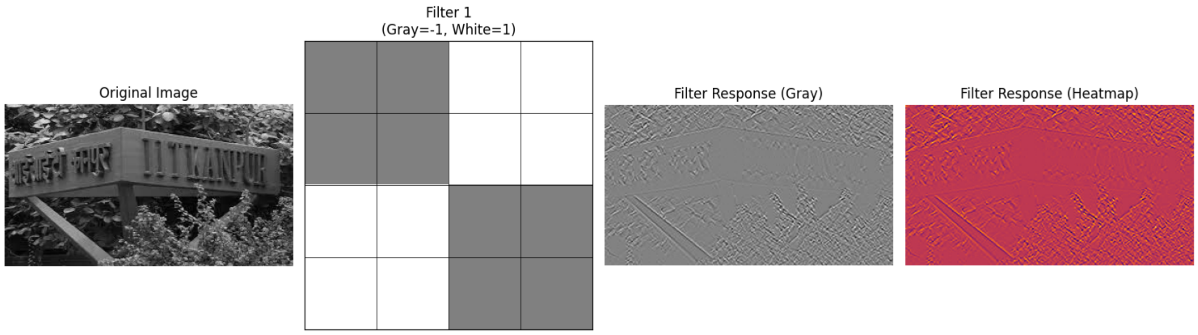

Figure 1: Filter 1 Outputs

This filter enhances edges oriented at 135 degrees, highlighting diagonal structures in the image. The result highlights textures along slanted structures while suppressing horizontal and vertical features.

3.2 Filter 2

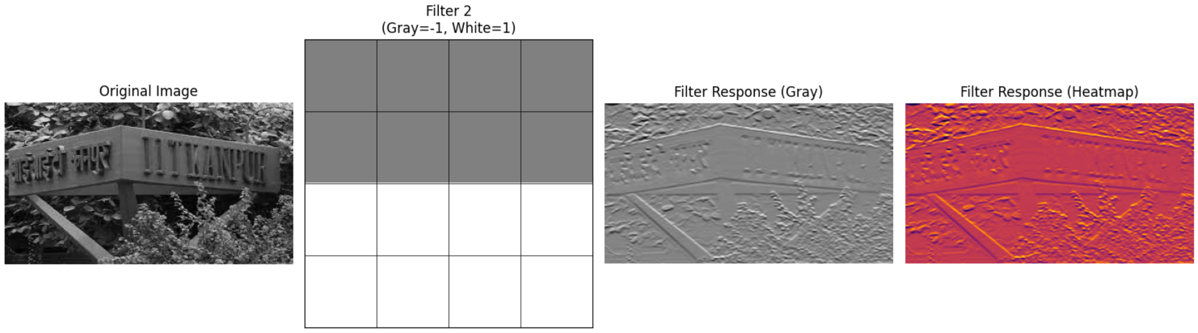

Figure 2: Filter 2 Outputs

This filter extracts horizontal edges by emphasising intensity changes along the horizontal axis. The heatmap confirms the enhancement of structures aligned along the horizontal axis.

3.3 Filter 3

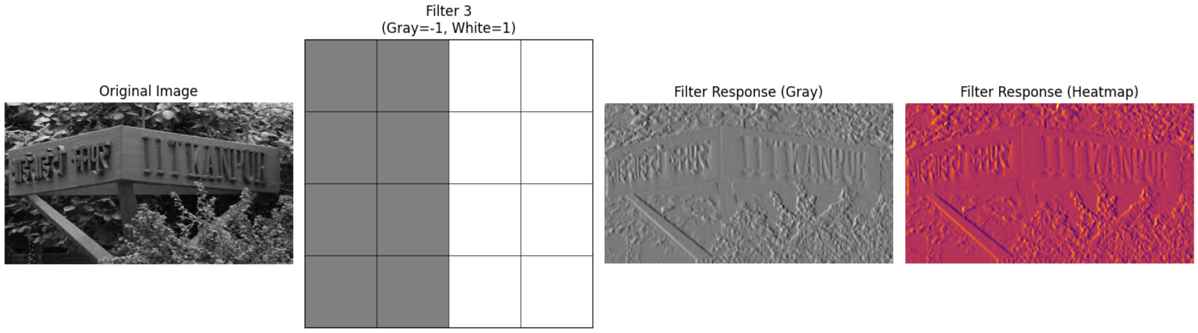

Figure 3: Filter 3 Outputs

Vertical edges are detected by focusing on intensity variations along the vertical axis, detecting strong vertical lines like the IIT Kanpur letters in the image. Both grayscale and heatmap versions emphasise these linear features.

3.4 Filter 4

The response of this filter focuses on diagonal edges oriented at -45 degrees. Compared to Filter 1, it captures the opposite diagonal direction, making slanted textures more evident.

