Experiment No: 5

Aim: Implementation of the different background subtraction techniques for foreground detection in video sequences.

Software Used: MATLAB R2024b.

Theory:

Background Subtraction Techniques: Background subtraction is a widely used technique in computer vision for detecting moving objects in videos. The core idea is to create a model of the background and then subtract it from the current frame to isolate the foreground (moving objects). The background model can be updated using recursive or non-recursive approaches.

Non-Recursive Approaches:
Frame Differencing: This method subtracts the current frame from the previous frame to detect changes. It is simple but sensitive to noise and illumination changes.

$$D_t(x,y) = F_t(x,y) - F_{t-1}(x,y)$$

Mean Filtering: The background is modelled as the mean of the previous N*N* frames. This method is less sensitive to noise but requires more memory.

$$B_t(x,y) = \frac{1}{N}\sum_{i=1}^{N} F_{t-i}(x,y)$$

Median Filtering: The background is modelled as the median of the previous N*N* frames. It is robust to outliers and noise.

$$B_t(x,y) = \text{median}(F_{t-1}(x,y), F_{t-2}(x,y), ..., F_{t-N}(x,y))$$

Weighted Average: The background is modelled as a weighted average of the previous frames, giving more importance to recent frames.

$$B_t(x,y) = \alpha_1 F_{t-1}(x,y) + \alpha_2 F_{t-2}(x,y) + \alpha_3 F_{t-3}(x,y)$$

Once the background is estimated, the foreground (moving objects) is detected by thresholding the difference:

$$M_t(x,y) = \begin{cases} 255, & |F_t(x,y) - B_t(x,y)| > T \\ 0, & \text{otherwise} \end{cases}$$

where is T the threshold.

Recursive Approaches: In recursive approaches, the background model is updated recursively using a learning rate (alpha). The background model at time *t* is a weighted combination of the current frame and the background model at time t−1.

$$B_t(x,y) = \alpha F_{t-1}(x,y) + (1-\alpha) B_{t-1}(x,y)$$

1. Alpha (learning rate) controls the contribution of the new frame.
2. Higher adapts quickly but is sensitive to noise, while lower adapts slowly and retains long-term background information.

