Edge Detection and Analysis Report

Mruganshi

B20CS014

Introduction

The goal of this task was to investigate algorithms, for image processing and edge detection. We took a step-by-step approach beginning with image preprocessing. Then we implemented advanced orientation filters using Gabor wavelets. Applied Winner Takes All (WTA) and normalization operations. In the end, we conducted an analysis of the results obtained from edge detection methods.

Assignment Objective and Challenges

The primary goals of this assignment were;

1.. Processing a collection of natural images.
2. Applying complex orientation filters using Gabor wavelets to extract edge features.
3. Implementing Winner Takes All (WTA) and normalization operations to enhance the extracted features.
4. Conducting an analysis of edge detection, across images taking into account factors like complexity, texture, and lighting.
5. Introducing noise to the images. Assessing the performance of edge detection in conditions.
6. Visualizing the detected edges and features on the images.
7. Analyzing the significance and limitations of using Gabor-based edge detection.

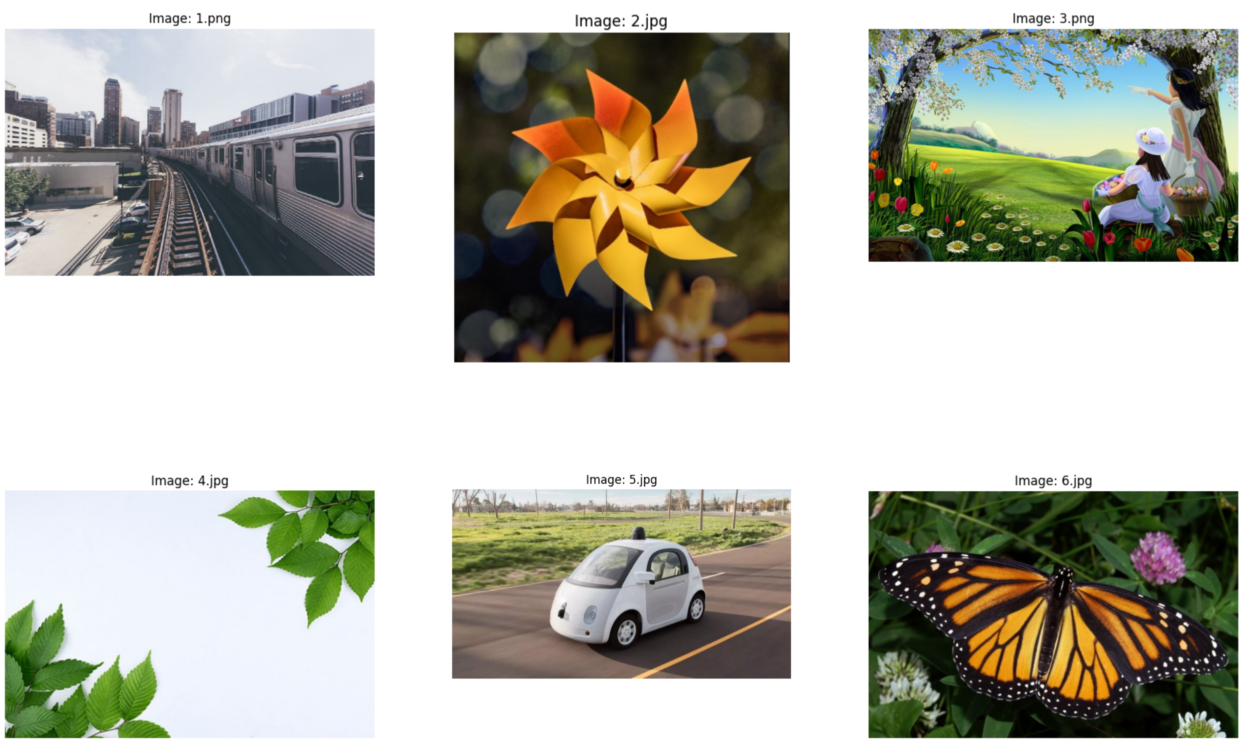
In the sections, we will provide explanations of each technique used in every part showcase intermediate and final results, for multiple images conduct a comparative analysis, and discuss both the importance and constraints associated with employing Gabor-based edge detection.

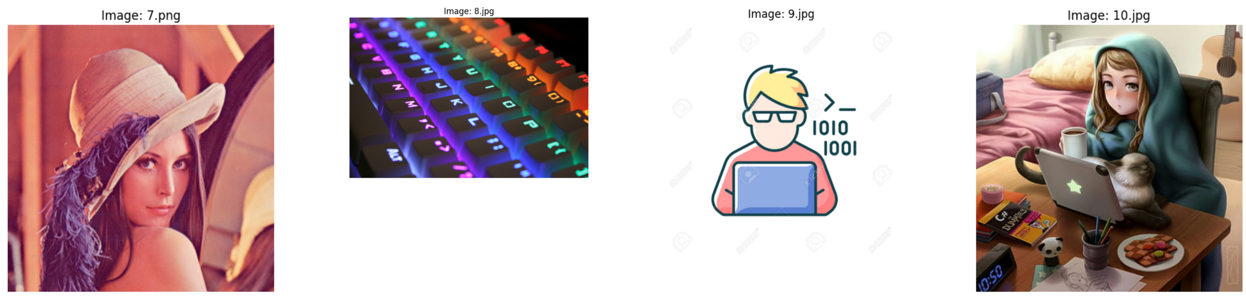
Part-1: Image Preprocessing

Loading and Describing Images

To start we initiated the process by setting up a folder on Google Drive. In this folder, we added images, with a minimum resolution of 400x400 pixels. Our aim was to include a range of images that captured complexities, perspectives, and lighting conditions. To provide context for each image we included a description, alongside it. You can refer to Figure 1 below for an example. (the shown image size is not the same as I used in the code)

```
Figure - 1
```

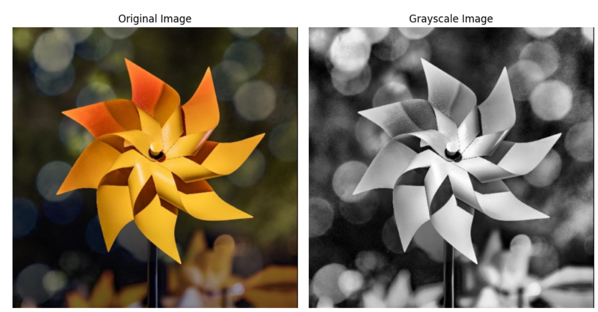


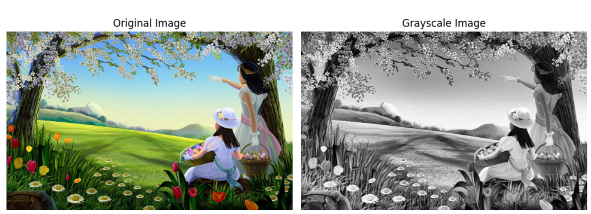


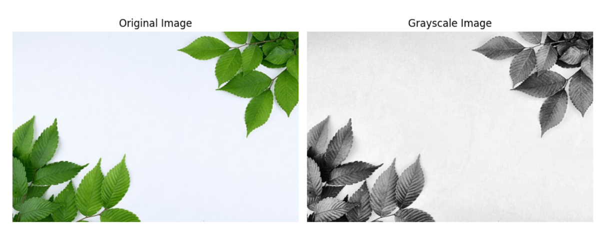
Grayscale Conversion

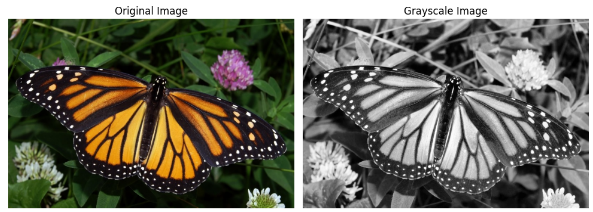
Afterward, we transformed every picture to black and white while guaranteeing ideal contrast improvement and dynamic scope conservation. The pictures were preprocessed to enhance their appropriateness for edge identification. Figure 2 demonstrates the transformation procedure for one of the pictures.

