# Camera Smear Detection

## Geospatial Vision and Visualisation

Alexis Bourdon
Kashish Goyal
Suhail Pallath Sulaiman

# Overview

**Objective**

To detect and then generate a mask of smear on camera lens from a given sequence of images.

**Method**

We used OpenCV library for python to process the images.

**The Code**

Please clone the following repository in the /sample_drive folder
https://github.com/Kashugoyal/camera_smear_detection.git

**Execution:**

```
$ python smear_detection.py <path to sequence folder>
```

(both absolute and relative paths work) - refer Readme for more details

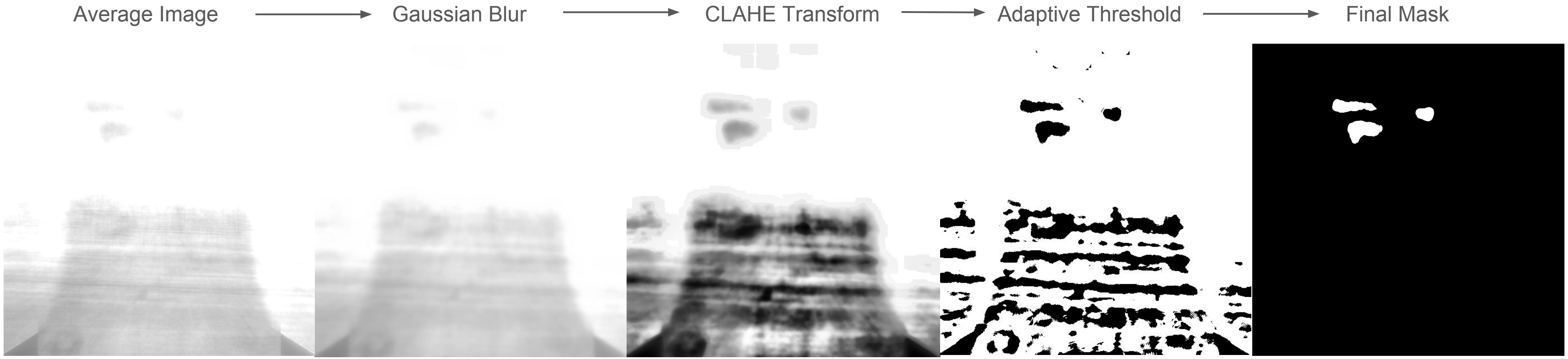
# Our approach

- From the given sequence, the difference of consecutive pairs of images are found.
  - Subtraction of two images blackens the areas in the images which are the same.
- The obtained differences are added together in a weighted average with equal weights (0.00001) and the final average image is calculated.
  - This approach is based on the assumption that the smear is the only area in the image which remains constant throughout the sequence and hence will show up as a dark region in the average.
  - Other areas keep on changing and will be brighter in the average result.
  - Areas that stay constant across many images can cause false positives in the final result.
- After the average calculation, the image is smoothed using gaussian blur.
  - This step helps in removing the noise in the average image thus improving the results obtained in the following steps.

# Our approach (continued)

- The blurred image is then equalised using CLAHE histogram equalization.
  - This improves the contrast in the image. It becomes easy to distinguish the smears from the false positives.
- After this we apply adaptive threshold on the image.
  - This step converts the image to binary.
- Next we find contours in the binary image
  - The contours are stored in an array and we also display the binary image.
- Finally we select the desired contour based on the area enclosed.This gives the final smear detected.
  - Area based filter helps in removing false positives like the sky and the road which show up in the average.
  - A mask is generated and is saved in the working directory.

# Intermediate Steps



Average Image → Gaussian Blur → CLAHE Transform → Adaptive Threshold → Final Mask