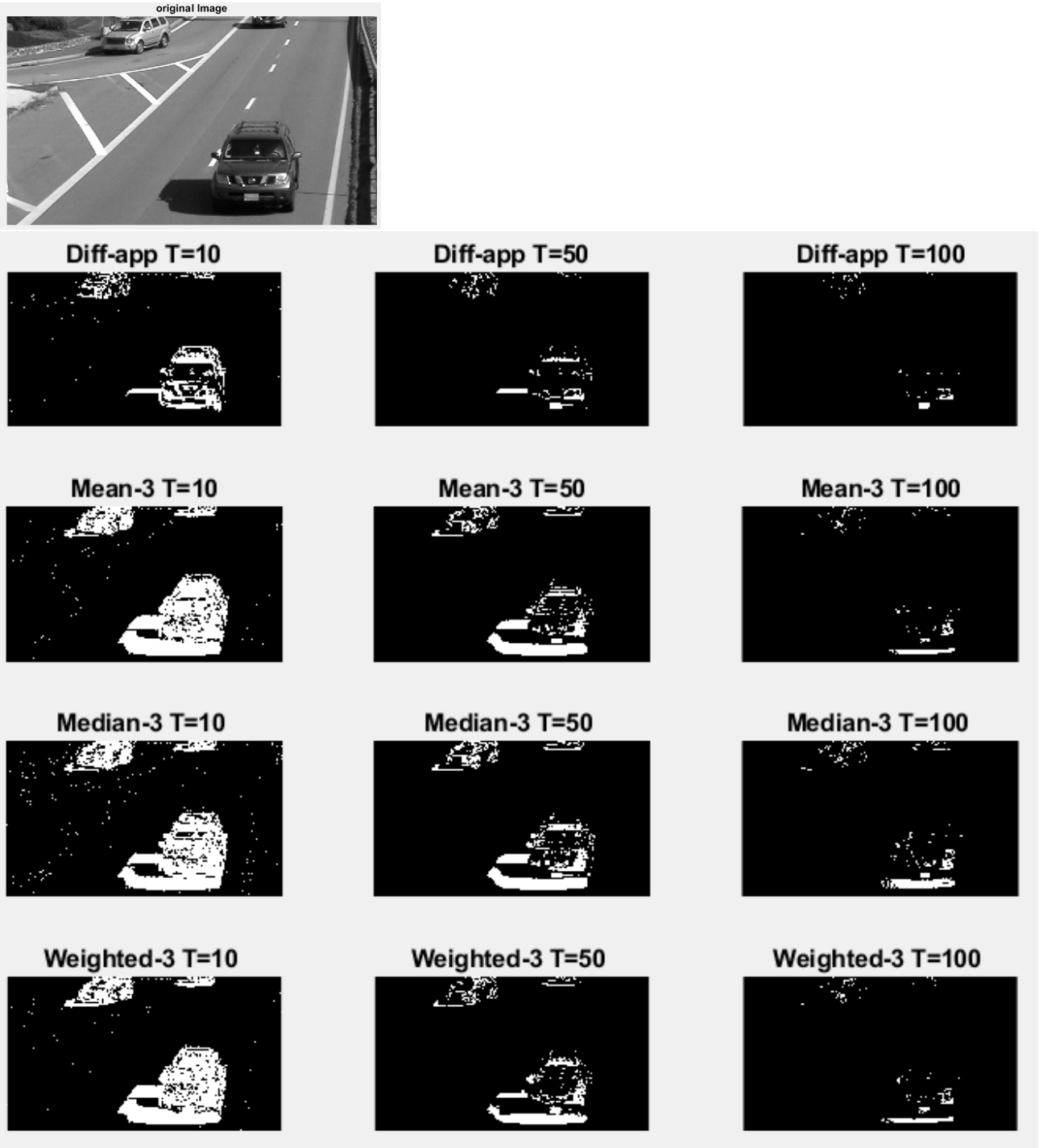
The foreground mask is computed as: $M_t(x,y) = |F_t(x,y) - B_{t-1}(x,y)|$
where B_t is the background model at time t, is the current frame, and α is the learning rate.

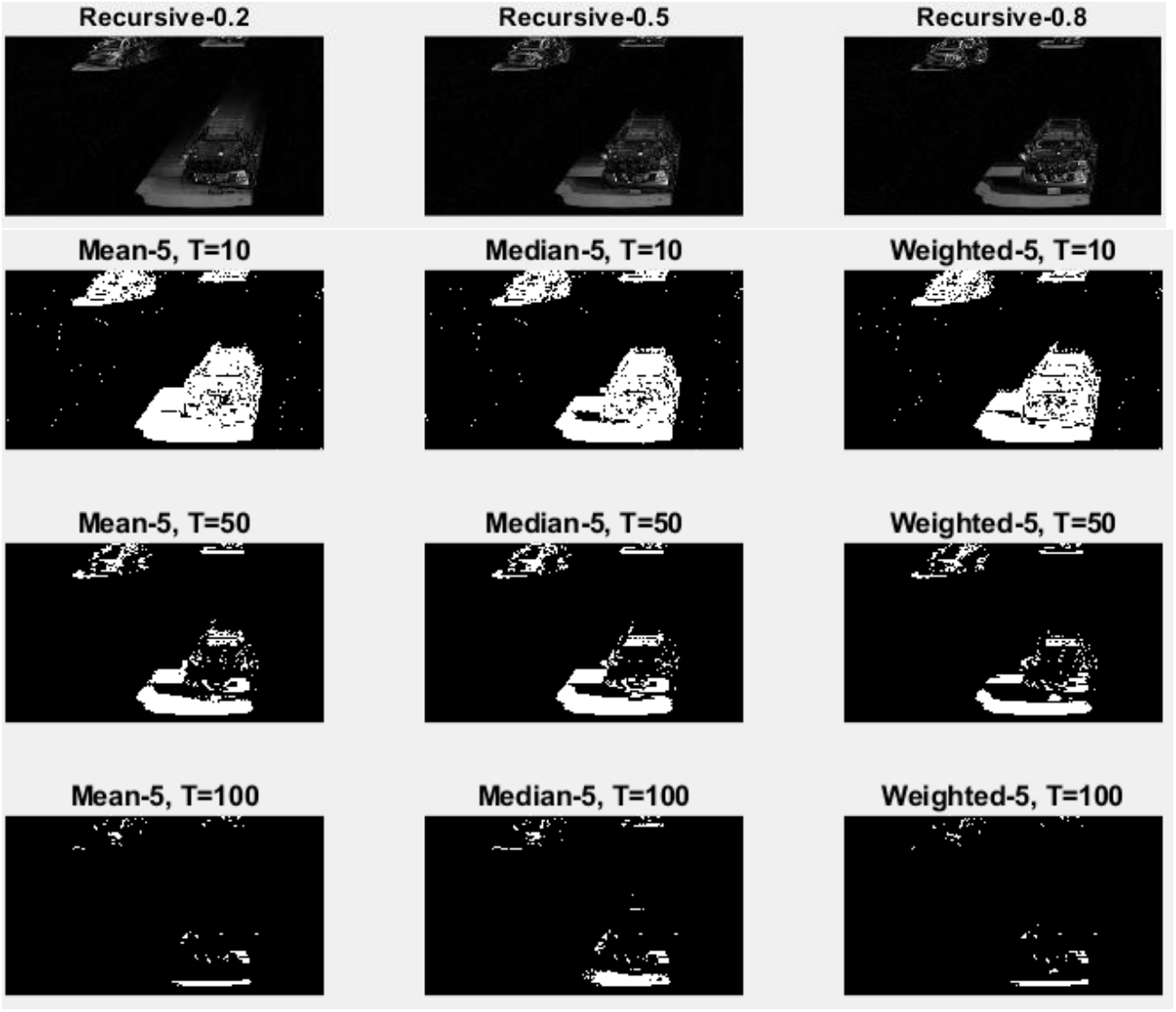
Thresholding: Thresholding is used to classify pixels as foreground or background. If the difference between the current frame and the background model exceeds a threshold T*T*, the pixel is classified as foreground. The choice of threshold affects the sensitivity of the detection:

- **Low Threshold**: More sensitive to small changes but may include noise.
- **High Threshold**: Less sensitive to noise but may miss subtle changes.

Kernel Size: In median filtering, the kernel size determines the number of frames used to compute the background. A larger kernel size makes the model more robust to noise but may also smooth out important details.

Result:





Discussion:

Frame Differencing:

Low Threshold (T=10):

- Detects small changes, including noise.
- Produces a cluttered foreground mask with many false positives.

Medium Threshold (T=50):

- Balances noise reduction and foreground detection.
- Detects most moving objects while reducing false positives.

High Threshold (T=100):

- Miss subtle changes but significantly reduces noise.
- Suitable for environments with large, distinct foreground objects.

Mean Filter (Temporal Averaging): This method provided a stable background estimation by averaging past frames, effectively reducing noise.it struggled in dynamic scenes where the background changed over time, as the averaging process retained outdated background information.
Low Threshold (T=10): Detects small changes but includes noise.

Medium Threshold (T=50): Provides a good balance between noise reduction and foreground detection.

High Threshold (T=100): Miss subtle changes but reduces noise effectively.

Effect of Kernel Size:

- 3-Frame Kernel: Less robust to noise but computationally efficient.
- 5-Frame Kernel: More robust to noise but requires more memory and computation.

Median Filter: The median filter was more robust against sudden changes, such as objects appearing/disappearing, compared to the mean filter.

- It effectively handled transient noise (e.g., flickering lights or occasional occlusions).

Low Threshold (T=10): Detects small changes but includes some noise.

Medium Threshold (T=50): Provides a good balance between noise reduction and foreground detection.

High Threshold (T=100): Miss subtle changes but significantly reduces noise.

Effect of Kernel Size:

- 3-Frame Kernel: Less robust to noise but computationally efficient.
- 5-Frame Kernel: More robust to noise and outliers but requires more computation.

Weighted Average Filter: By assigning different weights to past frames, this method provided a balance between adaptability and stability.

It performed better than the mean filter in dynamic environments but required fine-tuning of weights for optimal performance.

Low Threshold (T=10): Detects small changes but includes noise.

Medium Threshold (T=50): Provides a good balance between noise reduction and foreground detection.

