Scania Task - Perception Testing

Nikhil Challa

October 11, 2023

1 Introduction

The objective is to categorize images based on environmental conditions such as fog, snow, rain, and clear sky. This can be framed as a classification challenge where each image is assigned one of these attributes. In essence, the proposed techniques aim to analyze the images for all three conditions (fog, snow, and rain), ultimately choosing the most fitting one. If none of these attributes are a suitable match, the image is classified as a clear sky. The provided code employs image analysis methods to determine these attributes.

Given that our objective is to detect active snow rather than simply snow in the background, the msot of the available labeled dataset images are unsuitable. Additionally, many of these images do not accurately represent the images encountered in autonomous vehicles (AV). Therefore, significant efforts were undertaken to curate a more suitable image dataset. In addition to leveraging existing Kaggle datasets and images from Scale AI, images were also gathered from other sources such as PEXEL and SHUTTERSTOCK. These images are freely accessible for academic purposes, but they should not be used for commercial or distribution purposes and kindly request that you refrain from utilizing these images beyond testing the source code.

The source code is written in MATLAB, with utilized built-in functions listed in the appropriate section. To elucidate the source code, a concise feature summary is provided for each attribute. A pair of images, one that contains and the other that does not contain the attribute is provided for comparison. Drawing from the insights in [1], it's evident that Lidar technology encounters challenges in adverse weather conditions, necessitating additional precautions in object detection. The source code I've developed relies on the richness of image data to enhance the accuracy of fog and snow detection, thereby improving certainty in the process.

2 Fog Detection

Referring to the insights in [2], [3], several key characteristics of images affected by fog can be outlined. Color images tend to exhibit a reduced color distribution across the R-G-B spectrum, and each channel typically displays lower intensity, resulting in a prevalence of gray tones. To address this, we utilize the dehazing source code provided by the same reference author, which can improve both color distribution and intensity. Dehazing can occasionally introduce artificial color tones, unless there's prior knowledge that the image is inherently foggy. Since such prior knowledge is unavailable, the dehazing process is selectively applied to objects detected within the image. For object detection, a built-in MATLAB function called *yolov4ObjectDetector* is employed, which is derived from the YOLO4 publication [4]. However, it's worth mentioning that the defogging function may encounter challenges when processing images of small sizes. To mitigate this issue, the bounding box size is restricted to a minimum of 50 pixels on both axes.

Figure 1: Example of fogging image with object detection

Before and after applying the defogging process to each object in the image, the standard deviation (std) of each pixel. Subsequently, the mean and the std of these pixel std values is computed. The change in this mean and std value is evaluated and the results described in the following pages. Assuming the image has fog attributed; if the object possesses sufficient color diversity, a noticeable increase in both the mean and the std of these values are observed.

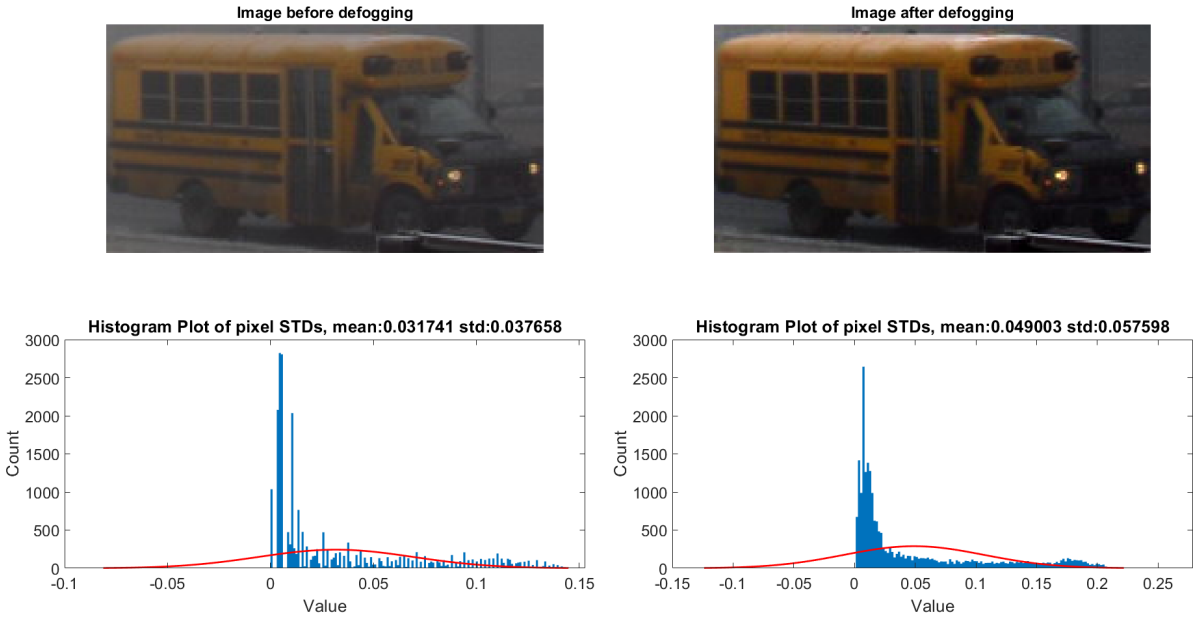

Figure 2: Foggy - Histogram plots for mean and std of the std of each pixel for Object 1

All the objects identified within the image are collectively plotted. To analyze the impact of dehazing, vectors are created for each object, connecting a blue dot representing the pre-dehazing point to a red dot representing the post-dehazing point. The axes are represented in complex number cartesian form. In cases where multiple objects are detected in the image, these vectors are aggregated, and their mean is computed (by dividing the resulting vector by the number of objects). The absolute value of this resultant vector is then employed to assess the likelihood of fogging. A formal methodology for determining the probability of each environmental attribute has not been established in this context. Instead, a straightforward thresholding approach is utilized to make these determinations.

