AUTOMATIC LENS SMEAR DETECTION

Presented by Nikit Bobba and Nisha Mallya

INTRODUCTION

- Our goal is to detect any smear present on a camera lens, given a sequence of street view images
- This aims to help overcome image quality issues due to any dirt or occlusions present on the lens
- Our project creates a binary mask indicating the size and position of any smear present on the lens

OPEN SOURCES/RESOURCES

We use Python 2.7 along with the following open sources/resources:

- Open CV (Open Source Computer Vision Library)
- Numpy
- Open CV documentation

METHODOLOGY

To detect the smear for a given camera, we perform the following steps:

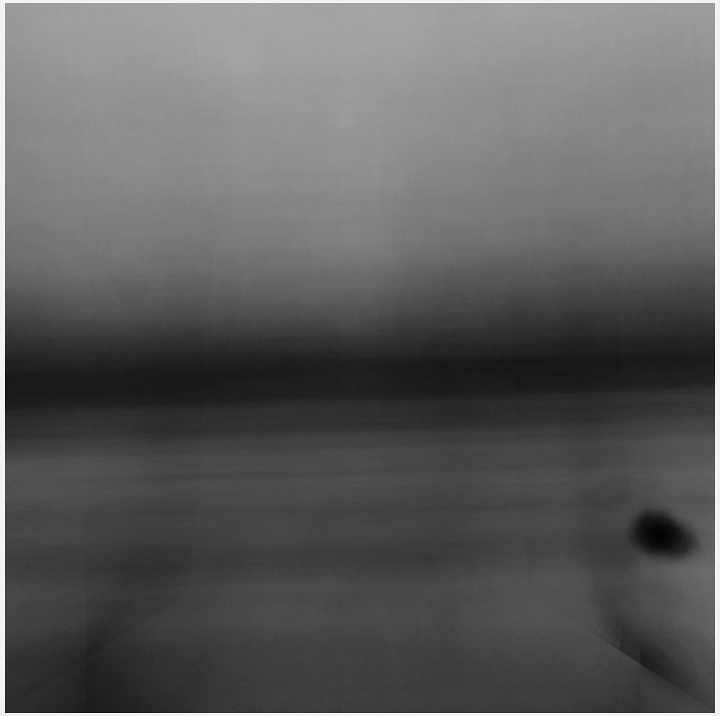
- We loop through the images from each camera to pre-process the images
- We pre-process the images by first converting each image to grayscale and then applying histogram equalization. This increases contrast between the lighter and darker areas thereby enhancing the visibility of any potential smear
- Next, we blur the image with a 3x3 kernel to remove any harsh lines in order to reduce noise, and apply an initial binary threshold to further increase smear visibility

METHODOLOGY

- Next, we split the total data from each camera into subsets of approximately 200 images (varies based on data size)
- We calculate an average of the images in order to create a single image from each camera
- Since the road landscape is consistently changing with the movement of the vehicle and the smear remains constant, calculating a mean image allows us to observe any potential smear, as the smear is darker than the rest of the image

MEAN IMAGE

camera 2 (no smear)



camera 3 (with smear)