High Threshold (T=100): Miss subtle changes but reduces noise effectively.

Effect of Kernel Size:

- 3-Frame Kernel: Less robust to noise but computationally efficient.
- 5-Frame Kernel: More robust to noise and adapts better to recent changes.

Recursive Background Approach:

- This method adapted quickly to background changes, making it effective for real-time applications.
- High α value allowed fast adaptation but made the background model susceptible to noise and short-term occlusions.
- Low α resulted in a more stable background but made the model slow to adapt to changes, such as sudden illumination variations.

Foreground Detection and Sensitivity to Threshold (T)

- Choosing an appropriate threshold T was critical for differentiating motion from noise.
- A low T detected minor movements but also captured noise, while a high T eliminated noise but missed subtle motion.

For non-recursive methods:

- A small kernel size made the background update faster but retained foreground objects longer.
- A large kernel size improved stability but failed to react to changes promptly.

For recursive methods:

- A high α made the background model dynamic but sensitive to noise.
- A low α kept the model stable but slow to react to background changes.

Conclusion: This experiment demonstrated the effectiveness of both non-recursive and recursive background modeling techniques for motion detection. Non-recursive methods, such as diff-frame, mean, median, and weighted averaging, provided robust background estimation but required higher computational resources. Recursive methods, particularly the exponential moving average, efficiently updated the background in real time but were sensitive to noise depending on the learning rate (α). The choice of method depends on the application, with non-recursive methods being more stable for static environments and recursive methods being ideal for dynamic scenes. Proper selection of parameters such as window size (N) and α is crucial for achieving optimal performance in background modeling.