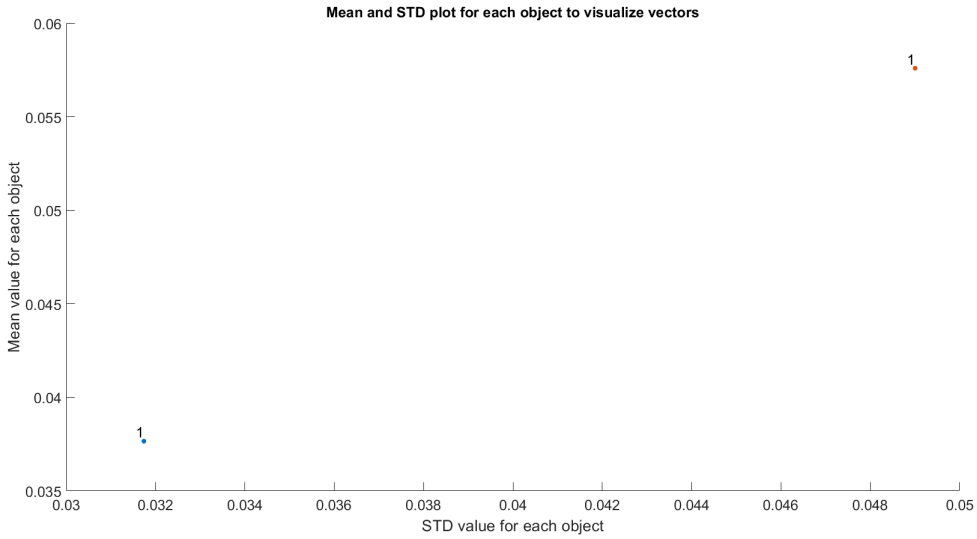

Figure 3: Foggy - Plotting mean and std for each object

The aforementioned steps will be replicated for an image that does not contain any fog.

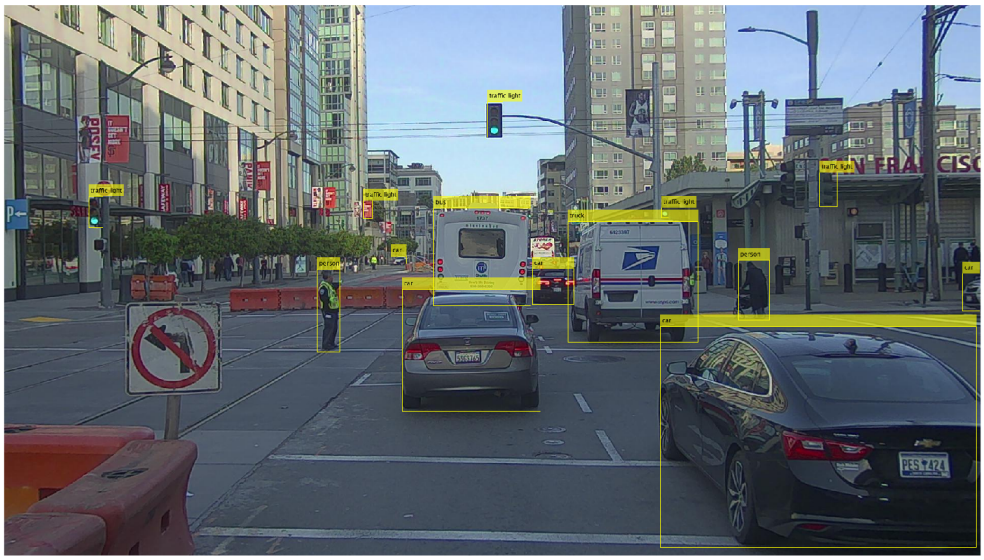

Figure 4: Example of no fogging image with object detection

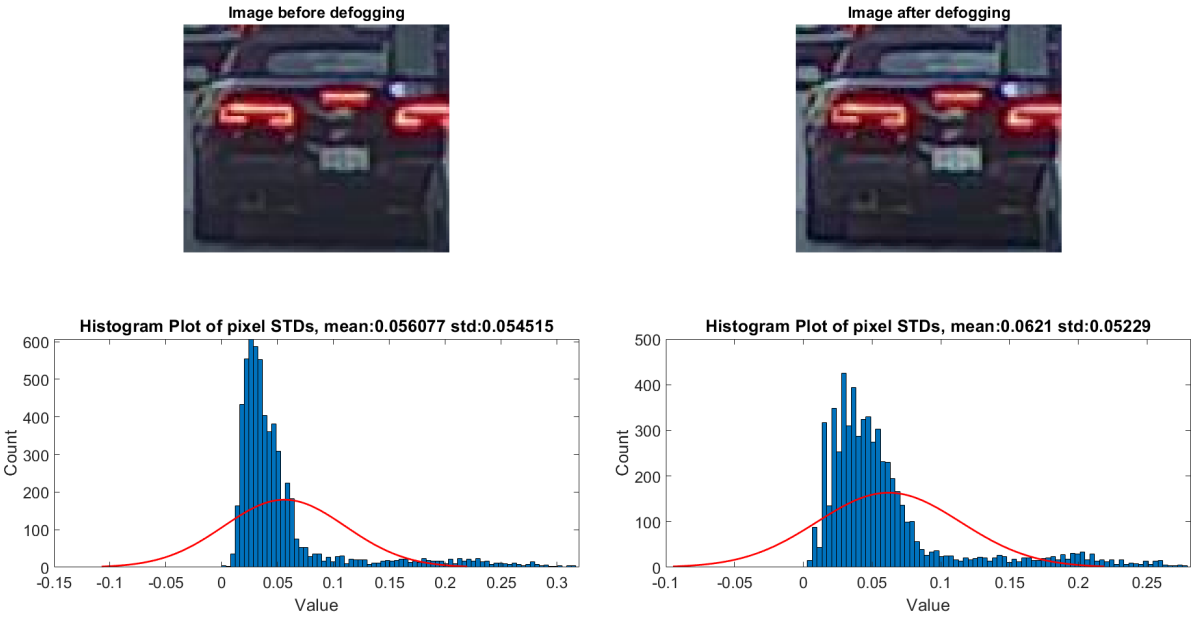

Figure 5: Not foggy - Histogram plots for mean and std of the std of each pixel for object 1