# Obtained Image Averages and Masks - Cam 0

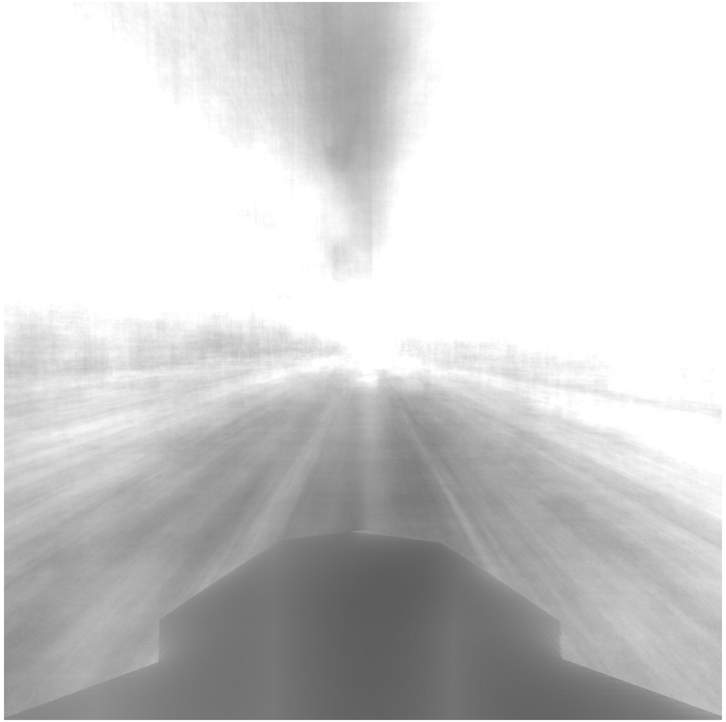


Figure 1: Average of all images

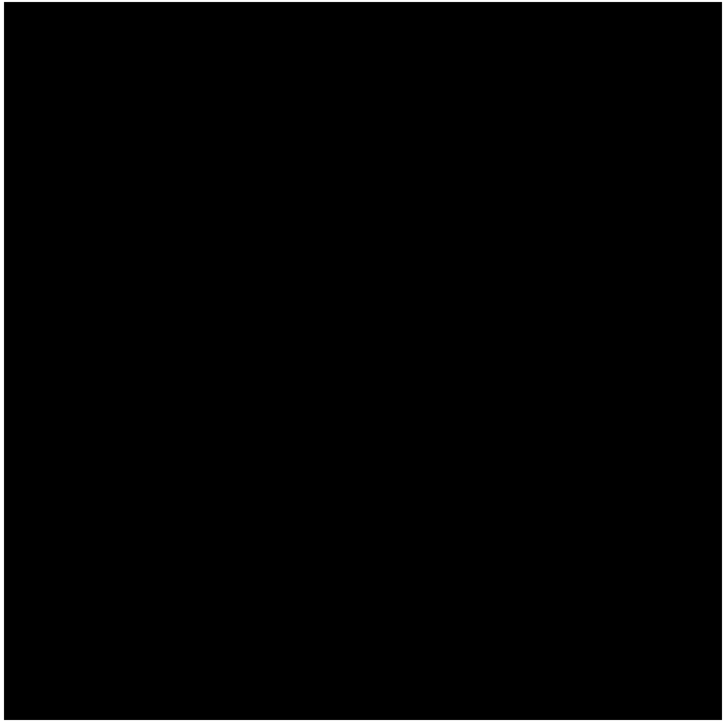


Figure 2: Obtained mask after processing

# Obtained Image Averages and Mask - Cam 1

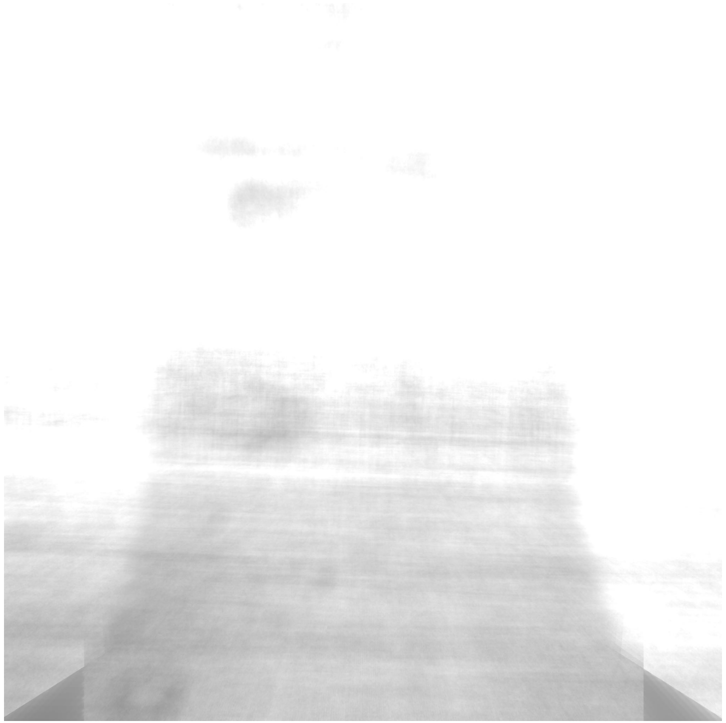


Figure 1: Average of all images

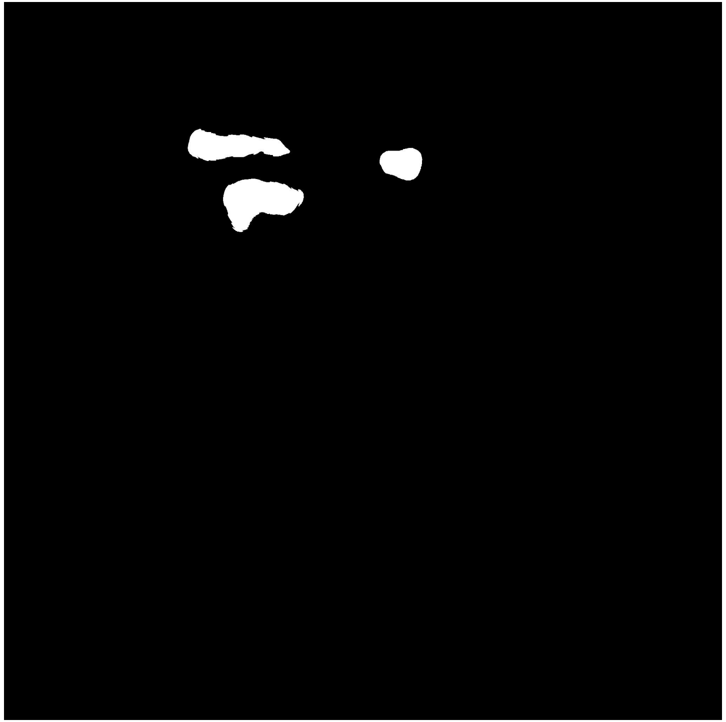


Figure 2: Obtained mask after processing

