1.1 In this module, we are performing frame difference by passing two sample frames and subtracting their corresponding pixel values. This can let us detect motion within frames and is potentially used in video surveillance cameras.

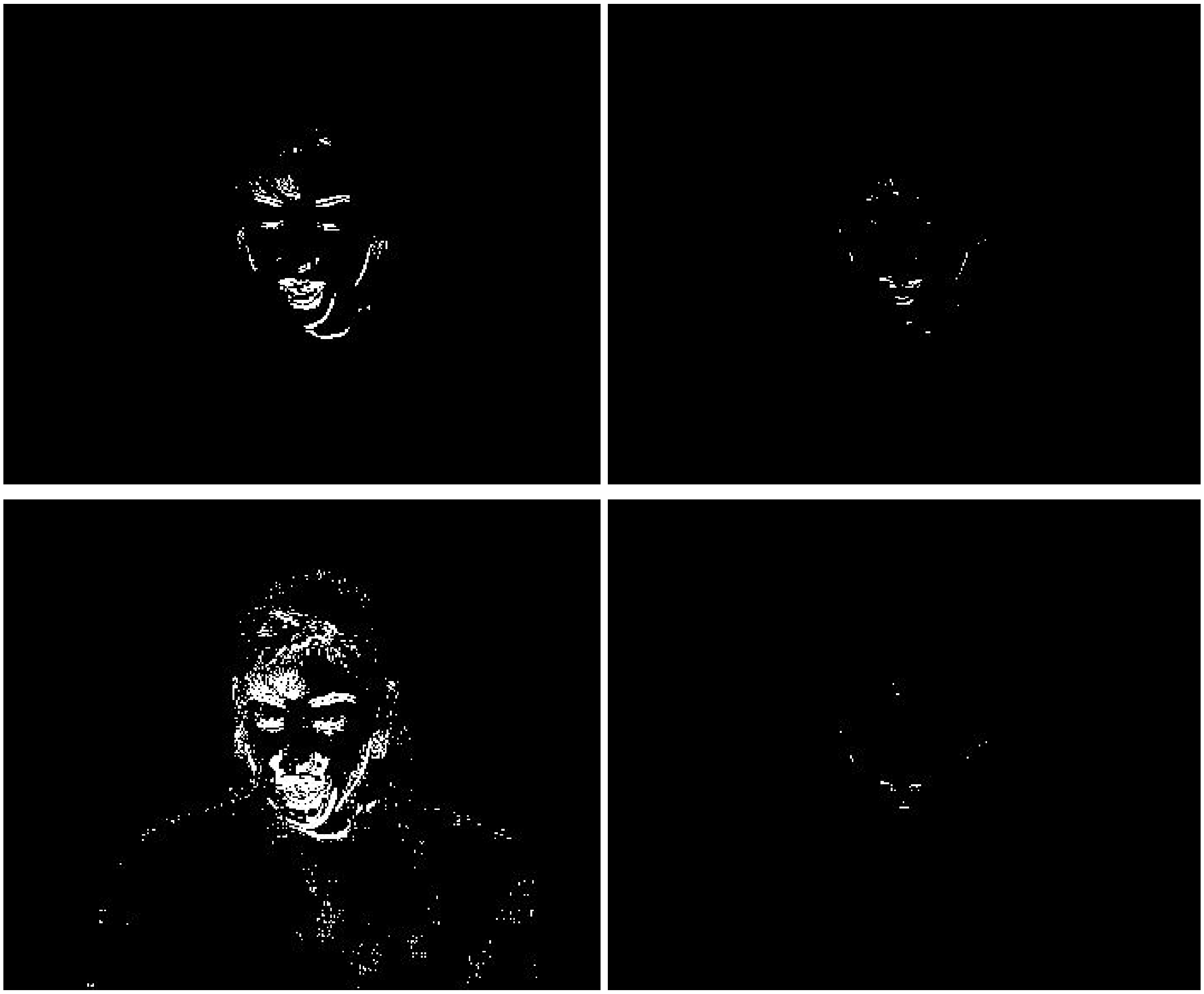
For this task, we are using two sample frames in the figures below.