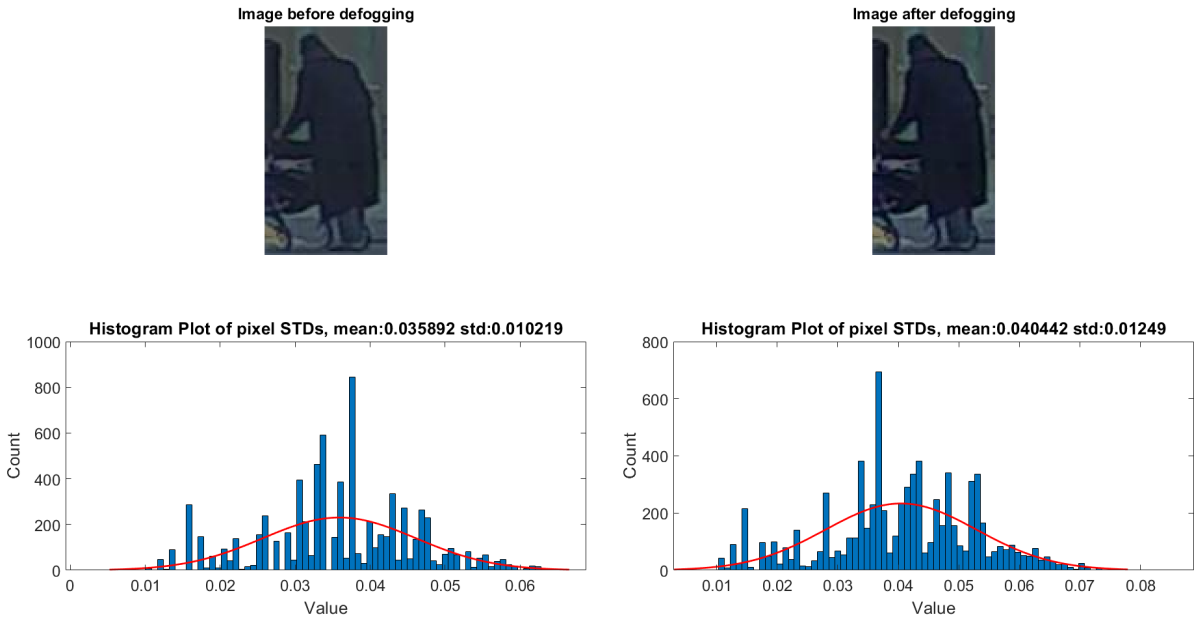

Figure 6: Not foggy - Histogram plots for mean and std of the std of each pixel for object 2

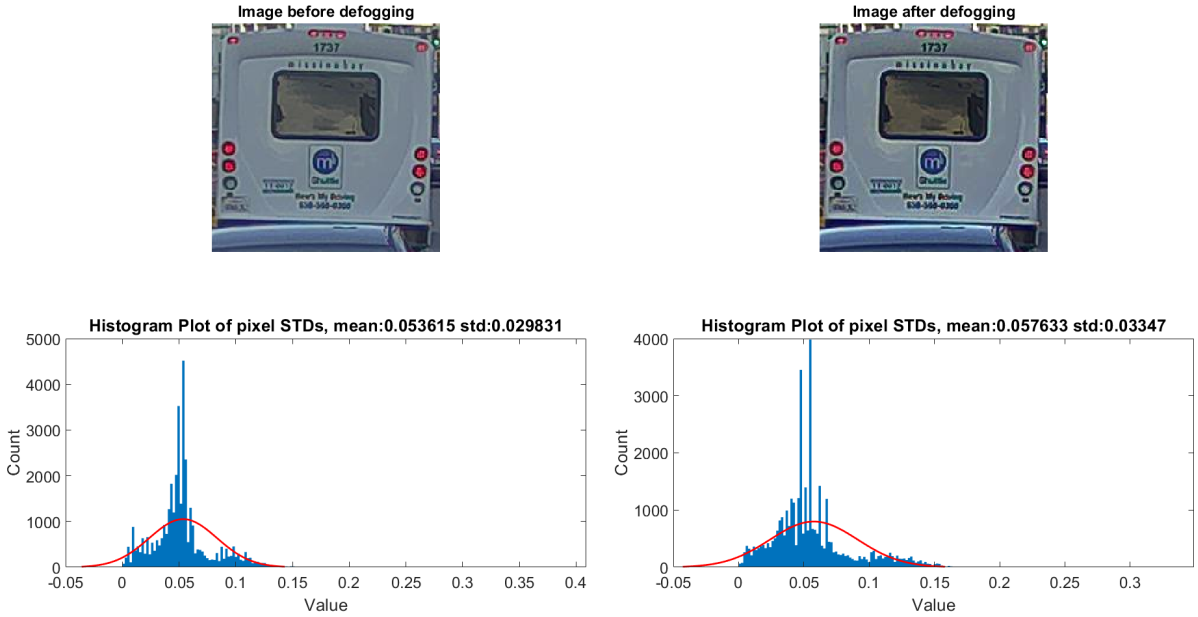

Figure 7: Not foggy - Histogram plots for mean and std of the std of each pixel for object 3

