COL780 : Computer Vision
Assignment 1

Gaurav Jain : 2019CS10349

1 Method

I have implemented the Gaussian Mixture Model Method for Background Subtraction with constant weights for the past N frames (Y = 1).

1.1 Gaussian Mixture Models

Stauffer and Grimson used the Mixture of Gaussians (MOG) to model dynamic backgrounds. The recent history of the intensity values (in RGB color space) of each pixel $X_1, \ldots, X_t$ is modeled by a mixture of K Gaussian distribution. The probability of observing the current pixel value is given by the formula:

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

where K gives the number of Gaussian distributions, $\omega_{k,t}$ is the weight of the k^{th} Gaussian in the mixture at time t having mean $\mu_{k,t}$ and covariance matrix $\Sigma_{k,t}$ and η is a Gaussian probability density function.

For computational reasons it is assumed that the red, green and blue color components are independent and have the same variances. The authors propose the value of K from 3 to 5. K-means algorithm is used in the original paper to update the values of weights, means and co-variances of the mixture of Gaussians in an online fashion.

1.2 Gaussian Mixture Models with Constant Weights for past N frames

The main difference between this method and the original method proposed by Stauffer and Grimson is in the update equations for weights, mean and co-variances of the GMMs. Instead of comparing all frames in time t, only N-recent window samples are processed which increases the processing speed of the algorithm.The N-recent window update equations give priority to recent data therefore the tracker can adapt to changes in the environment effectively.
The update equations of the N-recent window version are:

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

$$\mu_{k,t} = \mu_{k,t-1} + \frac{1}{N}(\frac{M_{k,t}X_t)}{\omega_{k,t}} - \mu_{k,t-1})$$

$$\Sigma_{k,t} = \Sigma_{k,t-1} + \frac{1}{N}(\frac{M_{k,t}(X_t - \mu_{k,t-1})(X_t - \mu_{k,t-1})^T}{\omega_{k,t}} - \Sigma_{k,t-1})$$

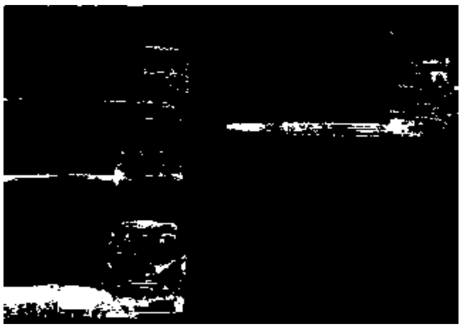
I have implemented the above update equations for updating weights, means and co-variances of mixture of Gaussians. I have taken $N = 10$ and $K = 4$ for the implementation. Similar to the original paper, I have ordered the K Gaussians according to the ω/σ ratio. The first B Gaussian distributions which exceed certain threshold T are retained for the background distribution.

$$B = \arg\min_b(\sum_{k=1}^{b} \omega_k > T)$$

The value of T in my implementation is 0.7.

2 Observations

- There are 5 datasets on which we evaluate the performance. They are Candela, Caviar, HighwayI, HallAndMonitor and IBMtest2.
- The model show decent performance on IBMtest2 and HighwayI datasets. The model is able to separate foreground (people) from background in IBMtest2 and foreground (cars) from background in HighwayI. Sample output of a frame of each dataset is present below:


(a) HighwayI

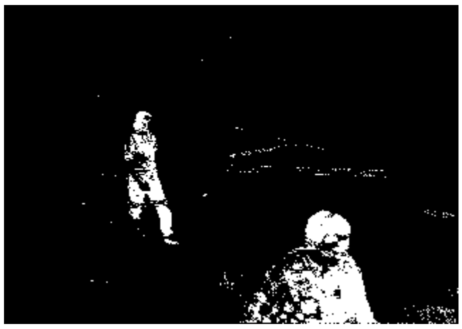
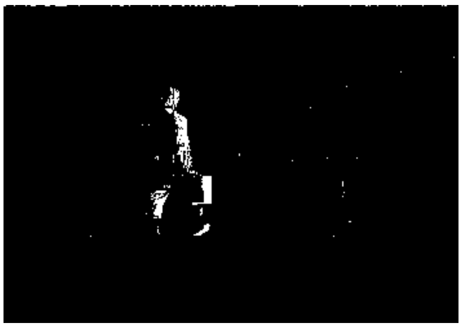