These frames were captured from a famous video of a reporter where there is nearly no motion across the entire image except for the area around the chin and the mouth, where we can detect some temporal differences.

To apply background subtraction, we first make sure that the two images are grayscale then we perform frame difference. For the differences between the pixels, we perform absolute subtraction to make sure all values are within the grayscale color range (0:255). We then transform the result into a binary image where only the values of the pixels that represent motion are white.

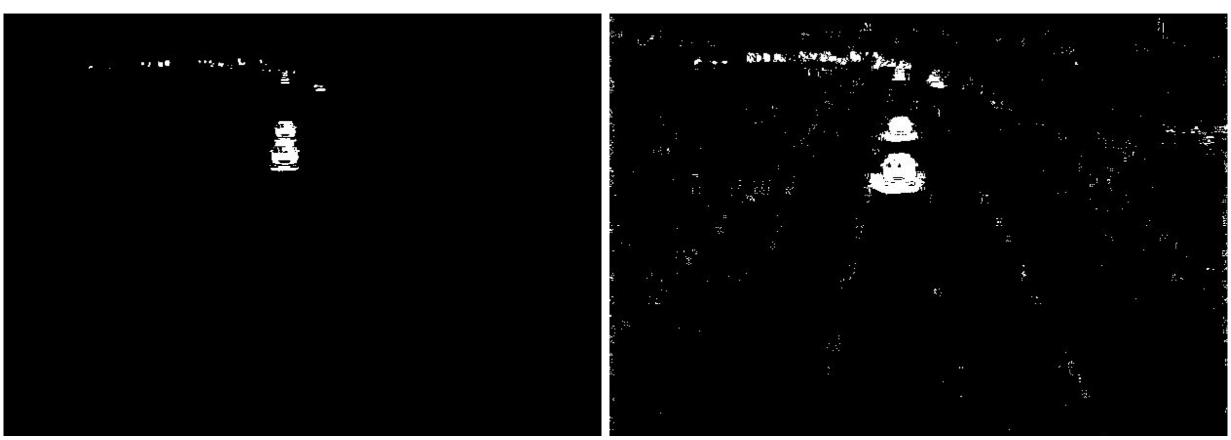
The result is represented in the next four figures. Each of them is produced when applying a different threshold. The first one is of threshold = 30, the second is of threshold = 60, the third is of threshold = 10, and the fourth is of threshold = 80.

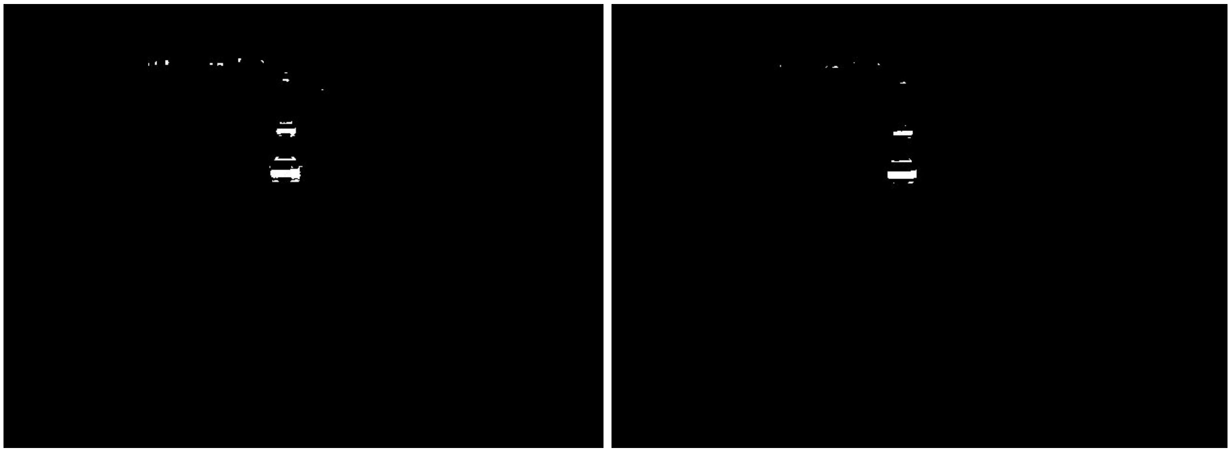
Clearly the ideal one is the first one (threshold = 30). This is because it represents the part where the motion is happening only without adding extra clutter. It is easier to understand as well.