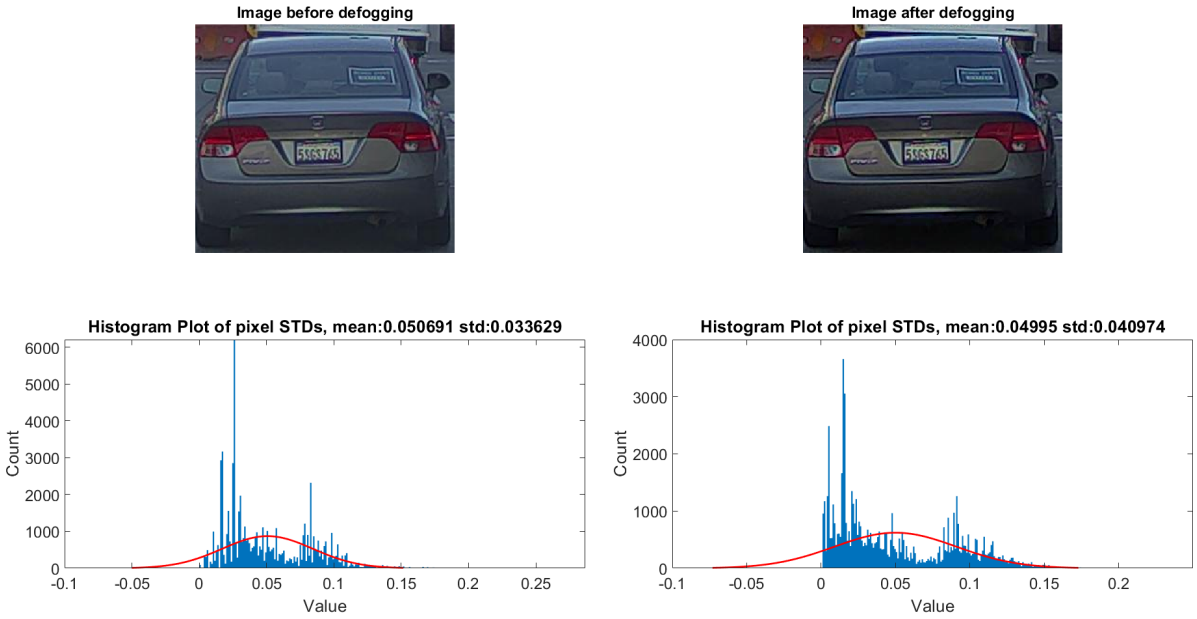

Figure 8: Not foggy - Histogram plots for mean and std of the std of each pixel for object 4

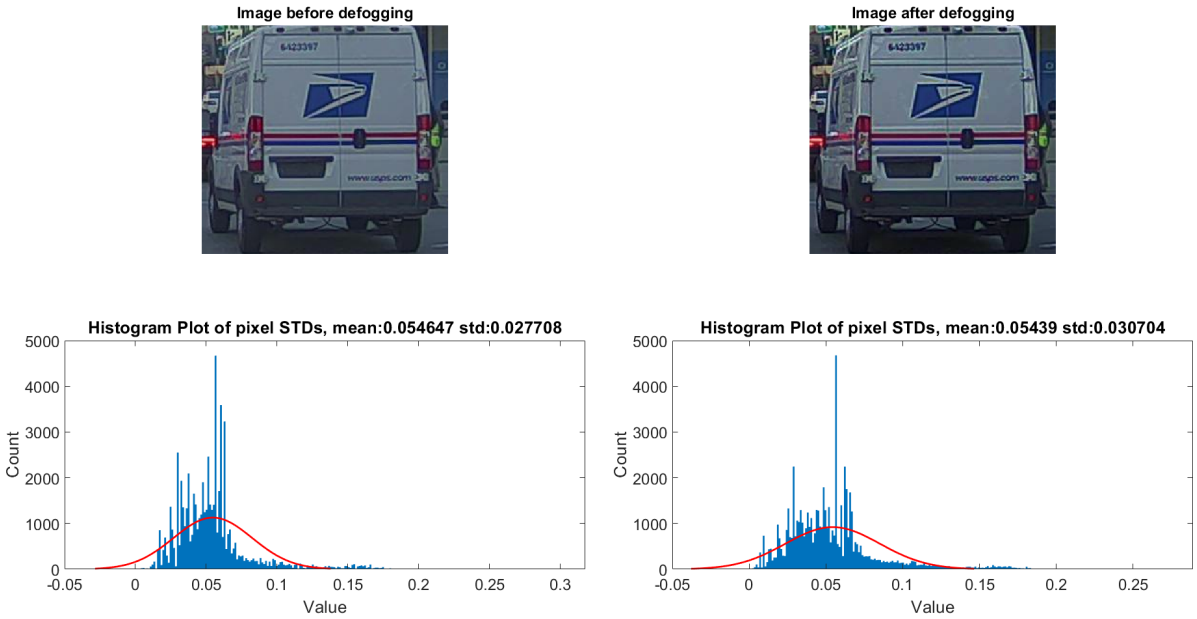

Figure 9: Not foggy - Histogram plots for mean and std of the std of each pixel for object 5

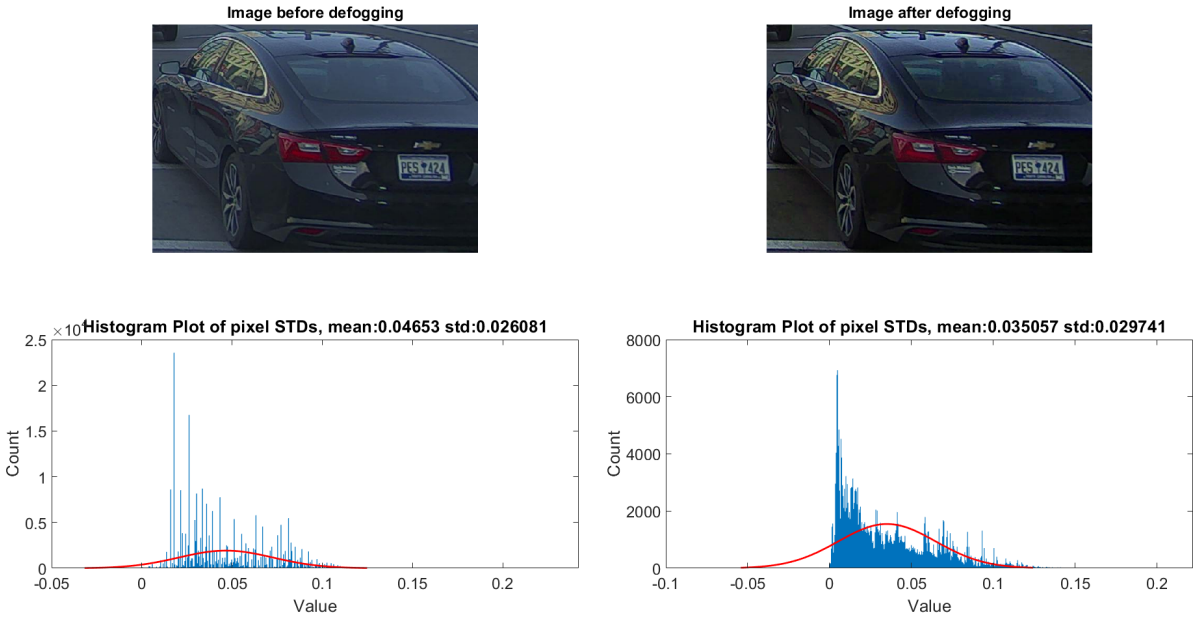

Figure 10: Not foggy - Histogram plots for mean and std of the std of each pixel for object 6