(b) IBMtest2

Figure 1: Sample output of a frame for HighwayI and IBMtest2

- The model shows decent performance on the HallAndMonitor dataset initially but due to lighting changes, there is a lot of noise in the foreground mask in later frames. The sample output for HallAndMonitor is present below:


(a) t = 40

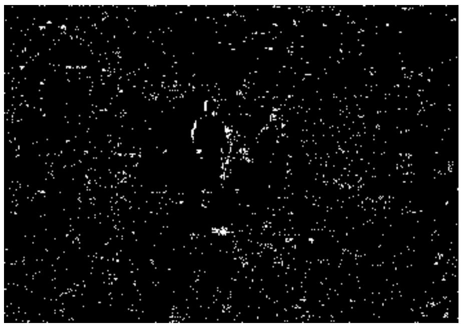
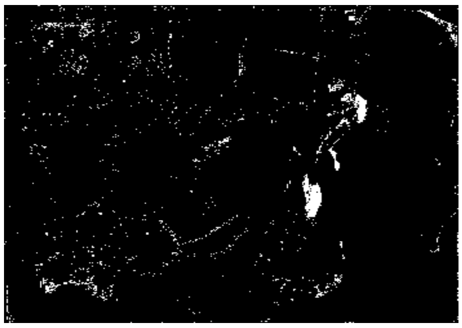

(b) t = 180

Figure 2: Sample output of a frame for HallAndMonitor

- The model shows relatively poor performance on Candela and Caviar datasets, there is a lot of noise in the model due to abrupt changes in the lighting conditions of the surrounding. The people in the datasets are not very recognisable in the output. Sample output is shown below:


(a) Candela

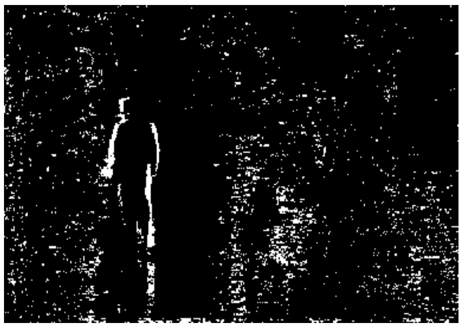
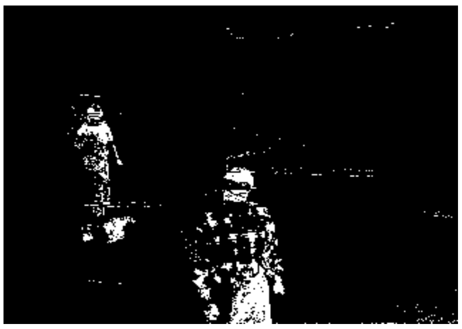

(b) Caviar

Figure 3: Sample output of a frame for Candela and Caviar

3 Comparison of Performance

I have compared the performance of my model against kernel-density based method with decaying weights implemented by Nishant Kumar (2019CS50586). He has kept a threshold of 1e-5 and used past 20 frames for calculating kernel density. The comparison is done on all datasets.