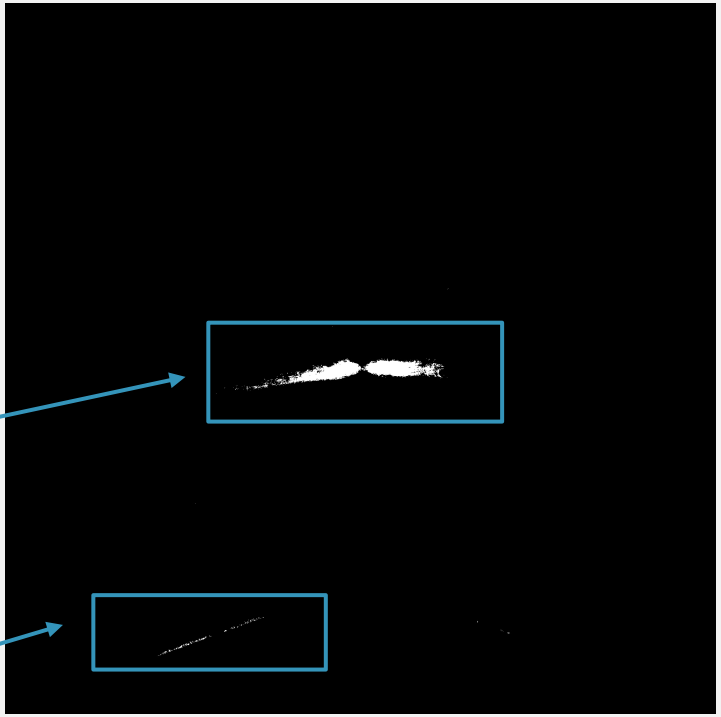
METHODOLOGY

- On the mean image, we apply a mean adaptive threshold which determines the threshold value based on the adjacent neighborhood of pixels
- After applying the adaptive threshold, we use a NOT bitwise operator to invert the image in order to create the intermediate binary mask for the smear
- At this stage, we have a mask for the smear with some noise present
- The two key sources of noise are due to the blurring panoramic effects of the camera at the bottom of the image, and the changing road patterns at the center of the image

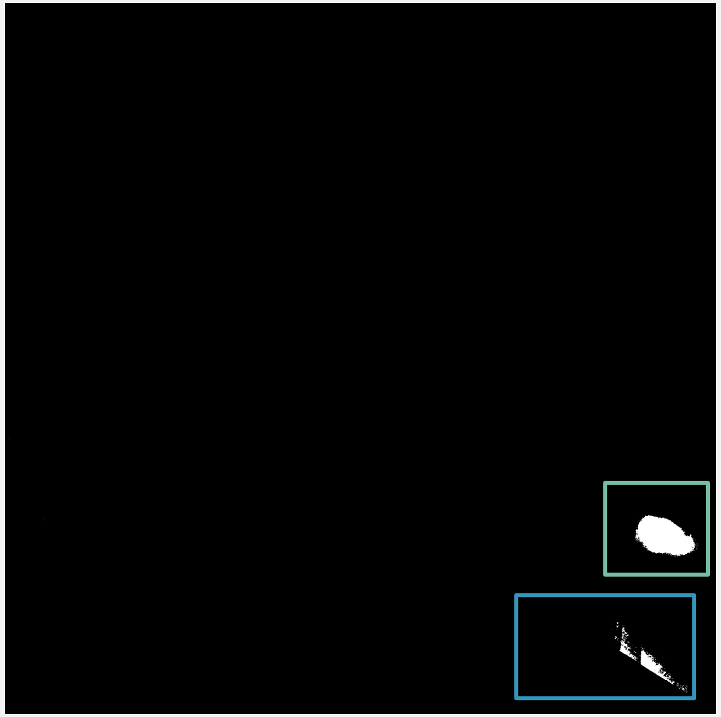
INTERMEDIATE MASK

noise due to road pattern

noise from panorama



camera 2 (no smear)



lens smear

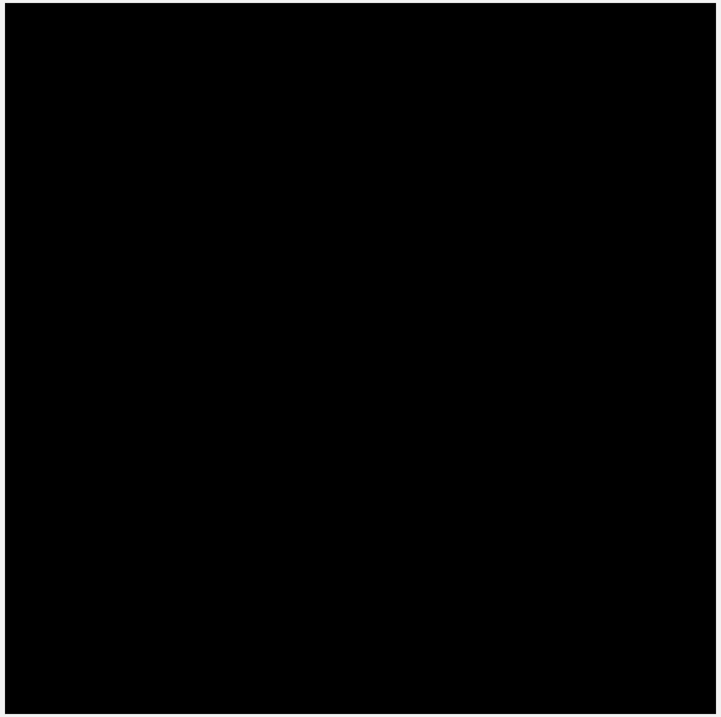
noise from panorama

camera 3 (with smear)