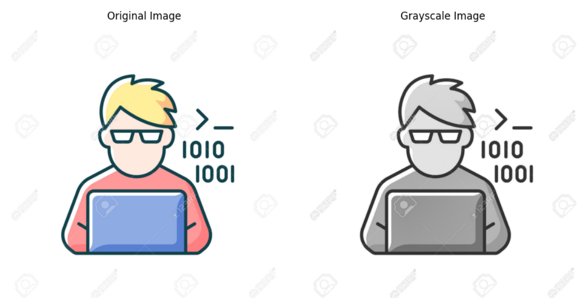
```
Figure - 2
```

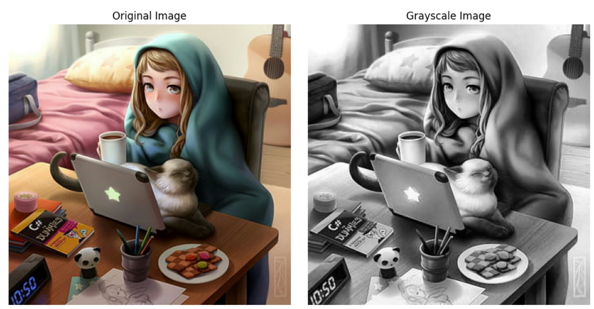












Part 2: Complex Orientation Filters

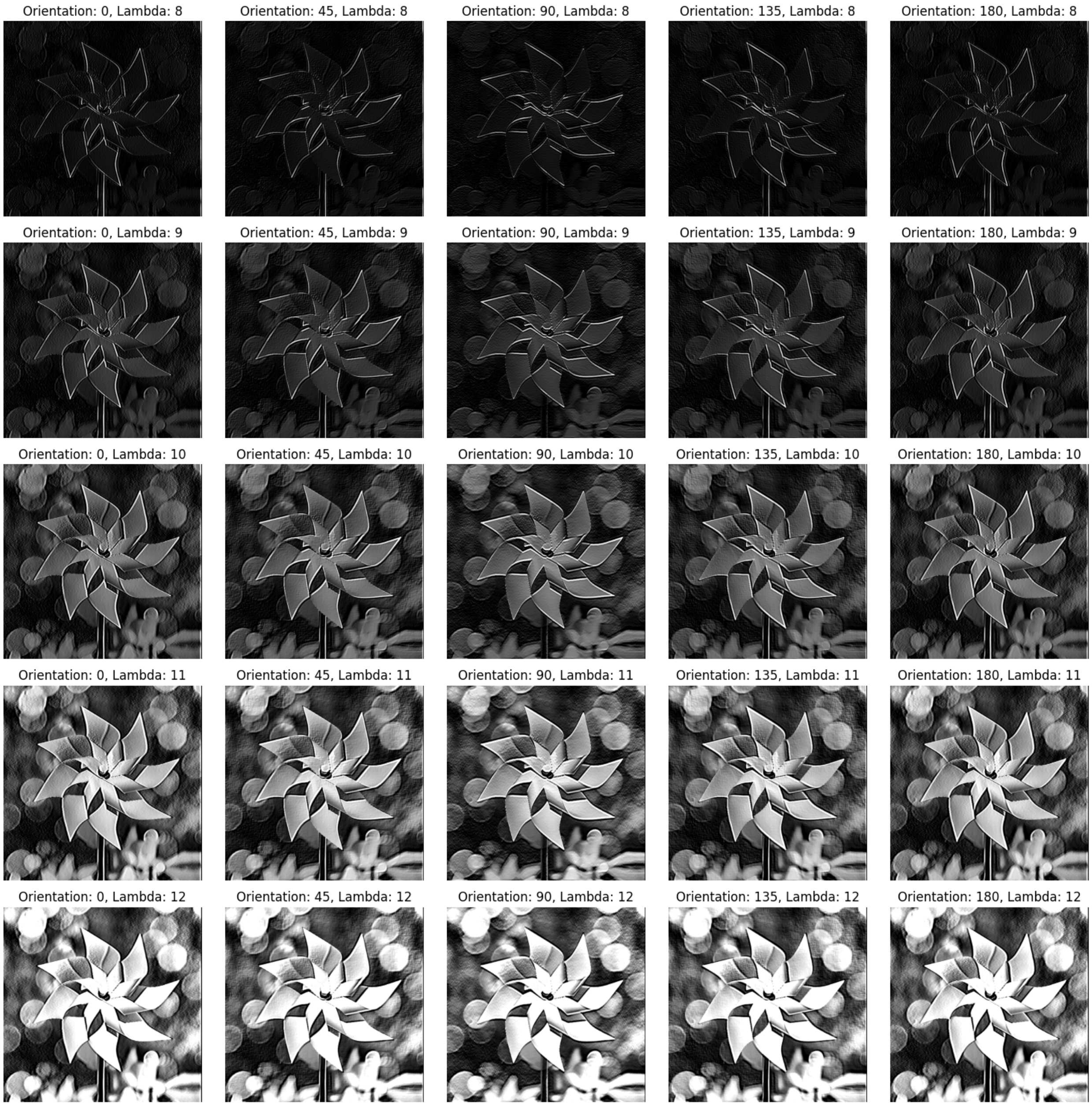
In this part, our attention was on implementing intricate direction filters based on Gabor wavelets to extract boundary characteristics. We generated several filters with different directions (0, 45, 90, 135, and 180 degrees) and frequencies (8, 9, 10, 11, 12). These filters were employed on a high-resolution grayscale image chosen from Part 1 using convolution operations. The processed images for each direction were displayed to highlight the extracted characteristics, as depicted in Figure 3.

code explanation:

1. The `gabor_filter` function takes three arguments: `img_indx` (index of the image to process), `thetas` (a list of orientation angles), and `frequencies` (a list of frequencies).