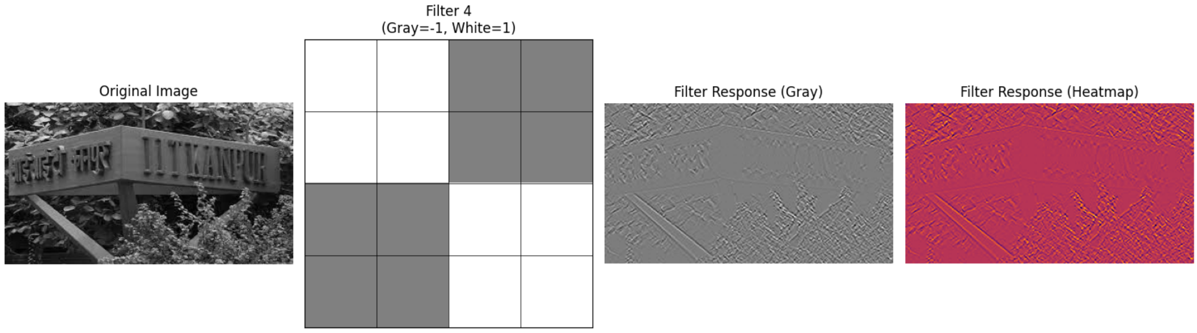

Figure 4: Filter 4 Outputs

3.5 Filter 5

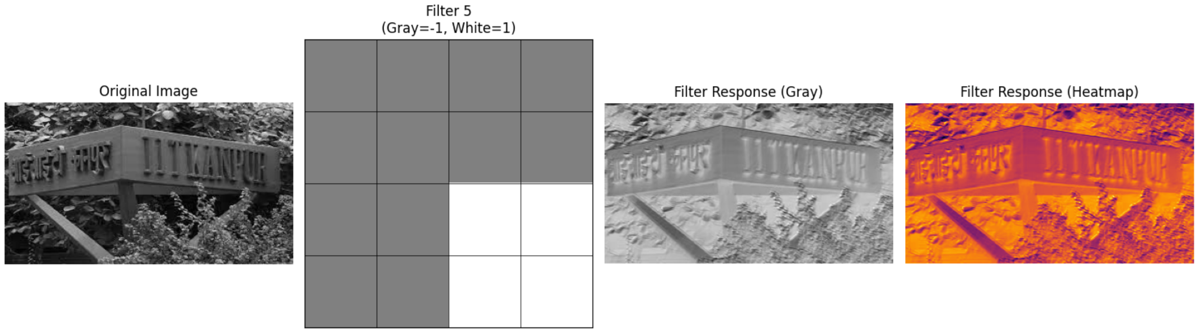

Figure 5: Filter 5 Outputs

This filter likely emphasises specific textures or noise patterns, further refining edge structures in the image.

3.6 Filter 6

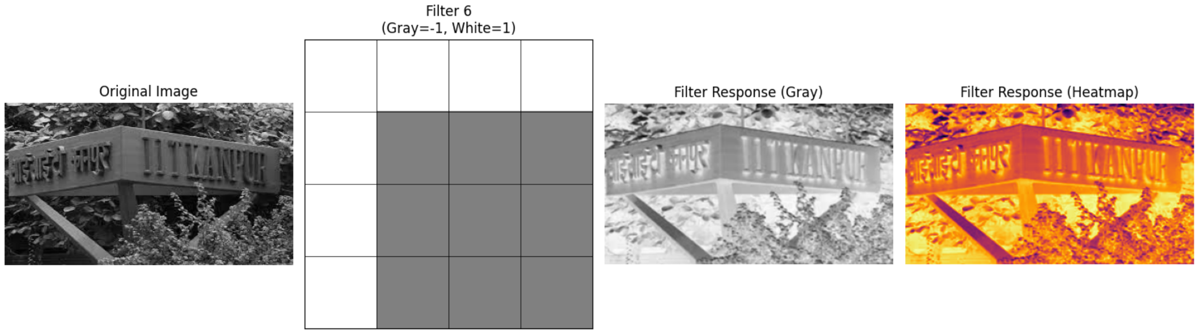

Figure 6: Filter 6 Outputs

This filter captures a unique aspect of the image's structure, possibly enhancing contrast or identifying finer details.

4 Problems Faced

During implementation, several challenges arose:

- **Indentation Issues:** Minor formatting errors caused runtime errors, resolved with careful debugging.
- **Difficulties with Integral Image Implementation:** Initially, applying filters correctly using the integral image was challenging. However, a correct implementation was achieved after reviewing lecture notes and consulting online resources.

5 Conclusion

The use of integral images for efficient filtering proved to be highly effective. The implemented filters successfully extracted various edge orientations, demonstrating the power of this technique. Despite initial hurdles, a deeper understanding of integral images and filtering methods was gained, making this assignment a valuable learning experience.

6 Code

Here is the complete code for this Homework Assignment:

Code Link: Click Here