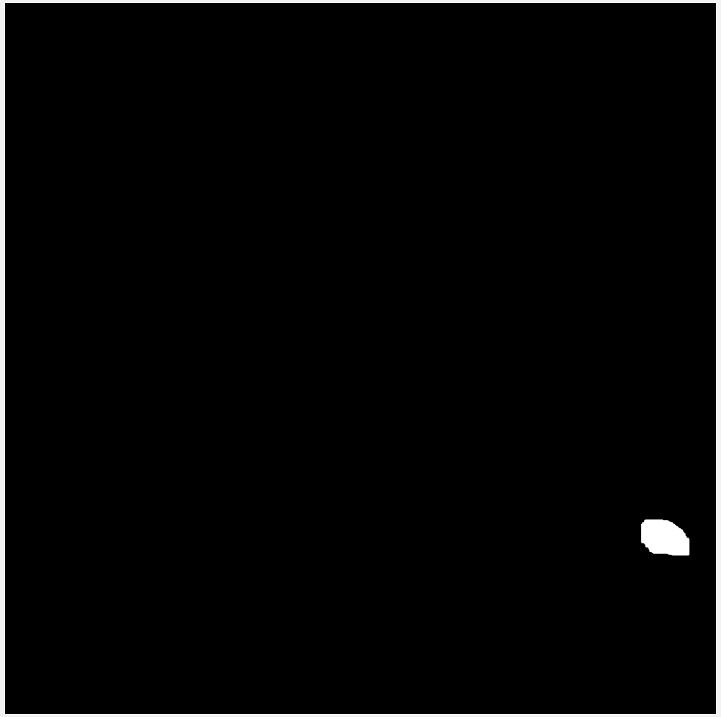
METHODOLOGY

- Next, we erode our image using a 10x10 kernel to remove the noise due to the changing road patterns and the panoramic camera effects
- Due to the erosion, any smear present also reduces in size. In order to counteract this effect, we dilate the image using a 10x10 kernel
- This produces a final binary mask for the camera lens
- If there is no smear, we are left with a binary mask that is completely black

FINAL MASK



camera 2 (no smear)



camera 3 (with smear)

DISCUSSION

- In the post-processing phase, we had to make tradeoffs between removing noise and preserving the full shape and size of the smear in the final mask
- We decided to perform five series of iterations of the erosion in order to fully remove any noise due to the panoramic effects of the camera and/or the road patterns
- Therefore, a potential weakness of our approach is that it may not be able to detect very small dirt or occlusions on the lens

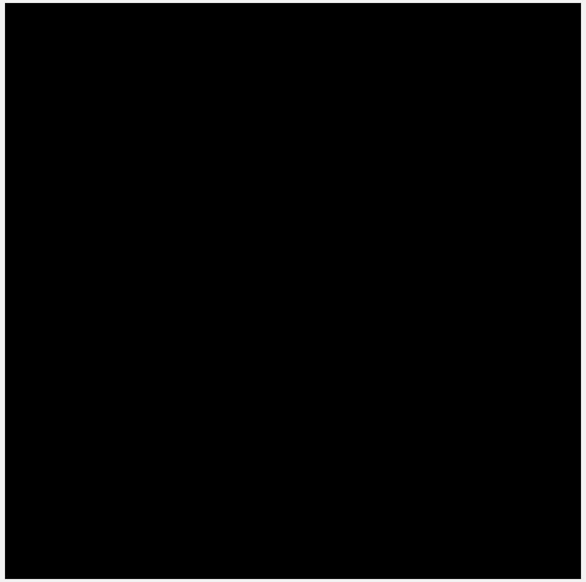
DISCUSSION

- We experimented with using both adaptive and binary thresholds
- We decided to use an adaptive threshold on the mean image, as it determines the threshold value based on the neighborhood of the pixel, which helps account for large constant light or dark areas of the images such as the sky, which helps reduce noise
- We also experimented with different kernel sizes for the erosion and dilation process and found that a 10x10 kernel allowed us to remove any noise while maintaining the structure of the smear within a reasonable number of iterations