2. Within the function, a loop iterates over each combination of orientation (`theta`) and frequency (`frequency`) from the provided lists.
3. For each combination, a Gabor kernel is created using the `cv2.getGaborKernel` function. This kernel is defined by parameters like size (31x31), standard deviation (`sigma`), orientation (`theta`), wavelength (`lambd`), aspect ratio (`gamma`), phase offset (`psi`), and data type (`ktype`).
4. The Gabor kernel is then applied to the grayscale image (`gray_img[img_indx]`) using the `cv2.filter2D` function, resulting in a complex-valued filtered image (`complex_img`).
5. The filtered images are appended to the `complex_images` list.
6. The function returns a list of complex-valued filtered images.
7. The code outside the function sets the `thetas` and `frequencies` for the Gabor filters, calls the `gabor_filter` function to process a specific image (indexed at 1), and plots the filtered images in a grid. Each subplot represents a different combination of orientation and frequency, and the titles indicate the values of these parameters

```
Figure 3: Complex Orientation Filters
```



if we see the detailed explanation of Gabor Filters then,

The primary purpose of complex orientation filters, or Gabor filters, is to analyze textures and capture edges in an image. They are well-suited for this task due to their ability to extract features at various scales and orientations. The mathematical basis of Gabor filters stems from the Gabor wavelet, which is a complex-valued function derived from the product of a sinusoidal wave and a Gaussian envelope.

$$g(x, y; f, \theta) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \cdot \cos\left(2\pi \frac{f x'}{\lambda} + \phi\right)$$

we use this concept especially in **Texture Analysis, Edge Detection, Feature Extraction etc.**

<u>Observation from the output:</u>

increasing lambda (lowering frequency) in a Gabor filter leads to a broader analysis of features with reduced sensitivity to fine details and increased emphasis on low-frequency components. Along with this increasing the lambda has several effects like wider spatial extent, smoothing effect, orientation sensitivity, etc.

The theta controls the orientation of the Gabor function. The zero-degree theta corresponds to the vertical position of the Gabor function. increasing theta in a Gabor filter results in a rotated orientation for the filter, which can enhance the filter's sensitivity to features aligned with the new orientation while potentially reducing its response to features at the original orientation. we can see the brightness change in the edges as a result of this.

Part-3: Winner-Takes-All and Normalization

Winner-Takes-All (WTA) Algorithm

I executed a **Winner-Takes-All(WTA)** algorithm that took into account both the size and direction of the intricate filtered pictures. This algorithm assisted in choosing the most pertinent characteristics from the filtered pictures.

In the code, `images` portray the collection of intricate filtered pictures, and `the threshold` is a variable that can be modified to manage the selection standards. The algorithm computes a personalized rating for each picture based on both size and direction, and the picture with the greatest rating is chosen as the supreme representation of characteristics.

Normalization

Winner-Takes-All (WTA) algorithm, which selects the most relevant features from complex filtered images. In this section, I will discuss the normalization process applied to the WTA output to enhance feature visibility while preserving contextual relationships.

To enhance feature visibility while preserving contextual relationships, I performed normalization on the WTA output images. The normalization process involved scaling the magnitude values to a suitable range and applying histogram equalization for improved visibility. Normalization is a crucial step to ensure that the WTA output is visually informative and suitable for further analysis. The normalization process involves the following steps, as implemented in the code:

Visualization

The WTA and normalized images were visualized to capture intricate features and texture patterns, as shown in Figure 4.

Output of WTA and Normalization of WTA image:

```
Figure - 4
```

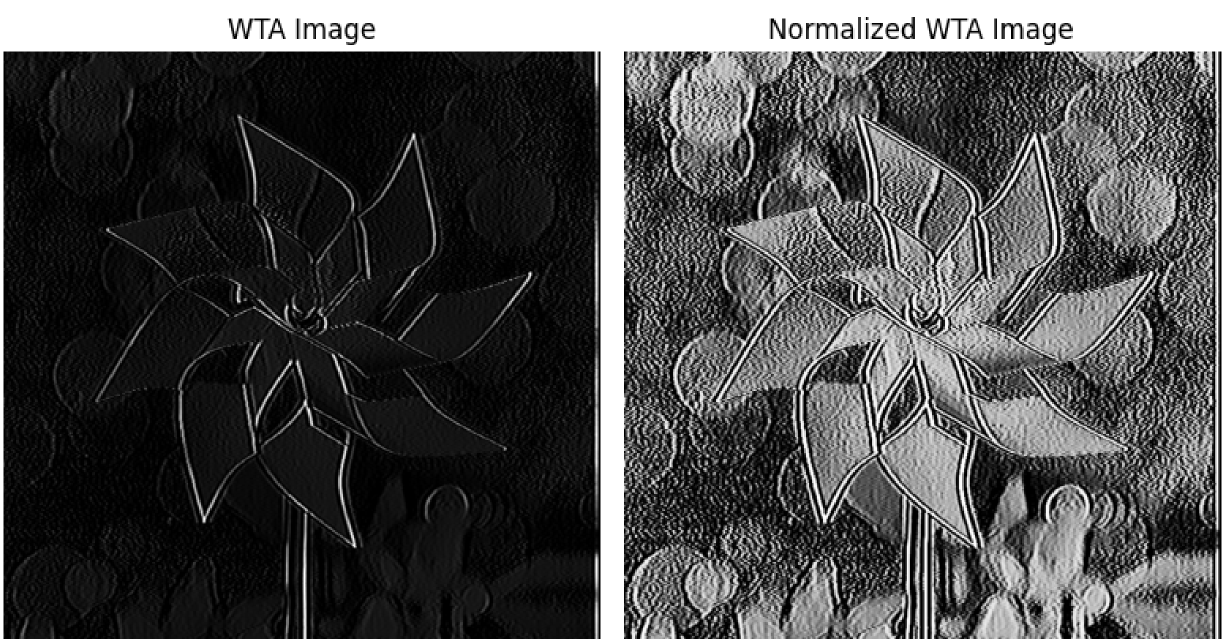


Figure 4 illustrates the impact of normalization on the WTA output, highlighting the enhanced visibility of relevant features while maintaining their spatial relationships.

Observed things:

We can see that in the WTA image output edge detection, magnitude emphasis and orientation sensitivity are high, and in normalized WTA image has more feature visibility, contrast enhancement, and contextual preservation. So, basically, we can say that the WTA image primarily focuses on highlighting edges and features with high magnitudes, leading to a binary-like representation with white edges against a black background. On the other hand, the normalized WTA image aims to strike a balance between feature enhancement and contextual preservation, resulting in a more visually informative representation with improved contrast and detailed features.

Part 4: Comparative Analysis

Image Processing Pipeline

Image preprocessing

Prior to implementing intricate direction filtering, I performed image pre-processing to enhance the images for subsequent analysis. This entailed transforming the loaded images to grayscale using Contrast Limited Adaptive Histogram Equalization (CLAHE). The pre-processing stage guarantees optimal contrast improvement and preservation of the dynamic range.

Complex Orientation Filtering

then I implemented complex orientation filtering using Gabor Filtering with with different directions (0, 45, 90, 135, and 180 degrees) and frequencies. This stage aimed to extract features with specific orientations and frequencies from the images.