# Obtained Image Averages and Mask - Cam 2

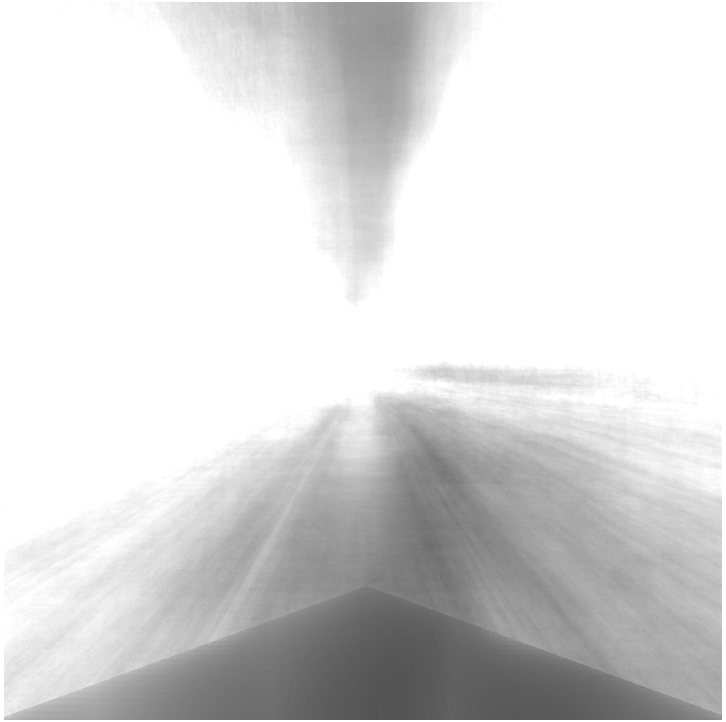


Figure 1: Average of all images

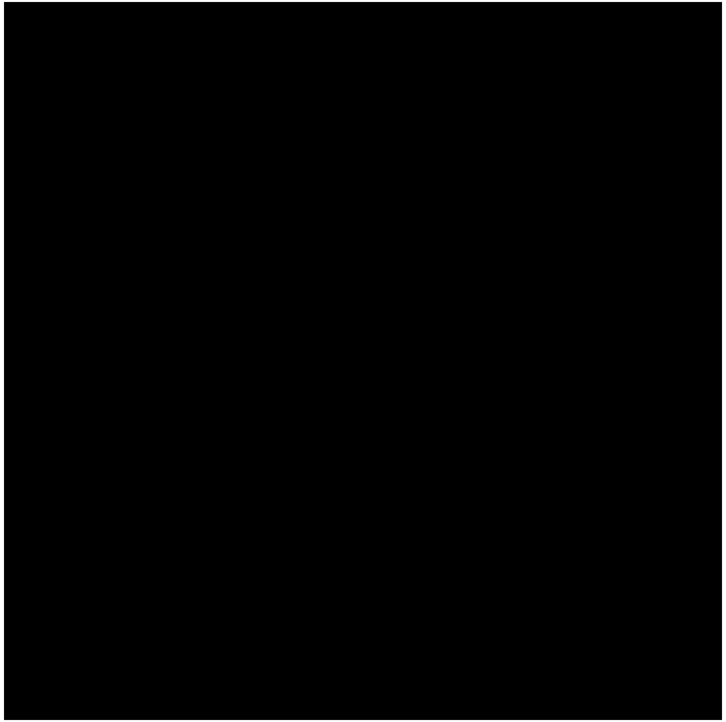


Figure 2: Obtained mask after processing

# Obtained Image Averages and Mask - Cam 3

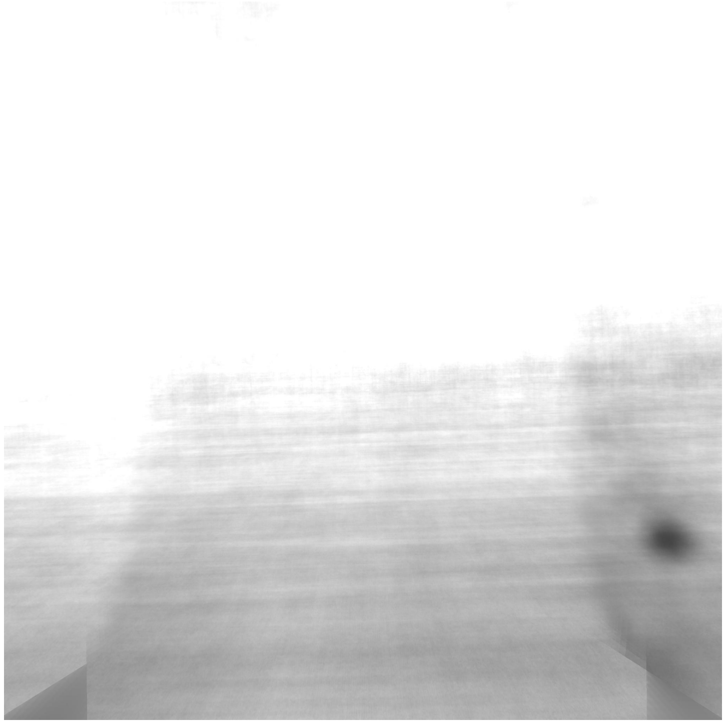


Figure 1: Average of all images

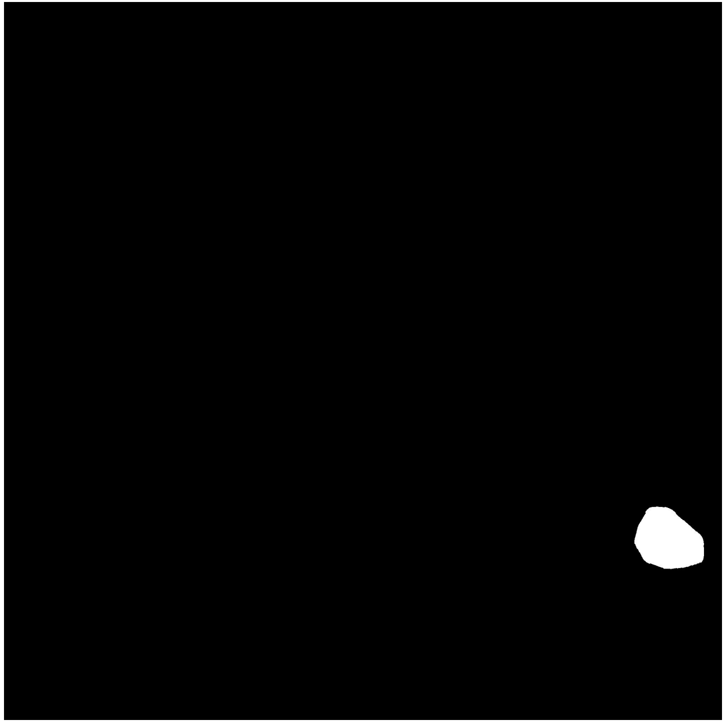


Figure 2: Obtained mask after processing