For this image, the vectors do not consistently align in the positive direction. For instance, the vector associated with object 1 generally points almost horizontally but slightly in the negative y-axis direction, while the vector for object 3 consistently points in the positive direction for both axes. This variability in vector direction is a consequence of attempting to dehaze an image that lacks fog. In such cases, the dehazing process doesn't consistently guarantee an increase in mean and std values.

When vectors with somewhat random directions are aggregated, the resulting vector tends to be small. Calculating the mean of these vectors further diminishes their overall intensity. To account for the non-alignment in the positive direction, an additional condition has been integrated into the code. Specifically, if the accumulated vector does not align with the positive direction for both axes, it is reset to zero.

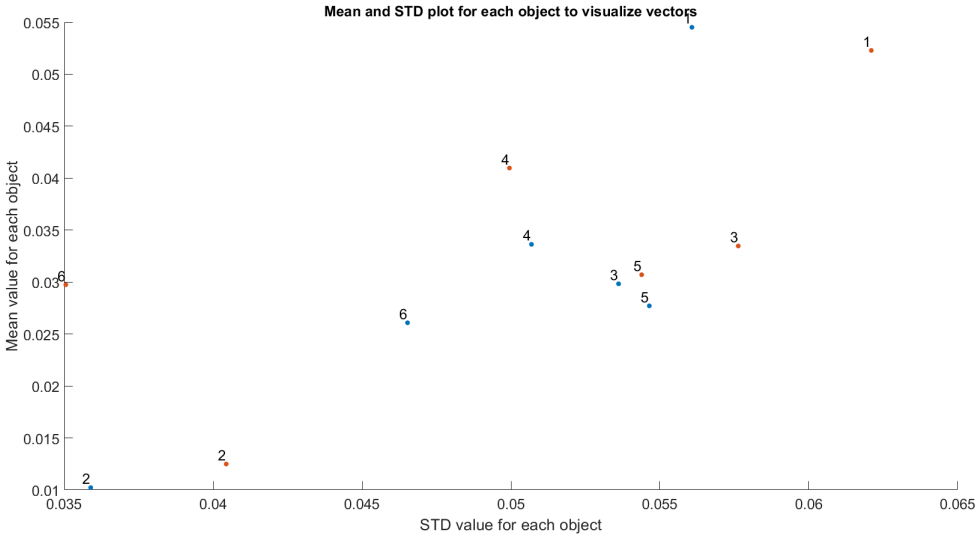

Figure 11: Not foggy - Plotting mean and std for each object

Following evaluation across multiple images, a threshold value of 0.01 has been chosen. If the magnitude of the resulting vector exceeds this threshold (i.e., it is greater than 0.01), the image is classified as containing fog.

3 Snow Detection

For snow detection, there were complexities with correctly classifying images that had snow in the background but no active snowing vs active snowing. For reliable detection, the image needs to contain objects. Again, the same YOLO4 method was utilized for this detection step. The following steps are performed for each of the boxed object images.

- Identify pixels that have intensity greater than 128 for all three channels and the standard deviation value of the three channel (same pixel) is less than 20
- Use *bwareafilt* to remove pixels whose connected shape exceeds the predetermine size. This is to discard background white images (snow collected on the ground or on trees) or obejcts that are near shades of white.
- Count remaining pixels in the boxed images for calculation percentage of snow particles in the boxed image ($TotalCount$).
- Perform image closing using *imclose* to discard streaks that are too close to each other. These streaks are usually from the background and not really contributing to actual snow particles.
- Count the remaining pixels that are marked as snow particles ($NumSnowPixels$).
- Calculate ratio of ($NumSnowPixels$) to ($TotalCount$).
- If more than one object is detected in the image, the mean values of all the ratios can be taken as the final value.