Winner-Takes-All (WTA) and Normalization

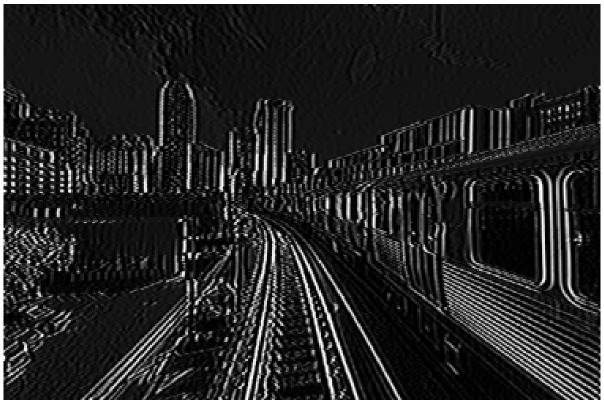
used the same algorithm as in part 3.

Output:

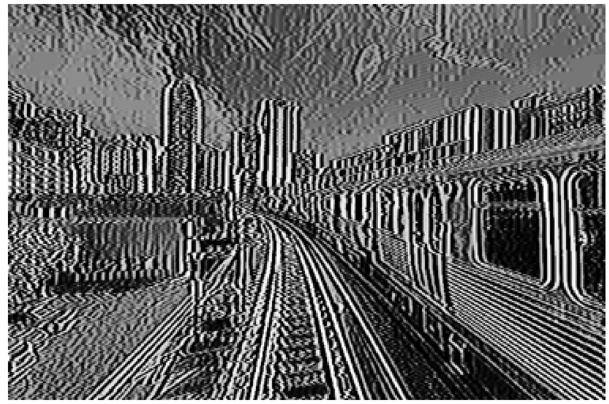
```
Figure - 5
```

<u>**Final output after applying the whole pipeline:**</u>

WTA Image

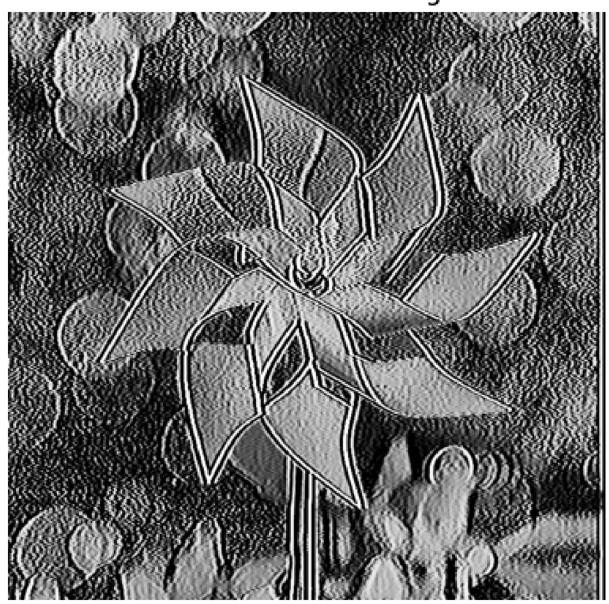


Normalized WTA Image



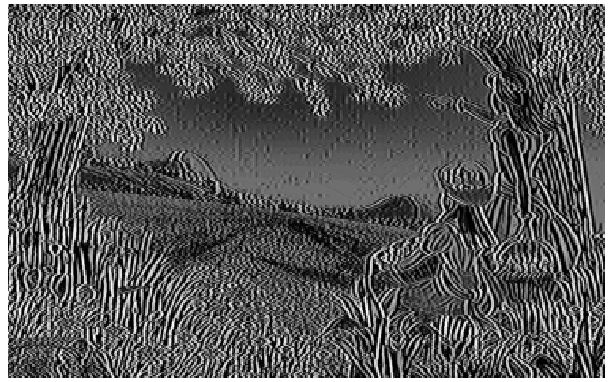
WTA Image

Normalized WTA Image



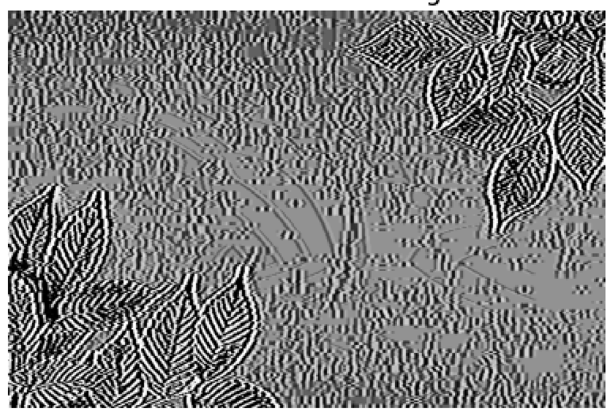
WTA Image

Normalized WTA Image

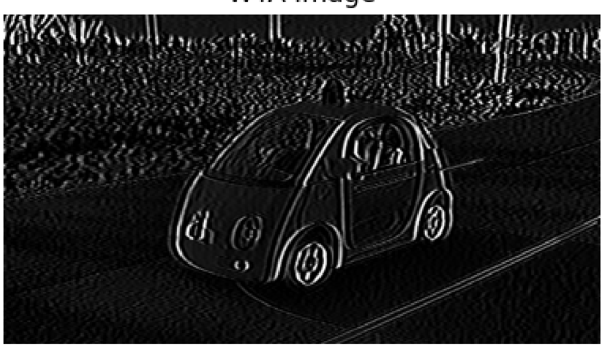


WTA Image

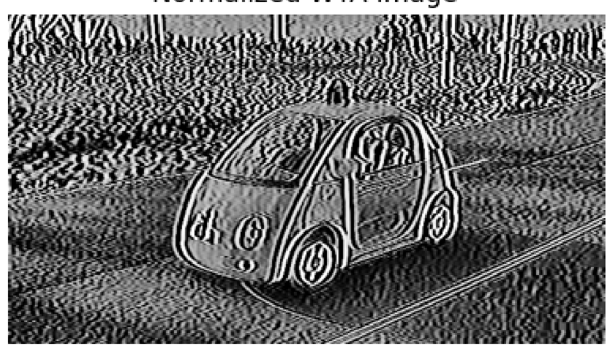
Normalized WTA Image



WTA Image



Normalized WTA Image



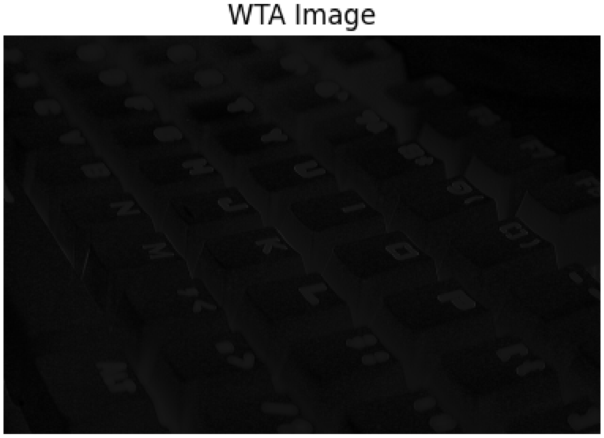
WTA Image

Normalized WTA Image

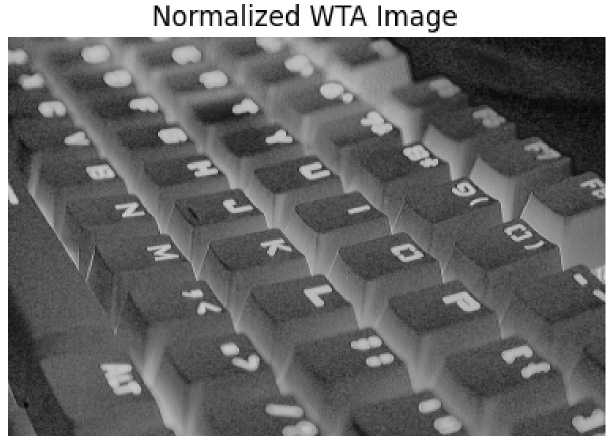
WTA Image

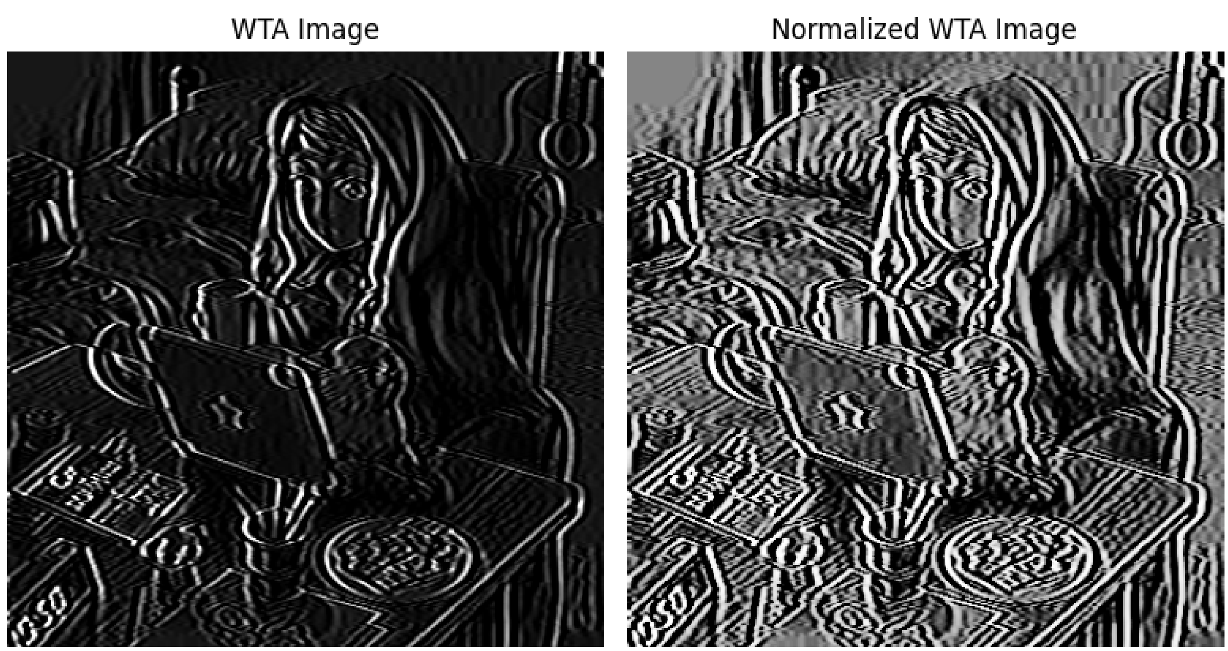
Normalized WTA Image

WTA Image



Normalized WTA Image





Observation from output:

from the above images, we can see that, we can see the same pattern as we saw in the one image. which is the WTA image that emphasizes the key details taken from the intricately filtered images. It draws attention to edges, texture patterns, and other pertinent information, making it simpler to identify key elements of the image. also In the WTA image, strong edges and distinct texture regions are often more noticeable.

features oriented in specific directions are preferentially enhanced, allowing for multi-orientation feature analysis. the normalized WTA image undergoes contrast enhancement through histogram equalization. it is optimized for human perception.

