ASSIGNMENT - 1

PSEUDOCODE:-

1) Initialize parameters:
   a) Mixture of gaussians (MOG) parameters for each pixel: mean, variance, weights.
   b) Pre-compute standard deviation and 2.5 times standard deviation.

2) Start a loop to process each frame from the video:
   a) Read the current frame.
   b) Convert the frame to grayscale.

3) Compare each pixel of the current frame with the MOG:
   a) Calculate the absolute difference between the current frame and the mean of each Gaussian component.
   b) Compute weights divided by standard deviation for each Gaussian component.
   c) Determine matched and not-matched pixels based on the difference and threshold (2.5 * stddev).

4) Update Gaussian models for matched pixels:
   a) Calculate the probability of the current pixel belonging to the background using a normal probability density function.
   b) Update mean, variance, and weights of the Gaussian components for matched pixels.

5) Update parameters for unmatched pixels:
   a) Update mean, variance, and weights for unmatched pixels. b) Normalize weights to ensure they sum up to 1.

6) Sort Gaussian models based on weights divided by standard deviation.

7) Update background and foreground indices:
   a) Identify foreground pixels based on the weights of Gaussian components.
   b) Separate matched pixels into two categories:
   indices where T< of weight of most probable gauss model and indices where T< of sum of weight of (middle + most) probable gauss model
   c) comparison with standard deviation.

8) Update the background image:
   a) Update pixels in the background image using the current frame and foreground indices.

9) Release the video capture object and close all windows.