Figure 12: Example of an obejct detected image with active snowfall

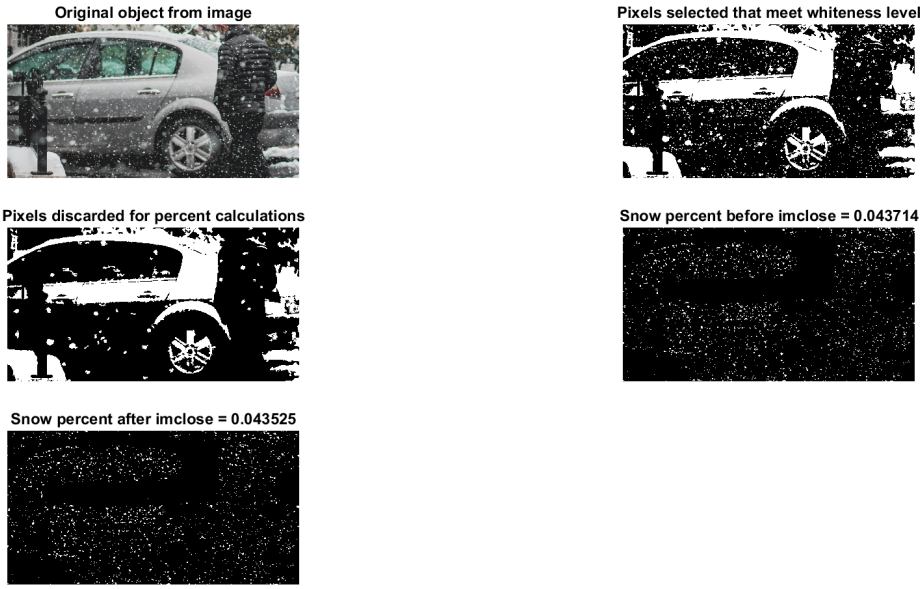

Figure 13: Snow - Object 1 with percentage calculations

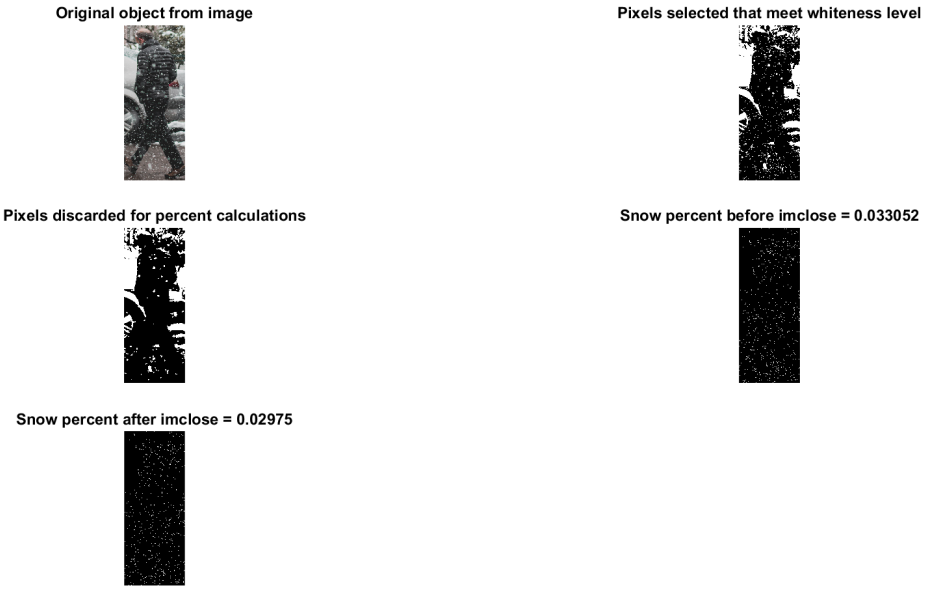

Figure 14: Snow - Object 2 with percentage calculations

The identical procedure is replicated for an image that does not exhibit active snowfall.