Comparative Analysis

We applied our image processing pipeline to a collection of 10 various nature images in order to evaluate its performance. We displayed the WTA picture and the normalized WTA image for each image as our two main outcomes. Understanding how the pipeline affects feature extraction and visualization across several images is the goal of this comparative investigation.

let's first consider the **complexity of the images**. for images with high complexity such as image1, image, etc. the Gabor-based filtering tends to highlight prominent edges, such as building outlines and object boundaries(ex- image-3). Normalization helped maintain feature visibility even in complex scenes by enhancing contrast.

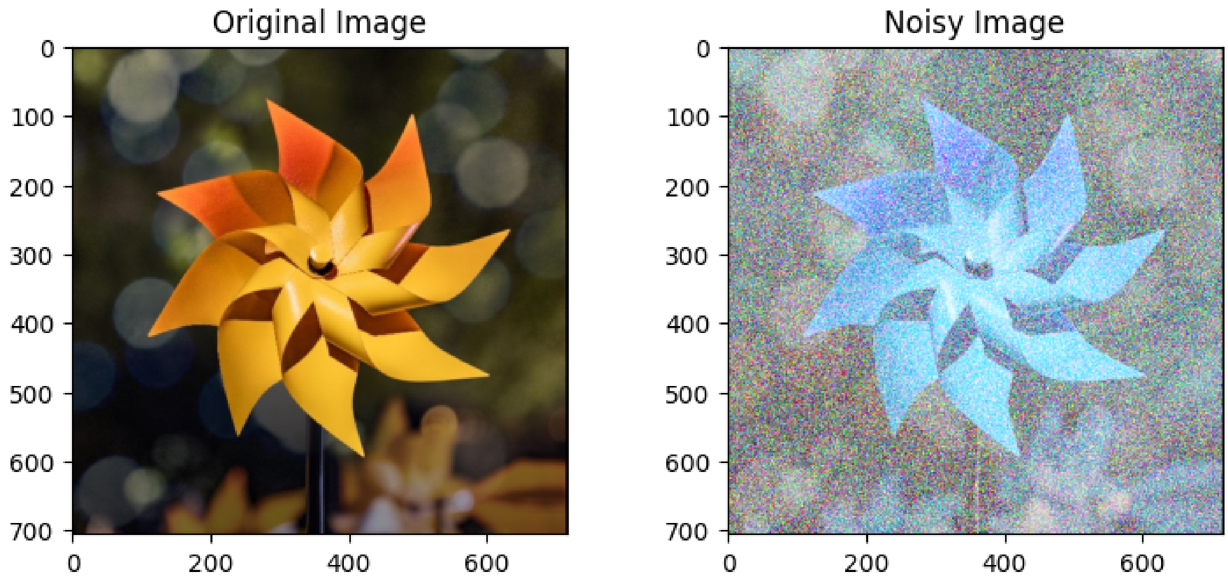
if we see the **Influence of Texture,** Texture-rich images, characterized by intricate patterns or repetitive structures, can impact edge detection differently (ex-image-6). Texture-rich areas in the images were effectively captured by the Gabor filters, producing detailed edge patterns.

and finally, **Effects of Lighting Conditions,** Variations in lighting conditions, including highlights and shadows, can significantly affect edge detection results. Strong lighting variations often resulted in prominent edges and intensity gradients. As we can see in image 2, The Gabor filters effectively detected edges influenced by lighting changes. The WTA operation prioritized edges with the most substantial intensity variations.