Code:

```
videoFile = 'visiontraffic.avi';     vidObj = VideoReader(videoFile);
frameRate = vidObj.FrameRate;  numFrames = vidObj.NumFrames;
rows = vidObj.Height; cols = vidObj.Width;
originalFrames =  zeros(rows, cols, numFrames, 'uint8');
for i = 1:numFrames
   originalFrames(:, :, i) = rgb2gray(read(vidObj, i));
end
mean_Frames = originalFrames; median_Frames = originalFrames;
weighted_Frames = originalFrames; diffFrames = originalFrames;
for_mean_Frames = originalFrames; for_median_Frames = originalFrames;
for_weighted_Frames = originalFrames; for_diff_Frames = originalFrames;
Threshold = [10, 50, 100];
figure;
for t = 1:length(Threshold)
   T = Threshold(t);
   diffFrames(:, :, 1) = originalFrames(:, :, 1);
   diffFrames(:, :, 2) = originalFrames(:, :, 2)-originalFrames(:, :, 1);
   diffFrames(:, :, 3) = originalFrames(:, :, 3)-originalFrames(:, :, 2);
   for i = 4:numFrames
     mean_Frames(:, :, i) = uint8(0.33 * originalFrames(:, :, i-1) + 0.33 * originalFrames(:, :, i-2)+ 0.33*
originalFrames(:,:,i-3));
     median_Frames(:, :, i) = median(cat(3, originalFrames(:, :, max(1, i-2)), originalFrames(:, :, i-2),
originalFrames(:, :, i-3)), 3);
     weighted_Frames(:, :, i) = uint8(0.5* originalFrames(:, :, i-1) + 0.3*originalFrames(:, :, i-
2)+0.2*originalFrames(:, :, i-3));
     diffFrames(:, :, i) = uint8(abs(originalFrames(:, :, i) - originalFrames(:, :, i-1)));
   end
 for f = 2:numFrames
     for_mean_Frames(:, :, f) = uint8((abs(double(originalFrames(:, :, f)) - double(mean_Frames(:, :, f))) > T) *
255);
     for_median_Frames(:, :, f) = uint8((abs(double(originalFrames(:, :, f)) - double(median_Frames(:, :, f))) > T)
* 255);
     for_weighted_Frames(:, :, f) = uint8((abs(double(originalFrames(:, :, f)) - double(weighted_Frames(:, :, f)))
> T) * 255);
     for_diff_Frames(:, :, f) = uint8((diffFrames(:, :, f) > T) * 255);
 end
 subplot(4, 3, t), imshow(for_diff_Frames(:,:,165)), title(sprintf("Diff-app T=%d", T));
subplot(4, 3, 3+t), imshow(for_mean_Frames(:,:,165)), title(sprintf("Mean-3 T=%d", T));
subplot(4, 3, 6+t), imshow(for_median_Frames(:,:,165)), title(sprintf("Median-3 T=%d ", T));
```

```
  subplot(4, 3, 9+t), imshow(for_weighted_Frames(:,:,165)), title(sprintf("Weighted-3 T=%d ", T));
end
figure();
for t = 1:3
    T = Threshold{t};
    for i = 6:numFrames
        mean_Frames(:, :, i) = uint8(0.2 * originalFrames(:, :, i-1) +0.2 * originalFrames(:, :, i-2) + 0.2 *
     originalFrames(:, :, i-3) + 0.2 * originalFrames(:, :, i-4) + 0.2 * originalFrames(:, :, i-5));
        median_Frames(:, :, i) = median(cat(3, originalFrames(:, :, i-1),originalFrames(:, :, i-2),originalFrames(:, :, i-3),
     originalFrames(:, :, i-4), originalFrames(:, :, i-5)), 3);
        weighted_Frames(:, :, i) = uint8(0.4 * originalFrames(:, :, i-1) + 0.3 * originalFrames(:, :, i-2) + 0.1 *
     originalFrames(:, :, i-3) + 0.1 * originalFrames(:, :, i-4) + 0.1 * originalFrames(:, :, i-5));
    end
    for f = 2:numFrames
        for_mean_Frames(:, :, f) = uint8((abs(double(originalFrames(:, :, f)) - double(mean_Frames(:, :, f))) > T) *
     255);
        for_median_Frames(:, :, f) = uint8((abs(double(originalFrames(:, :, f)) - double(median_Frames(:, :, f))) > T) *
     255);
        for_weighted_Frames(:, :, f) = uint8((abs(double(originalFrames(:, :, f)) - double(weighted_Frames(:, :, f))) >
     T) * 255);
    end
    subplot(3, 3, 3*t-2);  imshow(for_mean_Frames(:,:,165));    title(sprintf("Mean-5, T=%d", T));
    subplot(3, 3, 3*t-1);  imshow(for_median_Frames(:,:,165));  title(sprintf("Median-5, T=%d", T));
    subplot(3, 3, 3*t);    imshow(for_weighted_Frames(:,:,165)); title(sprintf("Weighted-5, T=%d", T));
end
backgroundFrames2 = originalFrames; backgroundFrames5 = originalFrames;
backgroundFrames8 = originalFrames ;    for_backgroundFrames2 = originalFrames;
for_backgroundFrames5 = originalFrames ; for_backgroundFrames8 = originalFrames;
for i = 1:numFrames
    originalFrames(:, :, i) = rgb2gray(read(vidObj, i));
end
for f = 2:numFrames
    backgroundFrames2(:, :, f) = uint8(0.2 * originalFrames(:, :, f-1) + 0.8* backgroundFrames2(:, :, f-1));
    backgroundFrames5(:, :, f) = uint8(0.5 * originalFrames(:, :, f-1) + 0.5* backgroundFrames5(:, :, f-1));
    backgroundFrames8(:, :, f) = uint8(0.8 * originalFrames(:, :, f-1) + 0.2* backgroundFrames8(:, :, f-1));
end
for f = 2:numFrames
    for_backgroundFrames2(:, :, f) = uint8(abs(double(originalFrames(:, :, f)) - double(backgroundFrames2(:, :, f-
     1))));
    for_backgroundFrames5(:, :, f) = uint8(abs(double(originalFrames(:, :, f)) - double(backgroundFrames5(:, :, f-
     1))));
    for_backgroundFrames8(:, :, f) = uint8(abs(double(originalFrames(:, :, f)) - double(backgroundFrames8(:, :, f-
     1))));
end
figure();
subplot(1, 3, 1), imshow(for_backgroundFrames2(:, :, 165)); title(sprintf("Recursive-0.2"));
subplot(1, 3, 2), imshow(for_backgroundFrames5(:, :, 165)); title(sprintf("Recursive-0.5"));
subplot(1, 3, 3), imshow(for_backgroundFrames8(:, :, 165)); title(sprintf("Recursive-0.8"));
```