PARAMETERS:-

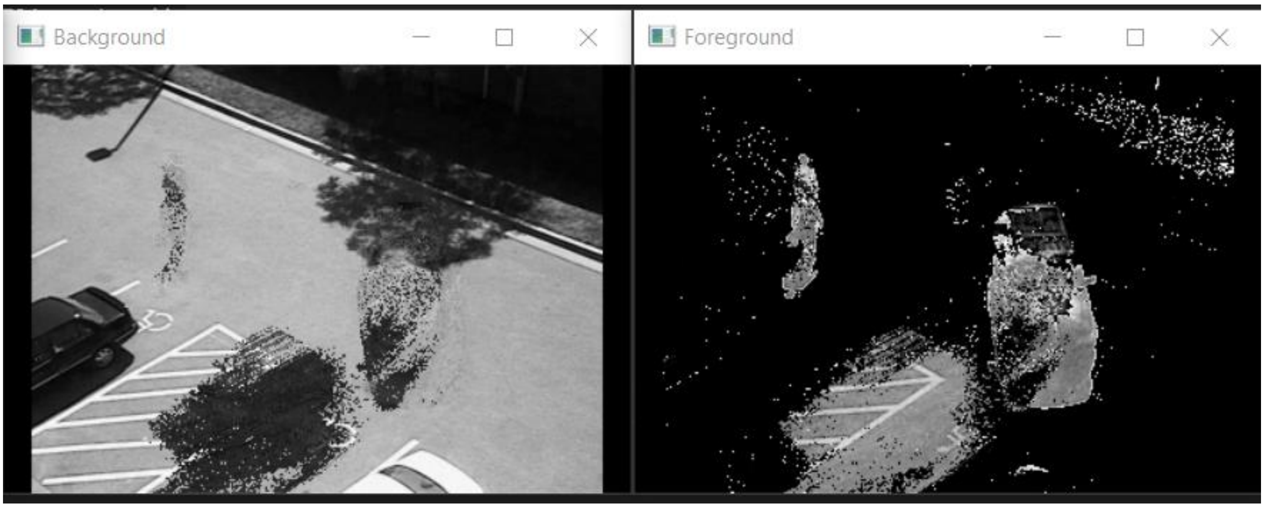
1) ALPHA:- Learning rate

2) THRESHOLD:- % of data accounted for background

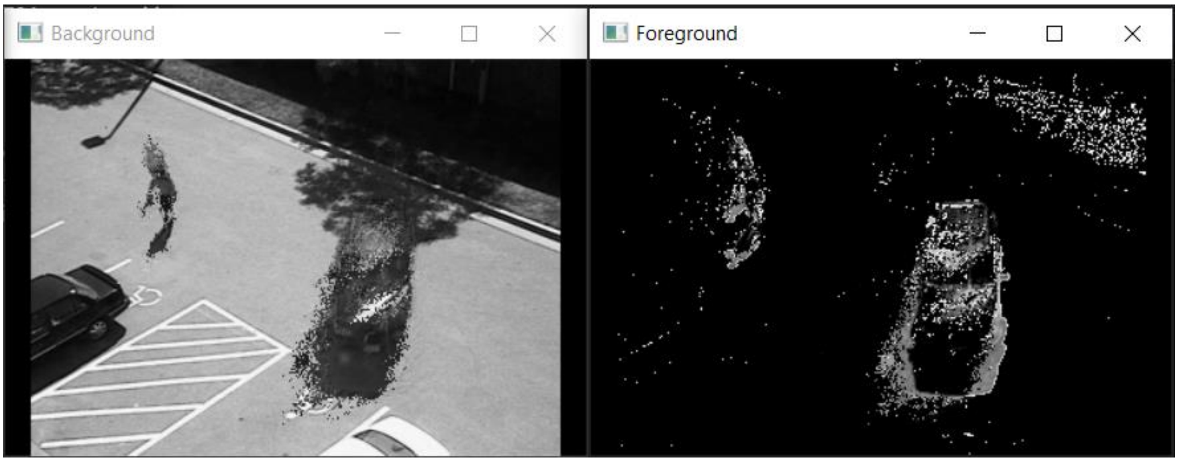
EXPERIMENTATION:-

Experiment	ALPHA	T
1.	0.01	0.85
2.	0.1	0.85
3.	0.01	0.30

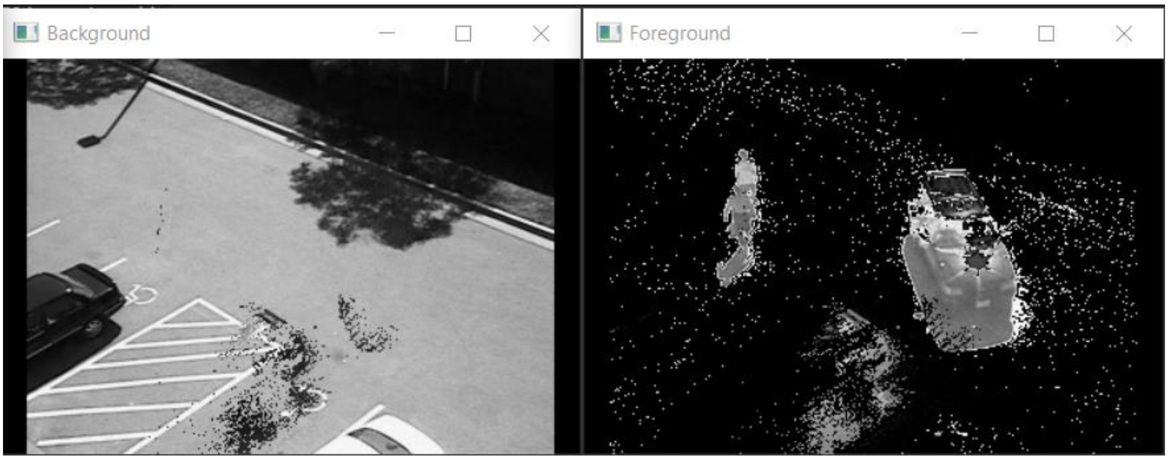
4.	0.1	0.30



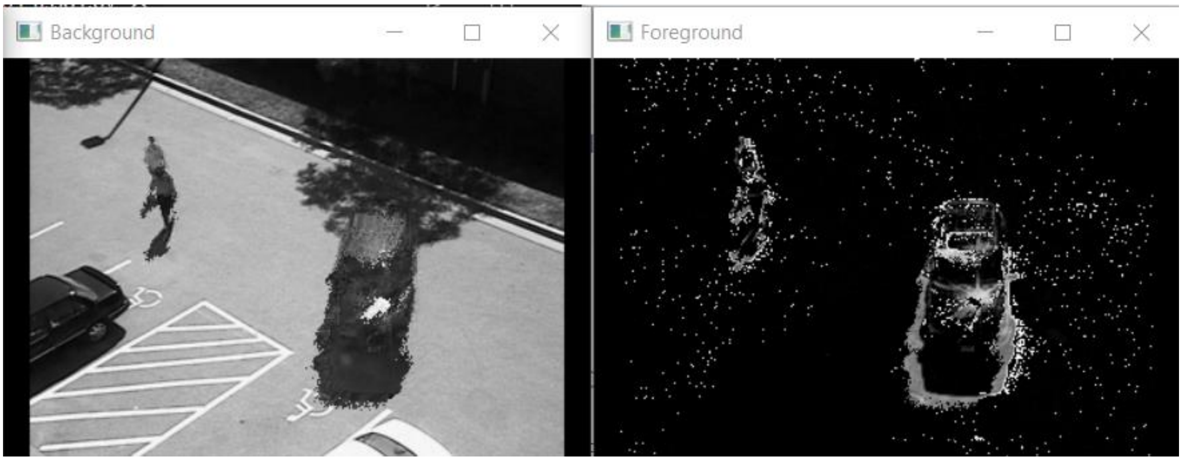
Exp1



Exp2



Exp3



Exp4

OBSERVATIONS:-

1) Setting alpha = 0.01 and threshold = 0.85 in the background segmentation algorithm yields effective foreground segmentation with minimal false positives. A slow learning rate (alpha) ensures stable background adaptation, reducing noise sensitivity and providing smoother transitions between background and foreground.

2) As the value of the alpha is more here so it is more adaptive than the previous observation Since the threshold is high, it will require a higher weight for a Gaussian to be considered as part of the foreground. Consequently, fewer Gaussians will meet this criterion, leading to fewer pixels being classified as foreground.

3) The slow learning rate is ensuring a stable background adaption, but the lesser threshold value means that less weight is required for a Gaussian to be considered part of the foreground. Consequently, more Gaussians will meet this criterion, leading to more pixels being classified as foreground.

4) It is more adaptive than the previous observation but because of

lesser threshold value it will consider more gaussians to meet the criteria and it will leads to classify more pixels as the part of the foreground.

INFERENCES:-

1) **Effect of Alpha:**

Lower values of alpha (e.g., 0.01) result in slower adaptation to changes in the scene, leading to a more stable background model over time. Conversely, higher values of alpha (e.g., 0.1) allow for quicker updates to the background model, making it more responsive to changes but potentially introducing more noise.

2) **Effect of Threshold:**

A lower threshold (e.g., 0.3) increases sensitivity to foreground objects, enabling better detection of moving objects but may also lead to more false results due to noise. Conversely, a higher threshold (e.g., 0.85) reduces sensitivity, resulting in a more conservative approach to foreground detection.