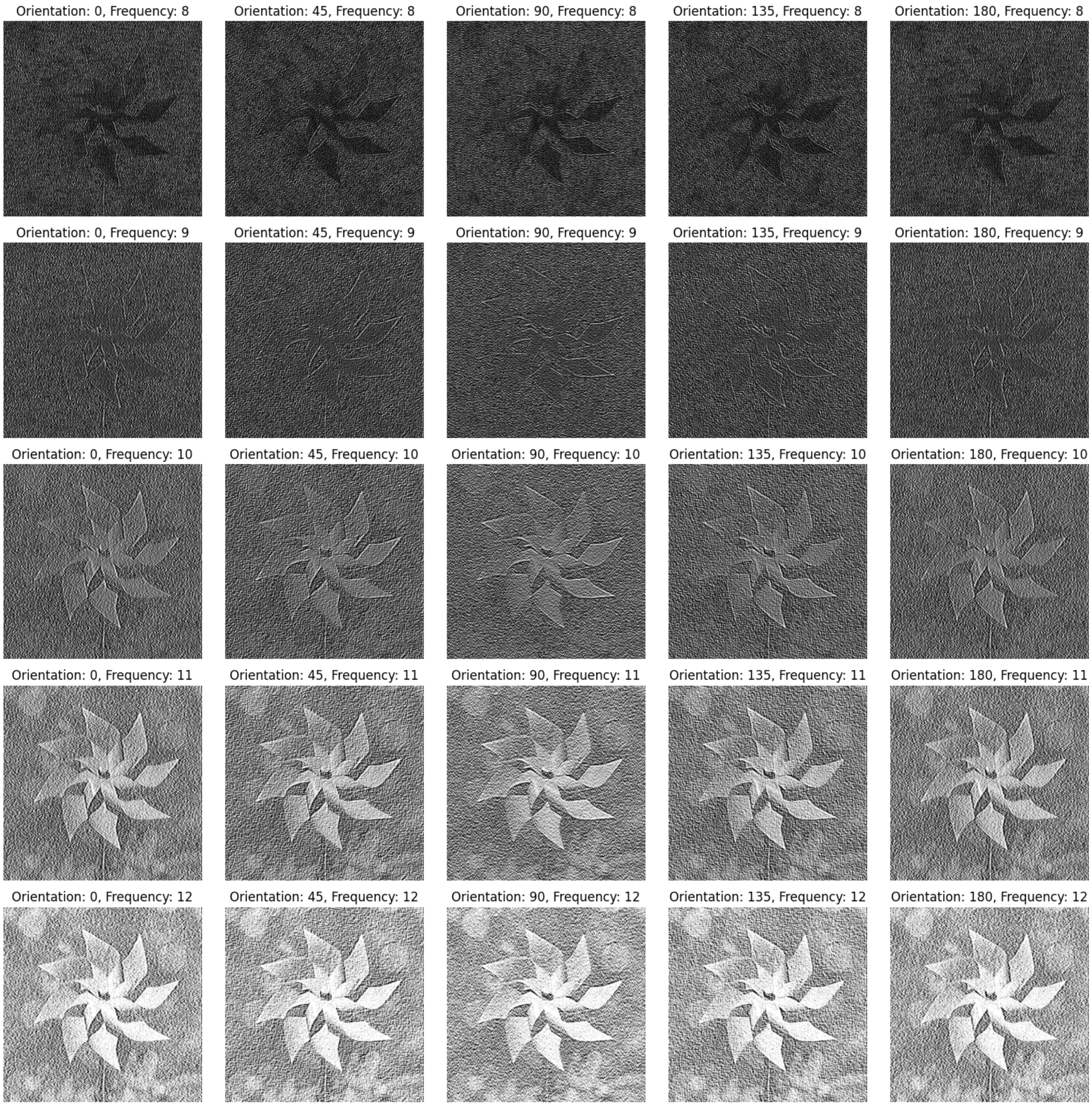
Effects of Noise on Edge Detection

To further evaluate the robustness of the image processing pipeline, we introduced Gaussian noise to one of the loaded images. This noise was added to simulate real-world scenarios where images may be affected by various types of noise, such as camera sensor noise or compression artifacts.

I started by choosing one of the loaded images as the target image for adding noise. mean and standard deviation were the two arguments along with the image used to construct Gaussian noise. and after that, I combined the original image with noise to construct a noisy image.

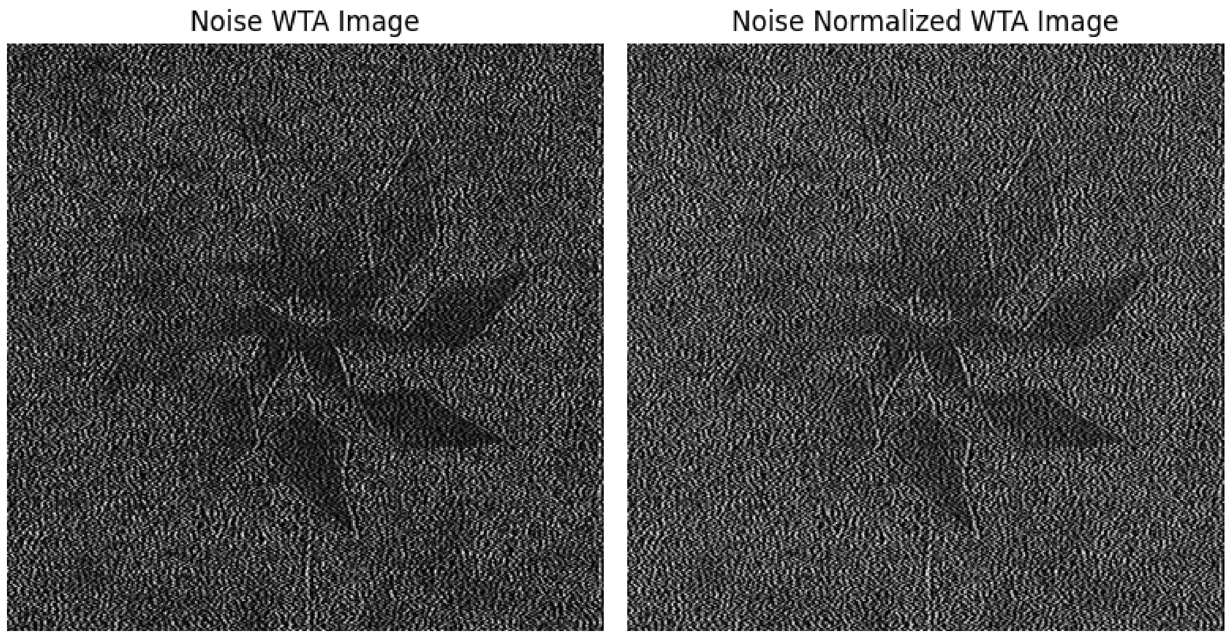


I applied similar techniques used before which included converting to grayscale using Contrast Limited Adaptive Histogram Equalization (CLAHE) and then I applied the edge detection techniques used before such as Gabor filtering and WTA with normalization.