1.2a In this module, we are also applying frame difference but for an entire dataset, where our reference frame is fixed. We use a sequence of images and compare each one of them with the original reference frame. This is useful in cases of indoors security cameras as the background of an indoor scene is usually static and there isn't much that can spoil the integrity of the original reference frame/scene. This is a sample image sequence:

Suppose we take the first (upper left) one as the reference frame for the lower two, then we get:





This is for thresholds 45, 10, 90, 120 respectively. The first one (threshold = 45) is the ideal one here because the second contains extra noise and the last two does not emphasize the details clearly enough. The first one represents the movement of the front car smoothly.

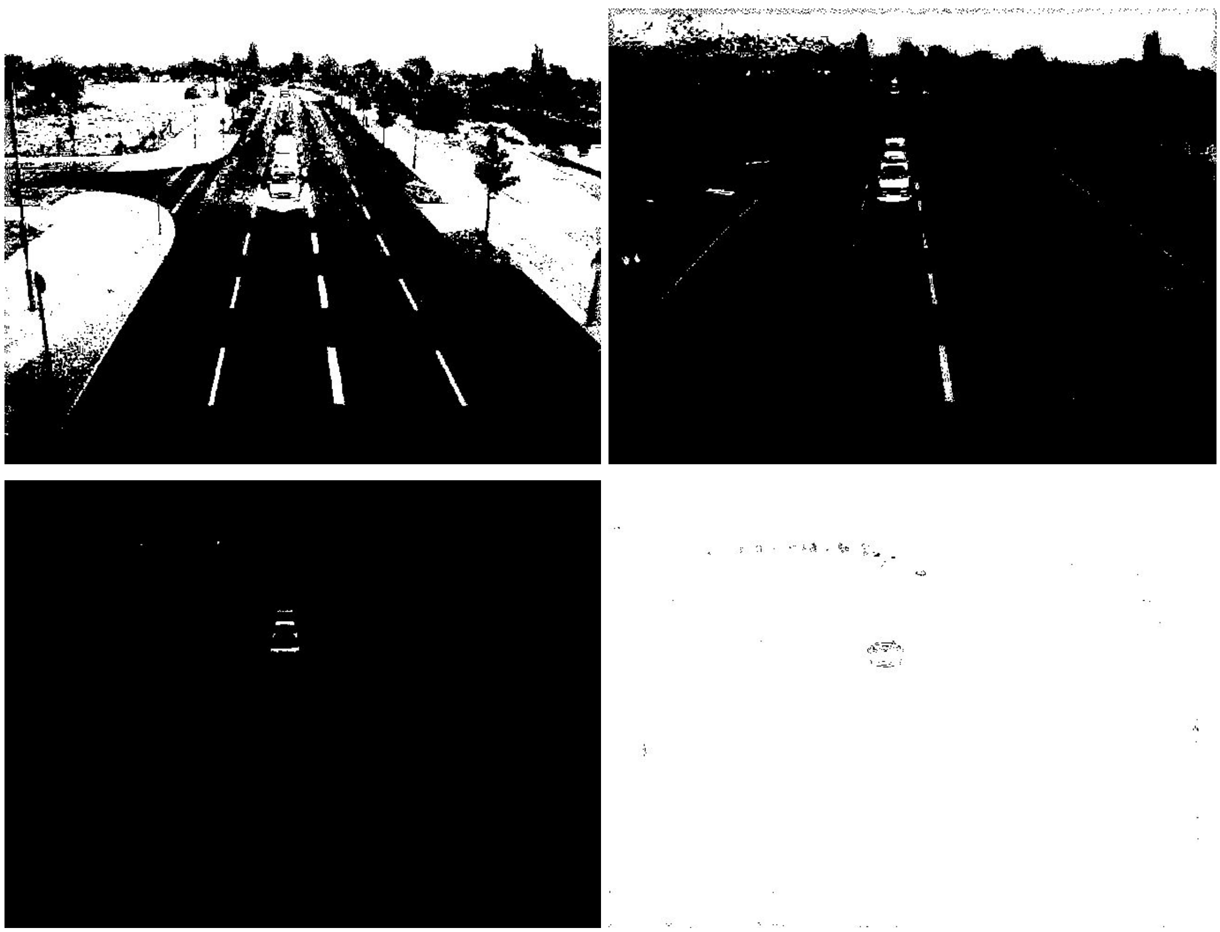
Testing this module on the second dataset for the input images:

We get the following results for applying frame difference between the first frame and the last frame, where we apply thresholds 150, 100, 200 and 220 respectively:

Clearly 150 is the ideal threshold to apply because it shows the moving car towards the camera without cluttering the external environment with white pixels nor rendering the motion in the scene ambiguous as in the last two frames.

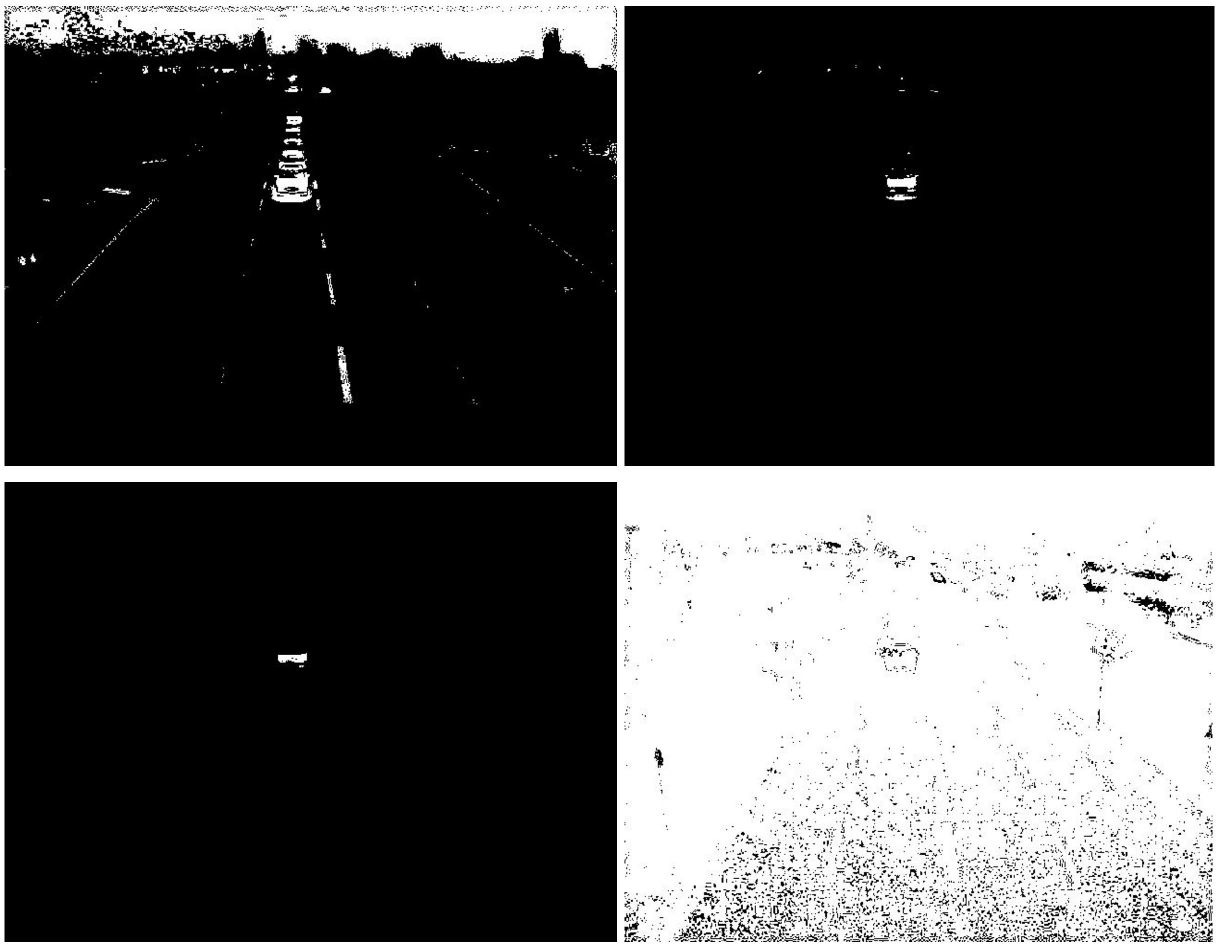
1.2b In this module, we perform dynamic frame difference, where we take the average of the last n frames as a reference frame for the successor frames. This is useful in cases of security cameras of shops and banks in semi-crowded streets and squares. This is because the background of these sceneries are always changing, so applying a fixed reference frame for background modeling doesn't count like a feasible option.

