Raindrop Mask Segmentation/Generation Formal Report Sp. 2025

The goal of this project was to develop an efficient and reliable method to generate labeled data for raindrop segmentation tasks, followed by training a custom machine learning model to perform automatic raindrop detection. This work supports future research projects requiring accurate raindrop mask generation under various environmental conditions.

I began by developing a custom Image Labeling Application using Python, OpenCV, Matplotlib, and the SAM2 segmentation model. Initially, the focus was on creating an efficient manual segmentation tool where the user could select points on raindrops. The application then used SAM2 to generate multiple candidate masks based on the user inputs, allowing for previewing, correction, and final mask saving. Throughout its development, the application evolved to include features such as an interactive GUI for selecting and deleting points, automatic mask aggregation, Gaussian blur preprocessing for smoother masks, and organized output storage to streamline dataset creation.

Using this custom tool, I labeled over 50 high-quality raindrop images and generated corresponding binary mask files. This resulted in a prepared dataset consisting of paired images and masks specifically structured for model training.

Recognizing that manual intervention remained a bottleneck for large-scale data creation, I transitioned to automating the segmentation process further. I researched and implemented the U-Net architecture using PyTorch to create a machine learning model capable of predicting raindrop masks automatically. The model was trained using the generated image and mask pairs with an input size of 256x256 pixels. The Binary Cross Entropy with Logits Loss (BCEWithLogitsLoss) was utilized as the loss function. Training was performed on CPU, and performance was monitored through batch loss tracking and average epoch loss calculations. Initial model predictions demonstrated basic success, though with noticeable limitations in capturing finer detail.

Based on feedback, I initiated an improved sampling technique. I am now dividing images based on a rough estimated horizon line (e.g., y = 200 pixels) to differentiate between sky and water regions. From these regions, I am extracting 256x256 samples to increase data diversity, targeting a total of 200 to 300 samples extracted from various images. Additionally, I am setting aside 10 complete image and mask pairs exclusively for validation, ensuring they remain completely separate from the training dataset. Retraining with this more diverse dataset is currently underway to improve the model's ability to generalize across different backgrounds.

The next phase includes completing the horizon-based sample extraction, retraining the U-Net model with the expanded and diversified dataset, comparing new model performance against initial results, and exploring additional improvements such as Dice Loss and higher resolution training if necessary.