RESULTS CAM_0

mean image

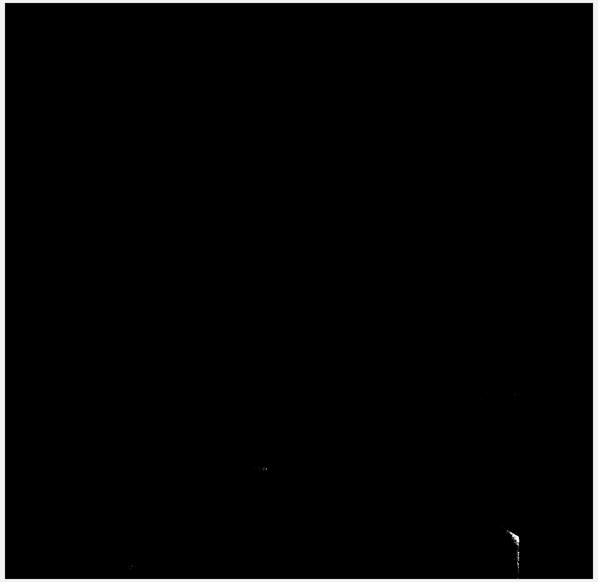
intermediate mask



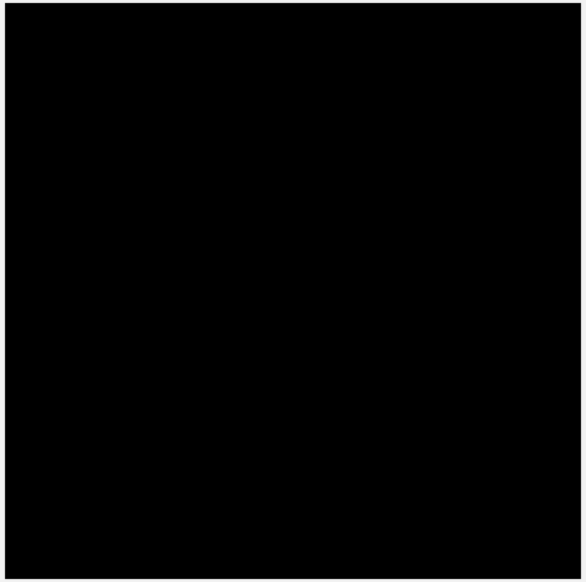
final mask

RESULTS CAM_1

mean image



intermediate mask

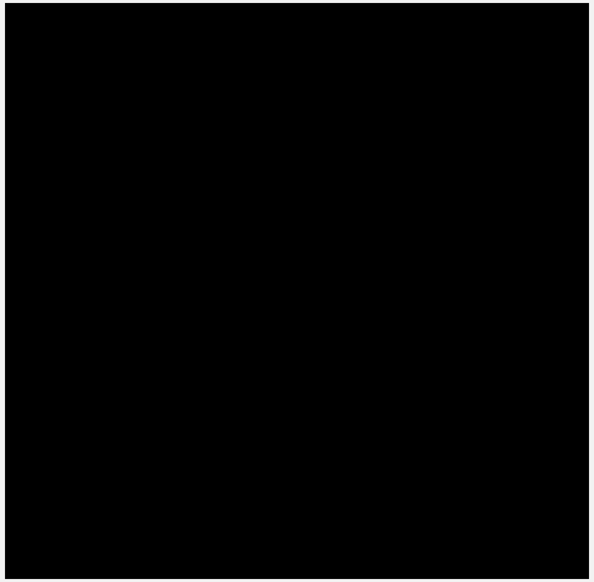


final mask

RESULTS CAM_2

mean image

intermediate mask