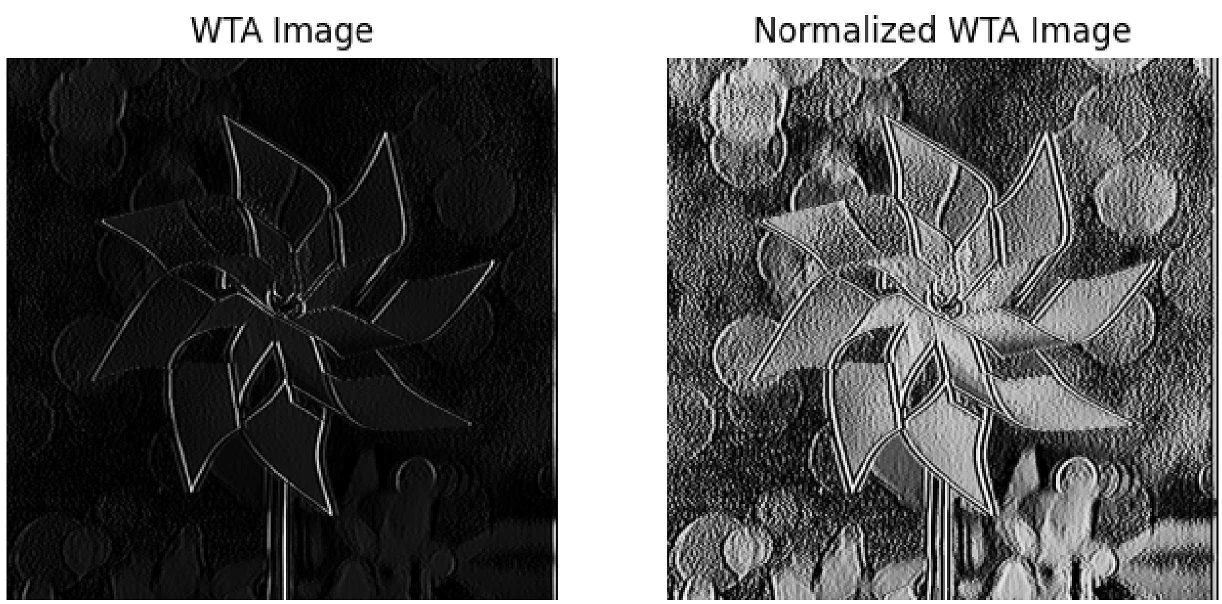


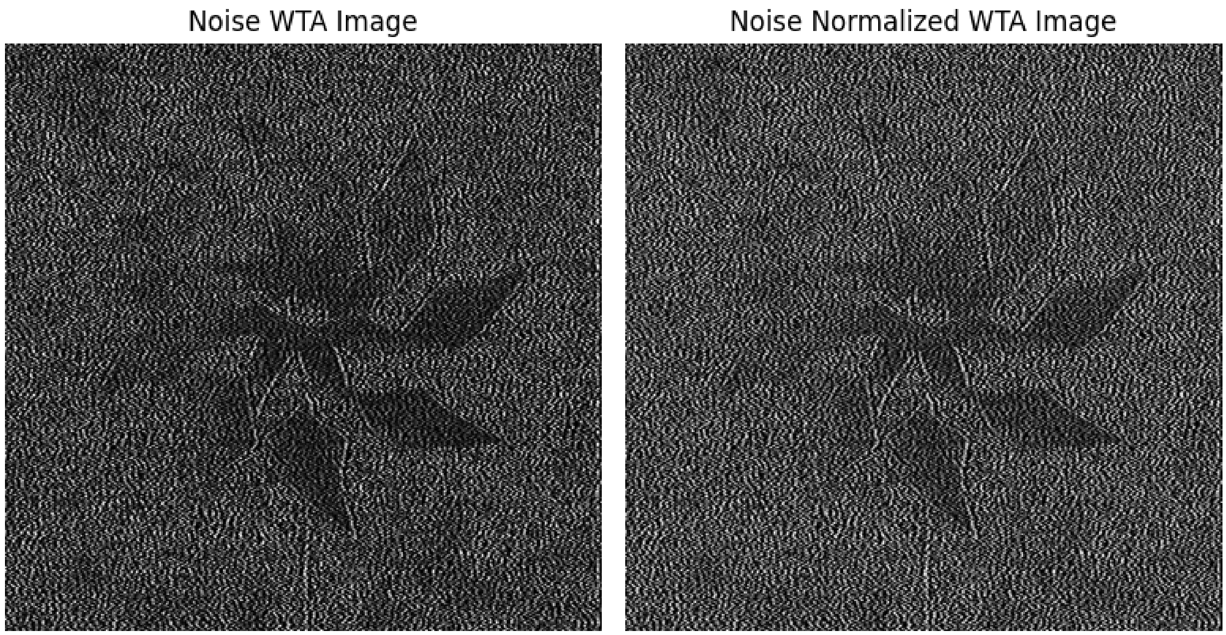
The resulting complex Gabor-filtered images of the noisy input were visualized for each orientation and frequency. These visualizations provide insights into how the noise influences the edge detection process and how the Gabor-based approach responds to noise-contaminated images.

```
Figure - 6
```



Comparison between the original edge detection and noise-influenced edge detection





To assess the impact of noise on edge detection, we compared the results of the original edge detection to those of the edge detection performed on the noisy image.

The SSIM is a metric used to measure the similarity between two images. In our analysis, we applied the SSIM to both the magnitude and phase components of the original edge detection and the noisy-influenced edge detection.

The SSIM score provides a value between 0 and 1, where a higher score indicates greater similarity between the images. A score of 1 suggests that the two images are identical, while lower scores indicate dissimilarity.

For the noisy-influenced edge detection, the SSIM score was calculated as 0.5182, indicating that there is a noticeable difference between the original edge detection and the noisy-influenced edge detection. This result aligns with our visual observations, where edges in the noisy image were less distinct.

observation:

It's important to draw attention to the visual distinctions between the original edge detection and the noisy-influenced edge detection in addition to the quantitative measures. As previously indicated, the noisy image presented difficulties, making edges less clear and obvious.

When the original edge detection and the noisy-influenced edge detection were compared visually, it became clear that the original edge detection had more prominent and clearer edges than the noisy image did. The visual inspection confirms that the edges in the noisy image were less pronounced and clear. This analysis underscores the importance of noise reduction techniques when dealing with real-world images to ensure accurate edge detection.

