Code overview :

1. Preprocessing image change
    a. RGB to Grey format : to reduce the effect of change in light over the frames
    b. Gaussian blur: to remove any salt pepper noise and soften the edges
    c. Remove unwanted regions using black mask (eg: parking lot ceiling)

2. Compare frame change detection
    a. Calculate absolute difference between two frames
    b. Binary threshold to convert it into a mask
    c. Dilate operation smoothen the edges after the threshold operation
    d. All the external contours are found
    e. Size of the contours are compared with a min_area value
    f. If the sum of the area is bigger than a given score, the two frames are considered different

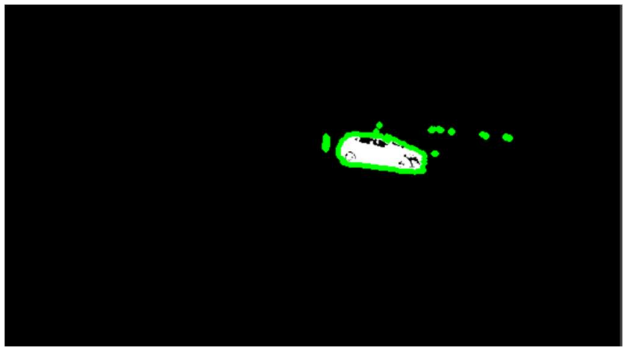
Example :

1. Original frames

2. After Preprocess

3. Absolute difference

4. Thresh -> Dilate -> contour



Questions :

1. Parameters used:
    a. Gaussian kernel size : [1,3,5]
        i. The size was decided on the basis of small edges in the background (floor tiles and poles)
    b. Black mask : [4,25,4,0]
        i. Black mask percentage was based on the garage ceiling as it is same across all the frames
    c. Threshold : 45
        i. Experimented with other values but higher values were not detecting missing black cars at the back
    d. Min contour area : 3000
        i. Lower values were detecting windows or glare as a change
    e. Score : 6000

Even though the given dataset is small , I have only used a subset of data . As in reality the datasets are quite big. To find the effectiveness of the above parameters , I used 20% of the given data to tune and validate the results.

2. Results
    a. Total no of images : 484
    b. Unique frames : 64
    c. Duplicates : 420

3. Data collection
    a. As this data will be further used for NN training, it's important to have unique images. The dataset should contains frames from different camera angle, time of day(light conditions), different seasons (snow and rain in background), different participants (Tri cycle, minibus, caravan, repair truck, car with trolley,etc)

4. Suggested pipeline :
    a. Bilateral blur could also be used instead of gaussian as it doesn't soften the object edges
    b. Loading and saving jpeg in cv2 is bit slow , instead TIFF could be used or any other uncompressed image format
    c. Adaptive threshold might perform well as compared to hard threshold as the actual dataset will have different lighting conditions
    d. instead of masking , image feature comparison could be used to detect different frames (SIFT, egde detection with cross correlation) but change of lighting conditions is also a good augmentation for the dataset.
    e. Training a Discriminator is also a viable option, but it will be bit slow and could be overkill for simple scenario distinctions.

Even though the above script works fine for most cases, there are still some drawbacks .I have kept the parameters for such that it selects most of the unique frames and along with it we have also some similar frames.

Ex :