# Obtained Image Averages and Mask - Cam 5

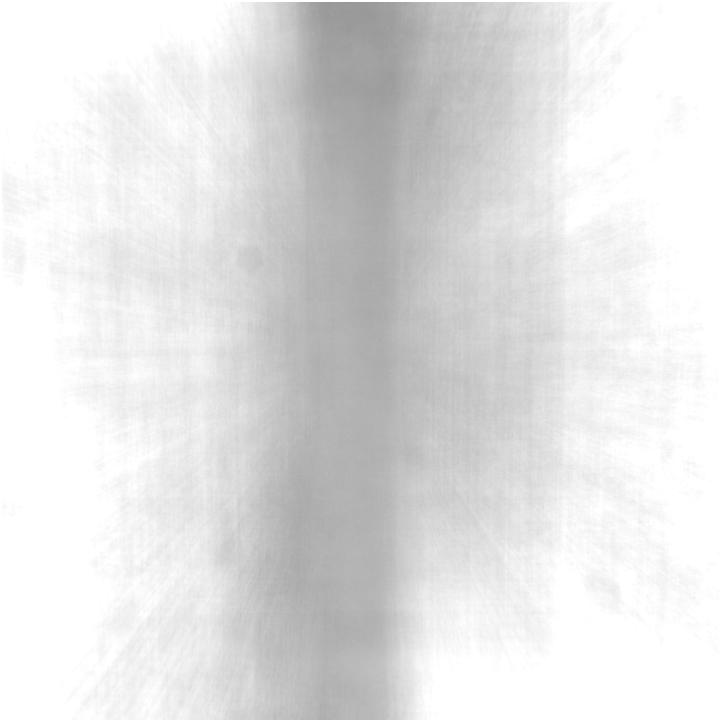


Figure 1: Average of all images

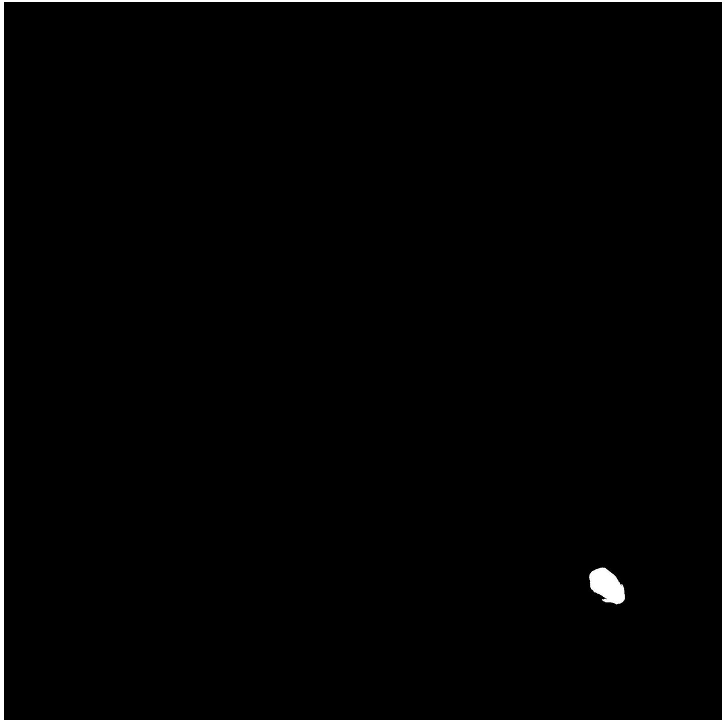


Figure 2: Obtained mask after processing

# Limitations & discussions

- A general approach did not provide a good output for all the cameras
- Custom filter parameters had to be given for each camera to filter out the smear more accurately.
- Smear visibility varies with different images. This makes the average less reliable.
- False positives obtained after processing made it unclear which patch was an actual smear. These were caused by the sky, or the road. The applied custom filters handled these cases.
- The possibility of using a convolutional neural network for camera smear detection was discussed as a possible improvement.