REAL-TIME TRACKING
REPORT

1.1 OBJECTIVE:

To implement the real-time object tracking algorithm using background subtraction and a Mixture of Gaussians.

1.2 CONCEPT:

The process of removing the static background from a series of video frames is known as background modelling. A method known as background subtraction enables the foreground of an image to be retrieved for subsequent processing(object detection, etc.) and is typically utilised after the backdrop has been modelled. We will perform the background modelling using Gaussian Mixture Models. The Mixture of Gaussians, or MoG, is a collection of Gaussian distribution models.

1.3 DATASET:

Dataset: A Video File is created and used for tracking. It is in the **.mp4** format. The video can be found here:

https://github.com/MannJain1609/Real-Time-Object-Tracking/blob/main/Video%20File/VID_MJ.mp4

Description: I extracted frames from the video with 10 frames per second. The number of frames extracted is 28. The frames are stored in a "Frames" directory. All the frames are image files of the format **.jpg**, compressed to 100*140 dimensions for better efficiency and reduced runtime.

1.4 ALGORITHM:

The algorithm is as follows:

1. Frames are extracted from the video file(Grayscale) using the saving_frames_ per_second parameter, set to 10 here.

2. Each pixel is modelled as a mixture of Gaussians (K-Gaussians). The value of K in our algorithm is 4.

3. The values of a particular pixel over time are called the pixel process. The pixel process is the time series of pixel values, scalars for grayvalues or vectors for color images(RGB).

4. Recent history of each pixel (X1, X2,...Xt) is modelled by a mixture of K-Gaussian distributions.

5. Probability of observing current pixel value:

$$P(X_t) = \sum_{i=1}^{K} \omega_{i,t} * \eta(X_t, \mu_{i,t}, \Sigma_{i,t})$$

Here,

K = no. of distributions
$w_{i,t}$=weight of ith gaussian at time t
$\mu_{i,t}$= mean value of ith gaussian in the mixture at time t
$\Sigma_{i,t}$= Covariance Matrix of the ith gaussian in the mixture at time t
η= Gaussian probability density function(pdf)

$$\eta(X_t, \mu, \Sigma) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(X_t - \mu_t)^T \Sigma^{-1} (X_t - \mu_t)}$$

6. Covariances matrices are assumed the form $\Sigma_{k,t} = \sigma_k^2 \mathbf{I}$, for efficient computation. We assume that RGB values are independent and variances are the same.

7. If the pixel process should be considered a stationary process, a standard method for maximizing the likelihood of observed data is expectation maximization(EM), but the pixel process varies over time,

8. We use the k-means approximation We find whether a pixel value matches a gaussian distribution. A match is when the pixel value is within 2.5 standard deviations of a distribution.

9. If none of the K distributions matches the current pixel value, the least probable distribution is replaced with a distribution with the current value as the mean value, initially high variance and low prior weight.

10. If any or more of the distribution matches, then the weights are changed as follows:

$$\omega_{k,t} = (1 - \alpha)\omega_{k,t-1} + \alpha(M_{k,t})$$

$M_{k,t}$ = 1 for models matched, else 0
α = Learning rate

11. The mean and covariance matrix of the unmatched distributions remains the same. The mean and covariance matrix for the matched distribution changes as follows:

$$\mu_t = (1-\rho)\mu_{t-1} + \rho X_t$$

$$\sigma_t^2 = (1-\rho)\sigma_{t-1}^2 + \rho(X_t - \mu_t)^T(X_t - \mu_t)$$

where the second learning rate[3], ρ, is

$$\rho = \alpha\eta(X_t|\mu_k, \sigma_k)$$

12. Gaussians are ordered in decreasing order by the ratio value (weights/standard deviation).

13. First B Gaussian distributions are chosen for the background model as per the threshold T.

$$B = argmin_b \left(\sum_{k=1}^{b} \omega_k > T \right)$$

14. If the image in a frame matches any of the first B distributions for that pixel, it is modelled as background. Else, it is modelled as foreground. The background pixels are turned to black to subtract them from the image. Thus, the resulting image has no(dark) background.

1.5 RESULTS:

The GMM algorithm implemented by me gave pretty good results as per the perception. The minimum and maximum RMSE between the output generated by my function and the inbuilt 'CreateBackgroundSubtractorMOG2()' is as follows:
Minimum RMSE: 46.68
Maximum RMSE: 155.23

1.6 REFERENCES:

1. C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149), 1999, pp. 246-252 Vol. 2, doi: 10.1109/CVPR.1999.784637.