

# Real Time Tracking using Gaussian Mixture Models

## Objective:

We have to get the real-time tracking of the objects in a given video, by using the Gaussian mixture model.

## Method:

1. **Extraction of the frames:** Initially, we have taken a video and extracted them into a file with around 80 frames.
2. **Parameters:** We define all the required parameters with the initial values. We have given the learning rate, weights. Each pixel with respect to time can be represented as below.

$$P(X_t) = \sum_{i=1}^K \omega_{i,t} * \eta(X_t, \mu_{i,t}, \Sigma_{i,t}) \quad \eta(X_t, \mu, \Sigma) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(X_t - \mu_t)^T \Sigma^{-1} (X_t - \mu_t)}$$

3. **Modeling the Gaussian distributions:**

To create the Gaussian mixture of models for every pixel and for every frame.

We have to follow the following procedure.

If there is at least one gaussian match, return true when  $(x_t - \mu_t)/\sigma$  is less than or equal to 2.5 and return false if its value is greater than 2.5.

If there is no match with any gaussian mixtures then, We create a new Gaussian for  $X_t$  which has the mean with current pixel. It has low weight and a large sigma. We replace the Gaussian with the least weight in case there are already K of them present.

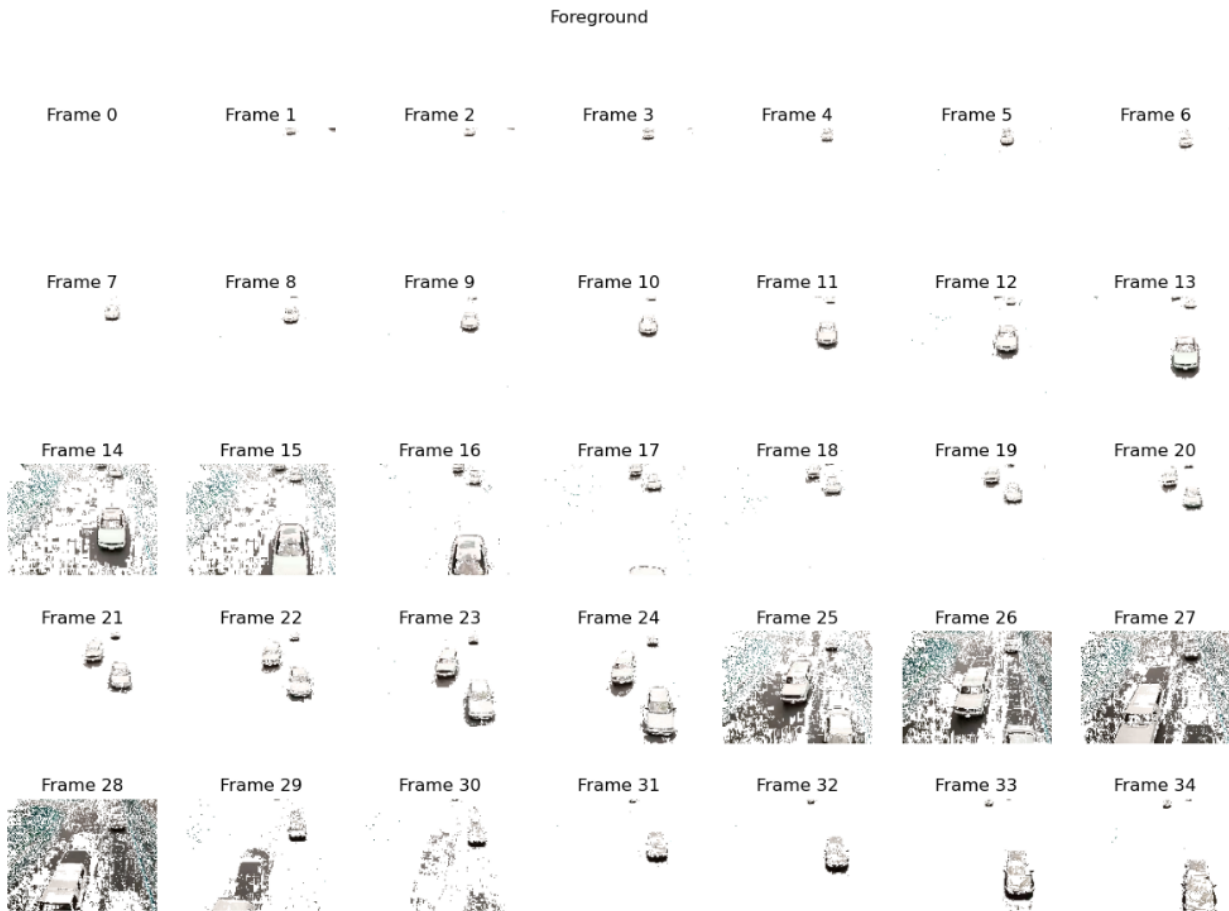
4. **Classifying the distribution as foreground and Background gaussians :**

For a pixel, we have to divide the distributions. We will sum all the weights of the gaussian in the descending order of the weight/standard deviation. If it is greater than threshold then those gaussians before it occurs to the threshold are considered as background gaussians, and the gaussians after them are considered as foreground gaussians.

5. **Classifying the pixels as foreground and Background:**

For a pixel, we can divide the pixel as foreground if it is not matched with any of the gaussians presented. Otherwise it can be taken as a background pixel.

## Foreground\_frames:



The above images are the first 35 frames of the given input video which has real time tracking of the objects

## Resources:

- <https://ieeexplore.ieee.org/document/784637>
- <https://hal.archives-ouvertes.fr/hal-00338206/en/>
- <https://www.sciencedirect.com/science/article/abs/pii/S1574013718303101?via%3Dihub>
- <https://www.geeksforgeeks.org/extract-images-from-video-in-python/>
- <https://stackoverflow.com/questions/22748617/python-avi-to-mp4>