**BINARY MASKS FOR RAIN DATASET**

The dataset consists of corresponding images with and without flash. The goal is to generate binary masks for the rain streaks (drops).

Guidelines:

RGB color space

1. **Color Thresholding:** Rain streaks appear as bright, nearly white lines. In the RGB, you can threshold for high values of R, G and B channels. For example, you could create a binary mask by setting a threshold where all three channels have values above a certain level (e.g., R, G, B > 200). *(Already tried, but also areas from the background with the same values are masked out and it doesn’t work for the no-flash dataset).*
2. **Highlighting Features:** Contrast enhancements make bright areas (such as rain streaks) brighter and dark areas darker, making it easier to distinguish between the streaks and the background. **Contrast Adjustment:** You can use histogram equalization or CLAHE (Contrast Limited Adaptive Histogram Equalization) to enhance contrast.

HSV color space

1. **Brightness Isolation (V Channel):** Rain streaks tend to be brighter than the background, so you can focus on the V (Value/Brightness) channel of the HSV color space. Thresholding this channel for high values can isolate the streaks.
2. **Saturation Filtering (S Channel):** Rain streaks usually have lower saturation because they are nearly white. You can apply a low threshold to the S channel to isolate regions with low saturation.

* Roboflow, Inkscape, GIMP

Once the masks are generated, train Nerfacto in Nerfstudio (once without, once with masks for both scenarios): Anaconda promt, conda activate nerfstudio (conda deactivate to get back to base)

ns-train nerfacto --vis viewer --data "data\_path" --pipeline.model.camera-optimizer.mode off nerfstudio-data --downscale-factor 1 --eval\_mode interval --output-dir output\_path --max-num-iterations=30000

ns-export nerfacto --load-config outputs\path\config.yml --output-dir exports/output/

ns-viewer --load-config "output\_path\config.yml" --vis viewer

With masks: (maybe masks need to be specified in transforms.json for each image)  
ns-train splatfacto --vis viewer -- colmap --data data/dataset --masks-path masks --downscale-factor 1