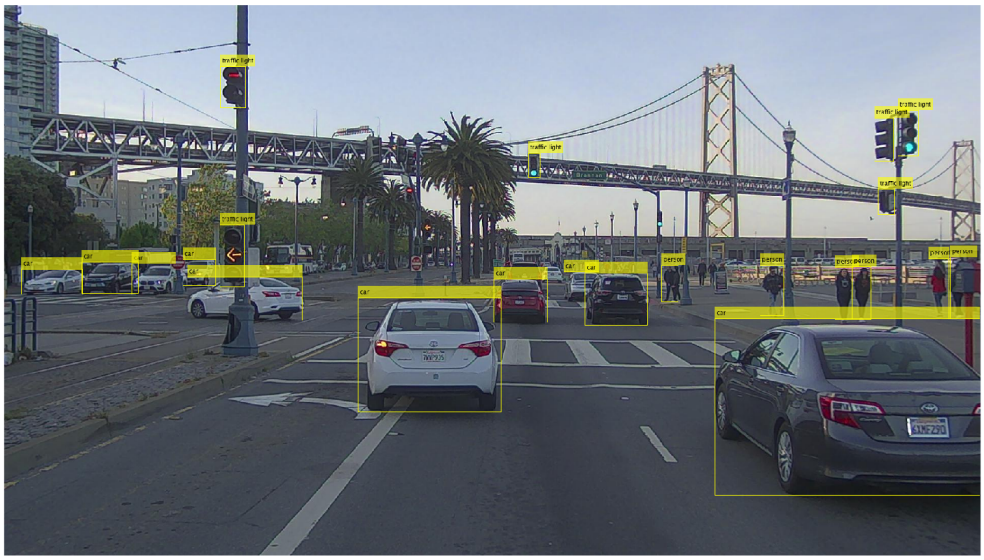

Figure 15: Example of an obejct detected image with no active snowfall

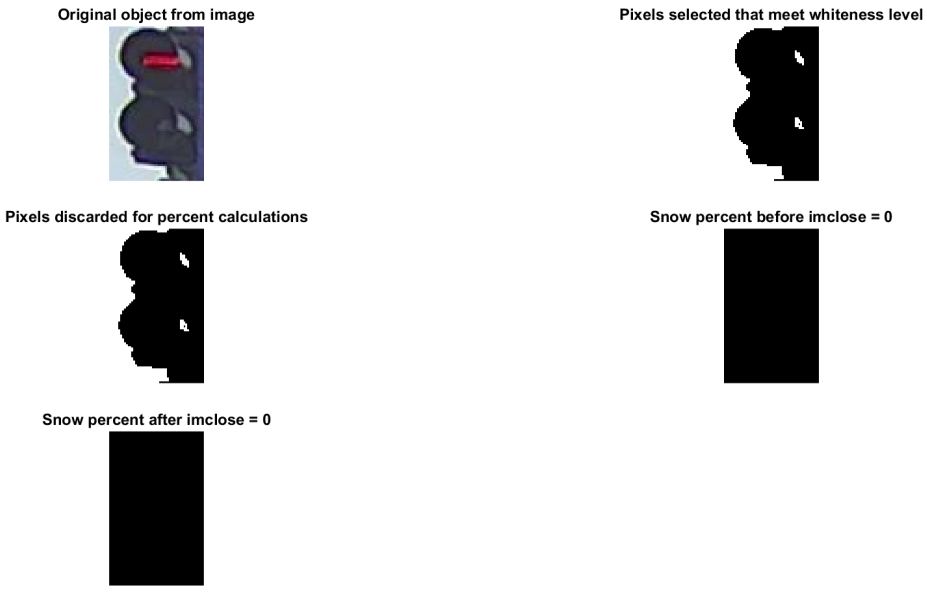

Figure 16: No snow - Object 1 with percentage calculations

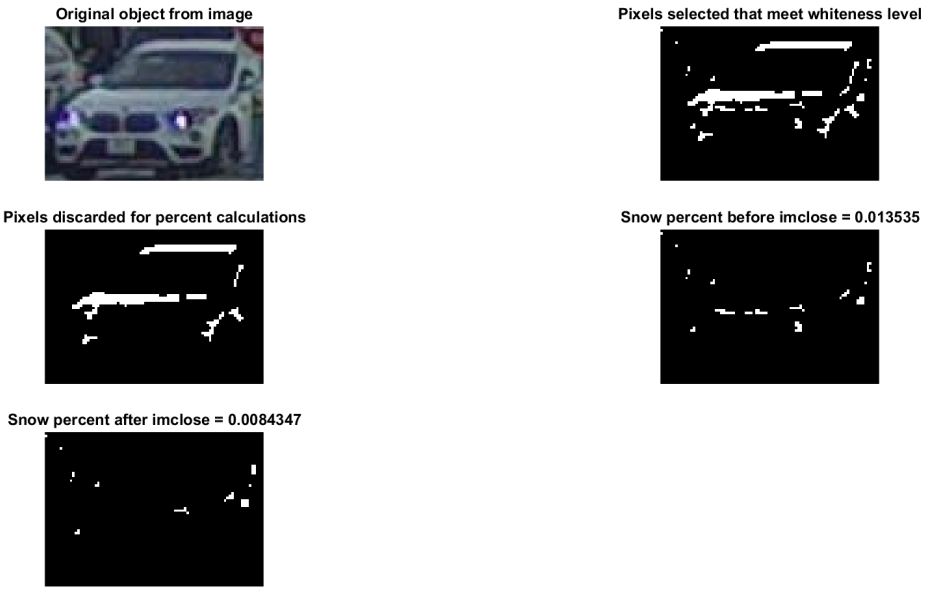

Figure 17: No snow - Object 2 with percentage calculations

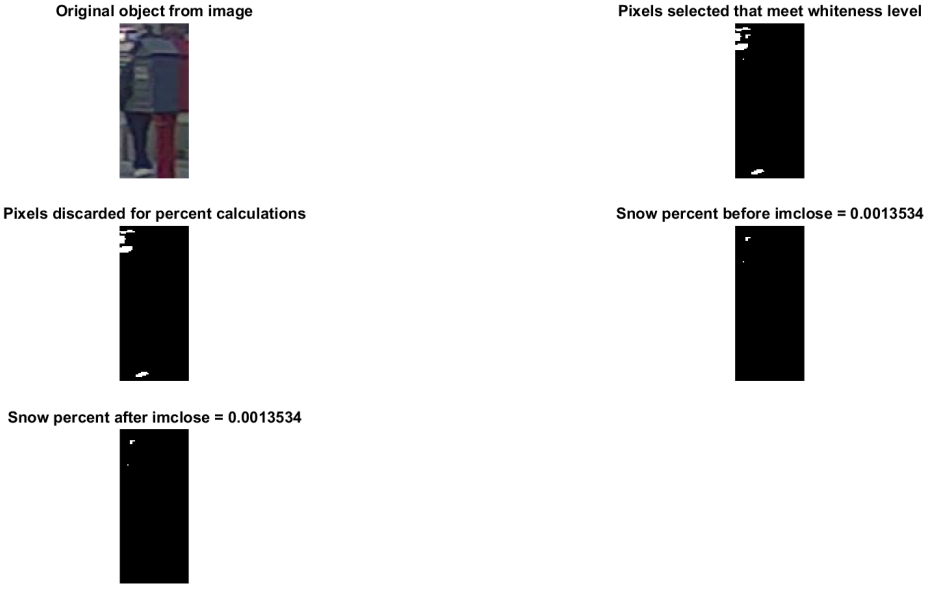

Figure 18: No snow - Object 3 with percentage calculations

After thorough evaluation involving multiple images, a threshold value of 0.01 has been determined as the suitable criterion. If the resulting percentage of pixels that exhibit active snowfall surpasses 1%, the image is classified as having active snowfall.

4 Rain Detection

Drawing upon insights from [5], properties of rain droplets on images can be leveraged. To elaborate, rain introduces noticeable gradients in the images and exhibits a high reflective index upon striking surfaces. This heightened reflectivity is attributed to light bouncing off surfaces containing water, and the ripples generated by raindrops hitting surfaces also contribute to the observable gradients. In the context of rain detection, the entire image is employed for analysis, as rain's impact can manifest across the entire frame.

In addition to the mentioned properties, there were also observations related to rain detection. Notably, it was observed that the whiteness of ripples produced when raindrops hit a surface was more pronounced compared to raindrops in mid-air. To account for these distinctions, two distinct methods were employed.

- Select pixels that meet certain grayness level (each channel value between 100 and 200, and the standard deviation less than 60).
- Select pixels that meet certain gradient intensity (greater than or equal to 50).

Using an aggregate of pixels with the above two methods, a more descriptive representation of rain is covered. To avoid falsely detected gradients due to natural color imbalances such as dirt on the road, grass, and trees, the spatial distribution of the selected pixels is evaluated.

By combining the information obtained from the two aforementioned methods, a more comprehensive and descriptive representation of rain is achieved. To mitigate false gradient detections caused by natural color imbalances, such as dirt on the road, grass, and trees, the spatial distribution of the selected pixels is assessed, allowing for more reliable rain detection.

After consolidating the pixels that may represent rain or rain-related effects, the entire image is divided into 3x3 subsections. In each of these subsections, the number of pixels that meet the rain criteria is counted. Subsequently, the standard deviation of the pixel count from each subsection is computed. This standard deviation value provides an indication of the evenness in the distribution of pixels. An image containing rain should ideally exhibit a relatively uniform pixel distribution across the entire image space.

Additionally, the percentage of selected pixels, relative to the total number of pixels in the entire image, is calculated. This metric helps quantify the extent of rain-related pixel presence within the image.