3.1 IBMtest2


(a) GMM with constant weights

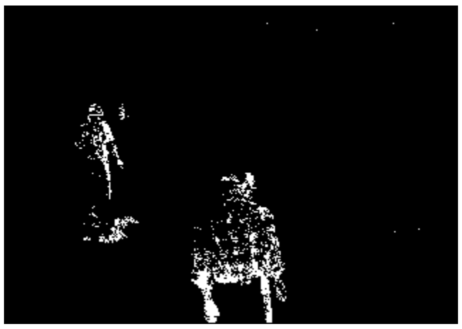

(b) KDE with decaying weights

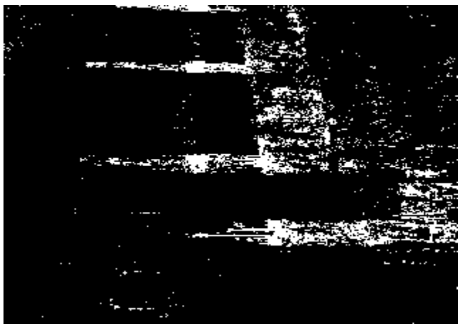
Figure 4: Comparison of outputs in IBMtest2 dataset

The performance of both methods are good on this dataset. Both methods are able to separate foreground from the background. The lighting conditions is stable and hence, both models are able to perform good.

3.2 HighwayI

(a) GMM with constant weights

(b) KDE with decaying weights


(c) GMM with constant weights

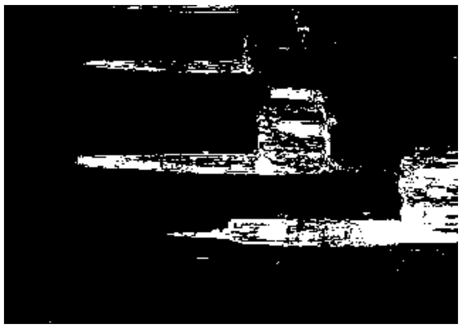
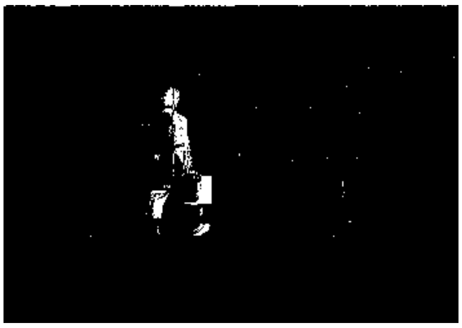
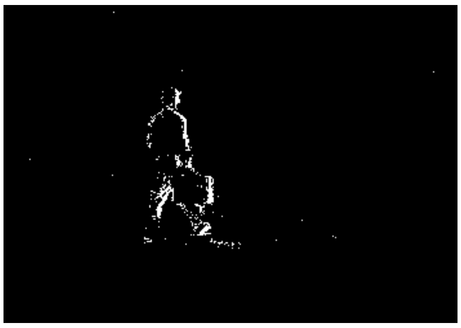

(d) KDE with decaying weights

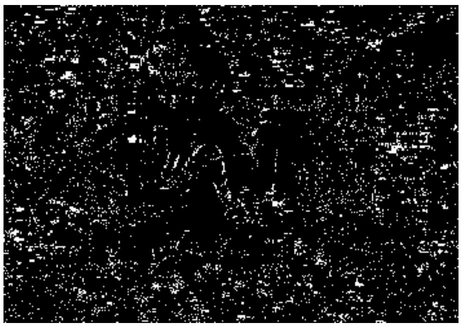
Figure 5: Comparison of outputs in HighwayI dataset

The performance of GMM with constant weights is relatively weak to KDE with decaying weights. In the initial frames, the performance is similar. Both in the later frames, the performance of GMM with constant weights declines. The update equations only considers last 10 frames so, the change is aggressive in GMM with constant weights and hence, the performance is not good.

3.3 HallAndMonitor


(a) GMM with constant weights


(b) KDE with decaying weights


(c) GMM with constant weights

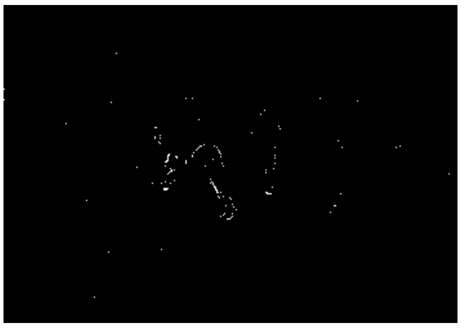
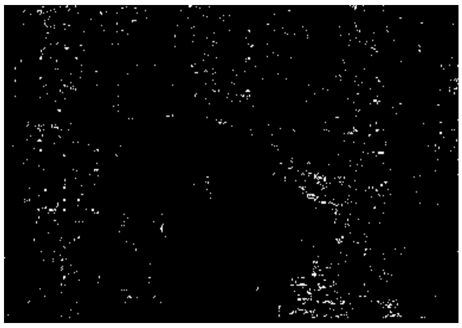