Pros and cons of Gabor-based edge detection

Pros:

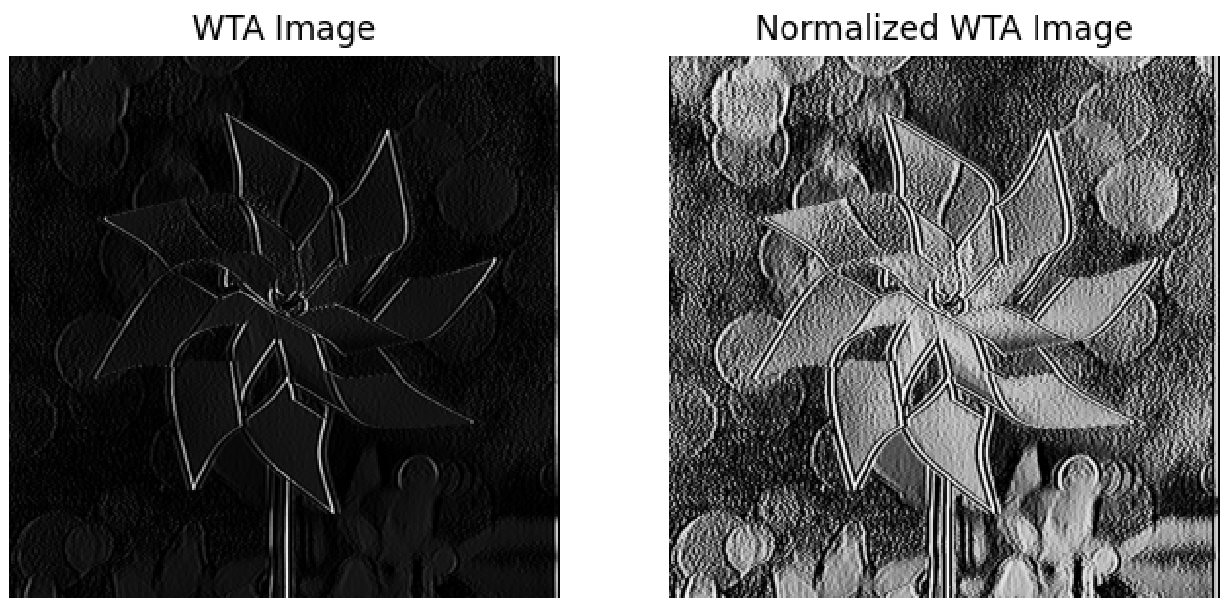
- Capable of accurately identifying edges at various scales and orientations.
- The preservation of texture information in photographs is advantageous for complex textures.
- Handles complex edge structures and junctions with Complex Edge Detection.
- Adapts to changes in lighting conditions and is invariant to illumination changes.
- Offers magnitude and phase data for in-depth study.

Cons:

- necessitates careful parameter adjustment; poor decisions can result in decreased performance.
- computationally expensive, particularly for huge pictures or numerous orientations/frequencies. intensive.
- Noise-sensitive, requiring noise-cancelling methods.
- May have difficulty dealing with sophisticated non-linear structures.
- Increasing scale may result in decreased spatial resolution.

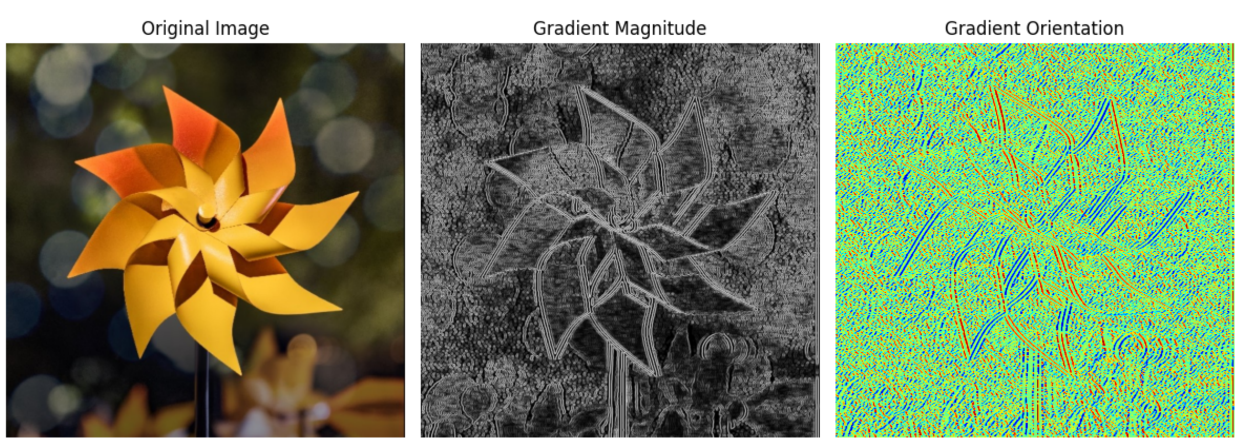
Part-5

Visualization to highlight edges



gradient magnitude and orientation maps

- We generated gradient magnitude and orientation maps from the normalized WTA image.
- The gradient magnitude map highlighted the strength of edges and features.
- The gradient orientation map depicted edge directions.
- These visualizations improved interpretability, aiding feature analysis.
- These techniques were consistently applied to all images for comprehensive analysis, enhancing the understanding of edge detection results.



effectiveness of visualization techniques

Gradient Magnitude Map:

- Effectively highlights regions with pronounced edges and features.
- Effectively emphasizes regions with distinct edges and features.
- Particularly useful for locating things with strong contrast.
- shows the intensity of changes in pixel values across the image, which helps to highlight edges and features.

Gradient Orientation Map:

- shows the direction of changes in pixel values across the image, which helps to understand the orientation of structures and shapes.
- gives information about the direction of edges and features.
- useful for figuring out how the structures in the image are oriented.

These visualization techniques provide a thorough understanding of the structure of the image.A comprehensive comprehension of feature locations and orientations is provided by the combination of gradient magnitude and orientation maps.

Conclusion and learnings

We were introduced to the field of image processing through this assignment, where we used a variety of tools to find edges and features in real-world pictures. Here is a quick summary of what we learned:

Gabor-based filters have been our go-to option for catching edges, and they shine. These filters demonstrated their prowess in improving edge identification by taking orientation and frequency into account.

The Influence of Image Characteristics: We found through comparison study that image complexity, texture, and lighting significantly influence the results of edge detection. Because no two photographs are alike, our processes adjusted accordingly.
Visualization Unlocks Insights: Visualization approaches, particularly gradient magnitude and orientation maps, enhanced our experience. These technologies revealed the intricate details of the photos, enabling us to determine the direction and strength of the edges.

It's important to draw attention to the visual distinctions between the original edge detection and the noisy-influenced edge detection in addition to the quantitative measures. As previously indicated, the noisy image presented difficulties, making edges less clear and obvious.

the WTA image that emphasizes the key details taken from the intricately filtered images. It draws attention to edges, texture patterns, and other pertinent information, making it simpler to identify key elements of the image. On the other hand, the normalized WTA image aims to strike a balance between feature enhancement and contextual preservation, resulting in a more visually informative representation with improved contrast and detailed features.

References:

https://journals.sagepub.com/doi/pdf/10.1177/0963721

https://en.wikipedia.org/wiki/Winner-take-all_(computing)

https://medium.com/@anuj_shah/through-the-eyes-of-gabor-filter-17d1fdb3ac97

https://www.baeldung.com/cs/gradient-orientation-magnitude#:~:text=Gradient magnitude is used to,into multiple regions or segments.