

Background Subtraction using Gaussian Mixture Models

Understanding of the algorithm

- The assignment requires us to implement a GMM based algorithm for real time tracking of object using background subtraction.
- To implement this algorithm, I have taken a video clip of moving cars as traffic on road (link to the clip is mentioned in the references section).
- Since the algorithm is applied on frames of the video. I first convert the video clip into frames using cv2 's functionalities.
- Then I store these frame images in a separate folder named frames. I create two 3D matrices to store all the frames of the video.
- I store these images in grayscale so as to reduce the time complexity of the algorithm.
- Then I initialise the model parameters used for training the model. This includes the threshold value which will be used to distinguish the Gaussian distributions of background from foreground, number of Gaussians to be used in the mixture of gaussians, learning rate (alpha).
- We iterate through all the pixels of the image and implement the following:
 1. We create a 2D array corresponding to each of the pixel which will contain information about the various parameters of the gaussian, and of the shape total number of gaussians in gaussian mixture model (GMM) times 5 (the number of parameters that needs to be changed during the algorithm).
 2. For the first Gaussian we initialise it with the proper values as shown in the code whereas for other gaussian we initialise them with zero and the values of the parameters of Gaussian gets updated as the algorithm proceeds ahead in time.
 3. Then to update the parameters of these gaussians for each time frame of the pixel we iterate the parameters over each time frame and over each gaussian.
 4. The pixel is a match for the Gaussian if the pixel lies within 2.5 standard deviation of the distribution.
 5. We update the parameters such that if none of the 5 distributions match the current pixel, the least probable distribution is replaced with a distribution with current value as its mean. Further the prior weights of the distributions at time are adjusted as:

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

Here $w(k,t)$ is the weight corresponding to the k th distribution and for time stamp t , α is the learning rate and $M(k,t)$ is 1 for the models that matched and 0 for the models that do not match.

6. The mean and the standard deviation for the unmatched distribution remains the same and the parameters of the distribution that matches the new observation are updated as:

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

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

Here Rho is the second learning rate represented in the code by lr2 and is given by:

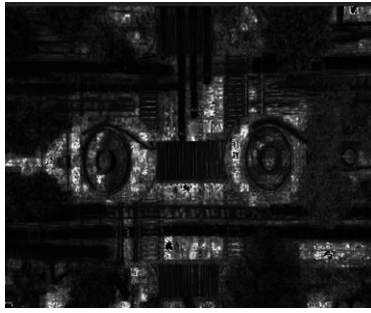
$$\rho = \alpha\eta(X_t|\mu_k, \sigma_k)$$

In the implemented code the mapping of the above variables and the code variable for kth Gaussian is as follows:

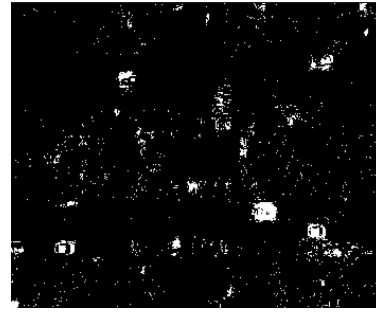
- a. Gaussian_param[k,0] --> Mean (mu)
 - b. Gaussian_param[k,1] --> Variance (sigma^2)
 - c. Gaussian_param[k,2] --> 2.5 * (Standard Deviation)
 - d. Gaussian_param[k,3] --> Weight of the Gaussian component (w)
 - e. Gaussian_param[k,4] --> w/sigma
7. After iterating through all the Gaussians for a given pixel at a given time frame. We sort these gaussians based on w/sigma value. This is based on the fact that as this value increases both distribution gains more evidence and variance decreases. Thus, indicating that the pixel has been constantly same for a long period of time indicating background pixel.
 8. We find the sum of the gaussians such that their sum is greater than the threshold and if the pixel lies matches with any of the other Gaussians apart from these then it is in foreground. Thus, the matrices corresponding to both are accordingly updated.
 9. We then create the images of frames using these two arrays and store them in separate folders named background and foreground. Then we make video clip from them to demonstrate the working of the algorithm.

Results obtained

- The results obtained are stored in the foreground.avi and background.avi videos.
- The results obtained are black and white images where the white spots in the image denotes the moving objects.
- The results can be as shown below:



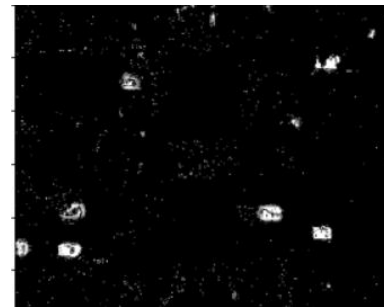
Background



Foreground



Original



Foreground obtained from CV2

- It is evident from above that the original image has been properly separated into Foreground and Background image. Further, on comparison with the inbuilt function of CV2 we can see that the result closely matches.

Accuracy and Inference

- Based on the research paper there is no metric to calculate the accuracy of the method.
- Still to compare the results, I computed the foreground for all the frames using the cv2 function. I obtained the MSE loss for the matrix obtained using my implementation of algorithm and that of CV2. The result obtained were fairly good as the minimum MSE loss for any pixel was 659 and maximum was 16137. I compared the frame 24 for both implementation (as shown above) and the result obtained is accurate as both depict similar features.
- We can further enhance the results by increasing the number of Gaussians in the model but this would subsequently increase the time to train. It will also lead to overfitting as when we will expose it to new data then it will predict completely based on the training set and not on some patterns.
- We can also fine tune the model by increasing or decreasing the learning rate.

Assumption

- For better results video has been taken such that the background remains constant and does not move so that proper separation can be made.
- Images frame in grayscale are considered and if not in grayscale, then are converted to greyscale for faster calculation.
- Threshold has been heuristically defined. Much experiment based on its value were not done since the training time of the model is in several minutes.

References:

- Video source: <https://mixkit.co/free-stock-video/city-busy-traffic-intersection-time-lapse-1755/>
- <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=784637&tag=1>
- <https://github.com/ZiyangS/BackgroundSubtraction>
- <https://github.com/suneelpadala529/Background-foreground-subtraction-in-a-video>