For the same four frames of the first dataset represented above, taking the average of the first three and comparing it with the fourth one with thresholds 40, 80, 120 and 10 respectively:



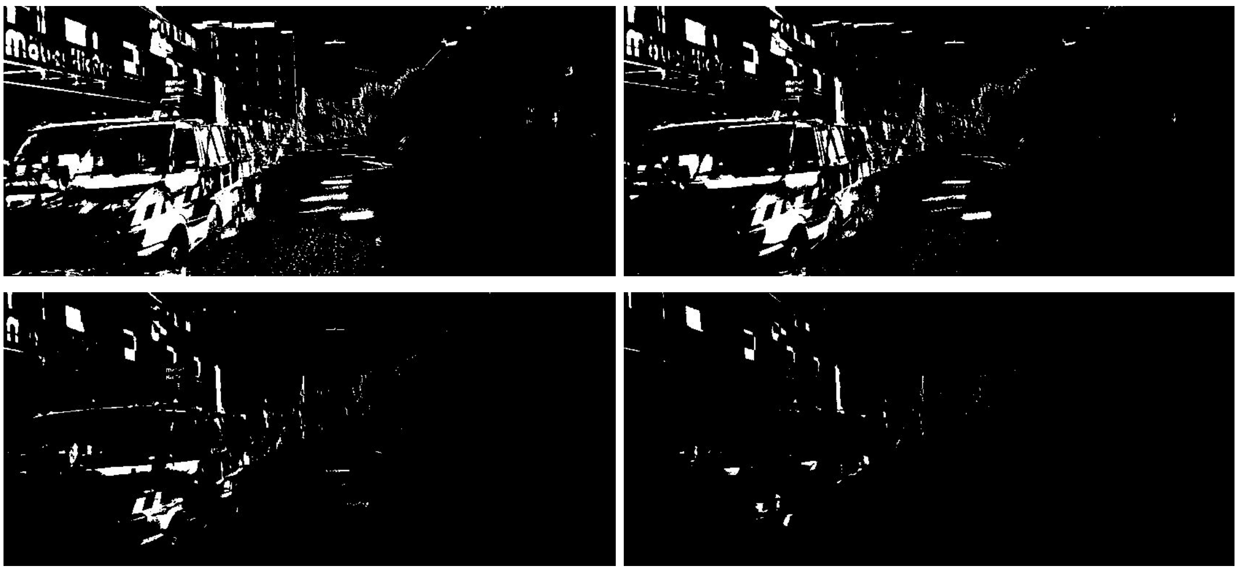
From this, we get that 40 is the ideal threshold because we can clearly see every aspect of the environment.

Changing the number of reference frames (taking older frames) to 6 (instead of 3) and applying frame difference with the same exact frame with the same thresholds results in:



In this case, also, 40 is the ideal threshold.

Applying the same module on the second dataset, taking the average of the first three frames represented above and comparing them with the fourth with thresholds 100, 130, 180 and 220 respectively results in:



In this case, 130 is the threshold that emphasizes the difference best.

Changing the number of reference frames from 3 to 2 with applying the same thresholds gives us:

After changing n, the ideal threshold is still 130 because 110 results in a grainy frame and more than 130 results in an overall black frame that doesn’t clarify differences.

I think the dynamic background modeling technique is the best one because the static alternative tends to be more error-prone and can sometimes provide false information about the scene such as the cases in which a static object enters the scene and remains there. In this case the static background modeling can detect this stationary object as a moving object which is not the case. So taking the average of the last n reference frames is always the best option.