(d) KDE with decaying weights

Figure 6: Comparison of outputs in HallAndMonitor dataset

The performance of both models is not good in the HallAndMonitor dataset. GMM with constant weights is able to detect in the initial frames but after some frames, it starts to perform poorly with lot of noise in the foreground pixels. KDE with decaying weights is not able to identify foreground pixels and only few pixels are visible.

3.4 Caviar


(a) GMM with constant weights

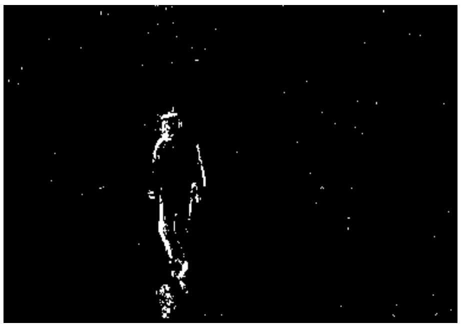

(b) KDE with decaying weights

Figure 7: Comparison of outputs in Caviar dataset

As visible in the above sample output, GMM with constant weights is not able to detect the person in the frame whereas, KDE with decaying weights is able to perform background subtraction. This is a failure case for GMM with constant weights.

3.5 Candela

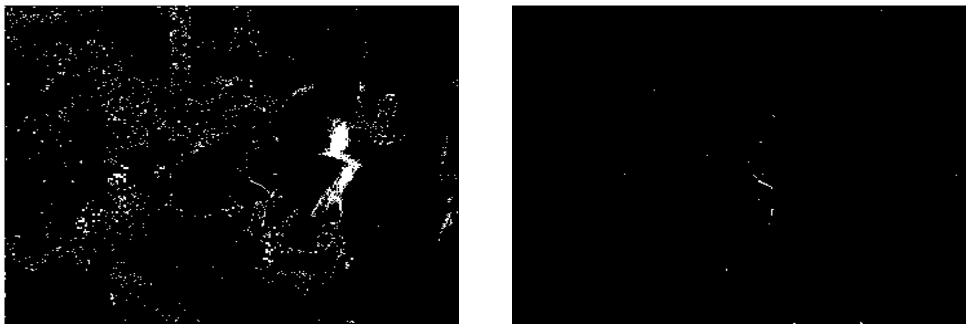

(a) GMM with constant weights (b) KDE with decaying weights

Figure 8: Comparison of outputs in Candela dataset

As visible in the above sample output, both models are not able to detect the person in the frame. GMM with constant weights contains too much noise whereas, KDE with decaying weights produces much less noise. Another reason for KDE with decaying weights to not able to detect is it absorbs the person sitting in the input frames into its background.

4 Cleaning up results with foreground pixel aggregation

I cleaned up the output frame got from the model using foreground pixel aggregation. I used a 3x3 filter to pass through each frame and aggregated foreground pixels according to some threshold (T = 0.6). But, it was created very few white pixels and thus, I decided to remove it from my implementation.

5 Link

This link contains the output videos for all datasets after performing background subtraction.

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), Fort Collins, CO, USA, 1999, pp. 246-252 Vol. 2, doi: 10.1109/CVPR.1999.784637.

[2] Goyal, K., Singhai, J. Review of background subtraction methods using Gaussian mixture model for video surveillance systems. Artif Intell Rev 50, 241–259 (2018). https://doi.org/10.1007/s10462-017-9542-x