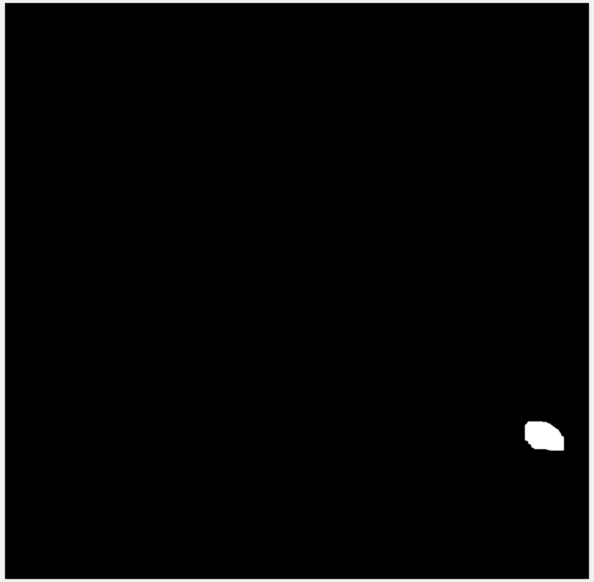


final mask

RESULTS CAM_3

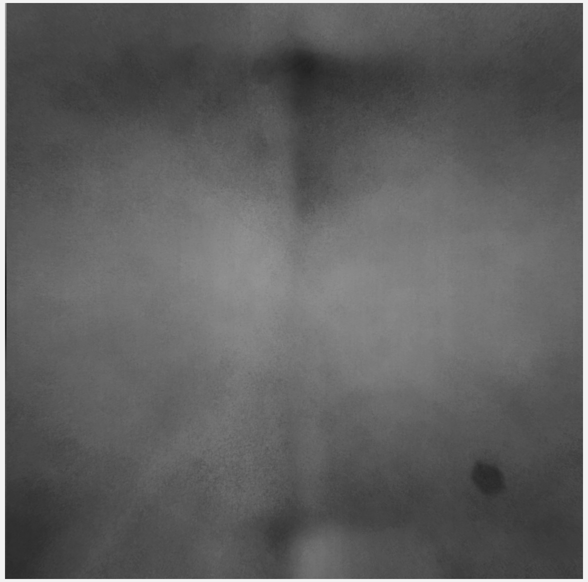
mean image

intermediate mask

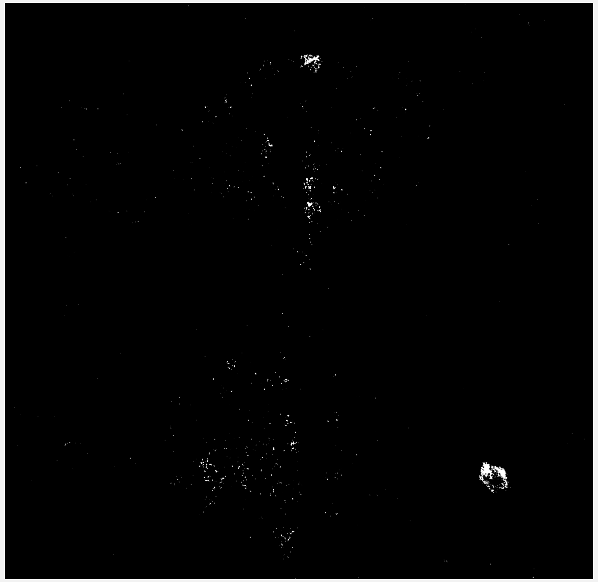


final mask

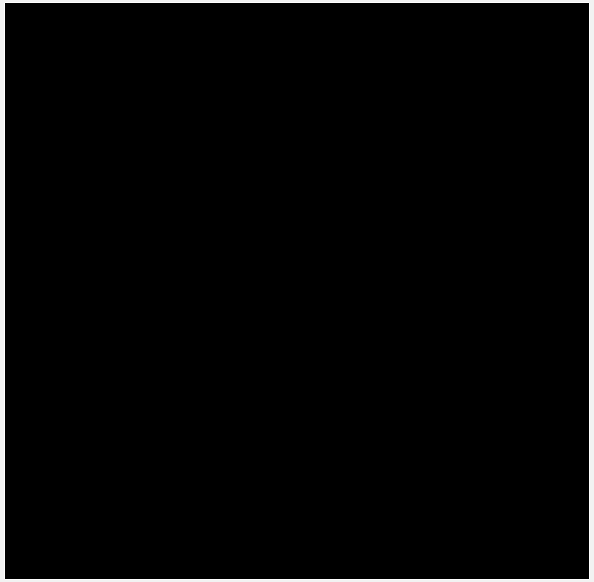
RESULTS CAM_5



mean image



intermediate mask



final mask

Thank you!