Figure 19: Example of an image with active rain fall

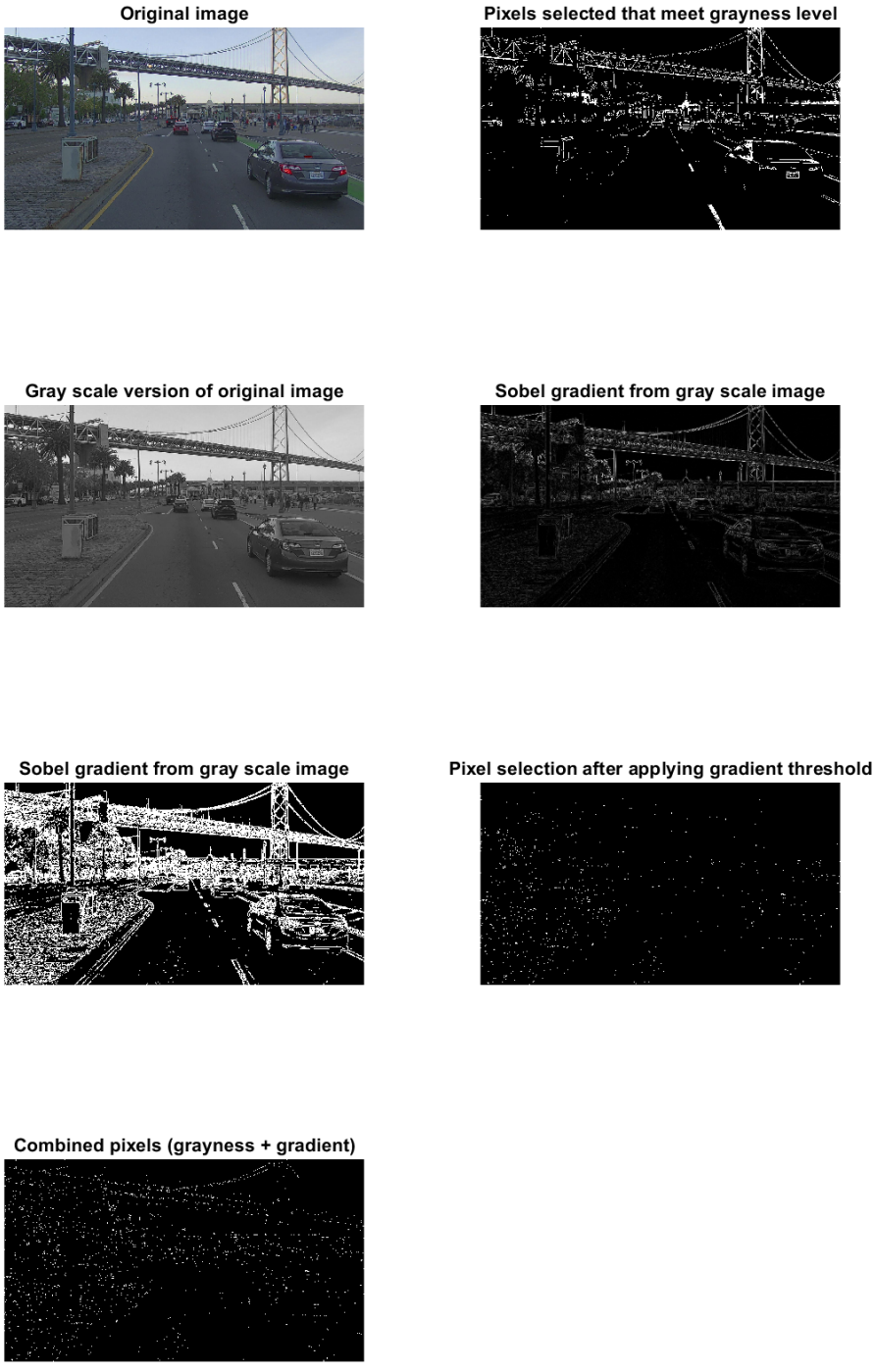

Figure 20: Example of an image with no active rain fall

Following a comprehensive evaluation involving multiple images, specific threshold values have been determined. To classify an image as containing rain, the selected pixel percentage relative to the entire image must exceed a minimum threshold of 0.02. Additionally, the standard deviation of the number of pixels within each subsection should be less than 1000. These threshold values help ensure reliable rain detection across various images.

5 Source Code Details

To simplify the process, a single file named **ClimateClassification.m** is provided for testing the classification results. The input image or its location can be specified at the top of the script. Several examples are included as comments, but any new image can also be provided for testing. The results will be displayed in the command window.

The **labelpoints.m** script was sourced from the labelpoints MATLAB File Exchange submission, while the contents within the **Dark-Channel-Haze-Removal** folder were obtained from the Single Image Haze Removal Using Dark Channel Prior MATLAB File Exchange submission. As for the source of images in the other folders, these details have been previously addressed in the introduction section.

6 Conclusion and Summary on the Results

Among the environmental attribute detection algorithms, the fog detection algorithm is considered the least reliable, while the snow detection algorithm is the most reliable. In scenarios where more than one attribute is selected, the following hierarchy is followed...

- If snow attributed is selected, the final conclusion will indicate *## Final Conclusion : This image has snowfall ##*
- If rain attributed is selected, the final conclusion will indicate *## Final Conclusion : This image has rainfall ##*
- If fog attributed is selected, the final conclusion will indicate *## Final Conclusion : This image has fog ##*
- If none of the three attributes (rain, snow, fog) are selected, the final conclusion will indicate *## Final Conclusion : This image has clear weather ##*
- In the event of lack of objects detected in the image, an additional image *Uncertainity in Fog and Snow* will be displayed

References

[1] T. Vattem, G. Sebastian, and L. Lukic, "Rethinking lidar object detection in adverse weather conditions," in *2022 International Conference on Robotics and Automation (ICRA)*, pp. 5093–5099, 2022.

[2] K. He, J. Sun, and X. Tang, "Single image haze removal using dark channel prior," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1956–1963, 2009.

[3] R. Castelli, P. Frolkovic, C. Reinhardt, C. C. Stolk, J. Tomczyk, and A. Vromans, "Fog detection from camera images," pp. 25–43, 2016.

[4] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," 2020.

[5] H. Wang, Y. Wu, M. Li, Q. Zhao, and D. Meng, "Survey on rain removal from videos or a single image," *Science China Information Sciences*, vol. 65, dec 2021.