Real-Time Tracking using Gaussian Mixture Models

Understandings:

1. GMMs allow for an observation to belong to more than one cluster with a level of uncertainty. GMMs learn the probabilities of that example to belong to each cluster k.
2. For each Gaussian, it learns one mean and one variance parameter from the data.
3. To learn such parameters, GMMs use the expectation-maximization (EM) algorithm to optimize the maximum likelihood. In the process, GMM uses Bayes Theorem to calculate the probability of a given observation x_i to belong to each cluster k
4. EM can be simplified in 2 phases: The E (expectation) and M (maximization) steps. In the E step, we calculate the likelihood of each observation x_i using the estimated parameters.
5. $$f(\mathbf{x}|\mu_k, \sigma_k^2) = \frac{1}{\sqrt{2\pi\sigma_k^2}} \exp(-\frac{(\mathbf{x} - \mu_k)^2}{2\sigma_k^2})$$
6. For each cluster k, we calculate the probability density (pdf) of our data using the estimated values for the mean and variance.
7. Then, we can calculate the likelihood of a given example x_i belonging to the k^{th} cluster.
$$\mathbf{b}_k = \frac{f(\mathbf{x}|\mu_k, \sigma_k^2)\phi_k}{\sum_{k=1}^{K} f(\mathbf{x}|\mu_k, \sigma_k^2)\phi_k}$$
8. Then, in the maximization, or M step, we re-estimate our learning parameters as follows.
$$\mu_k = \frac{\sum \mathbf{b}_k \mathbf{x}}{\sum \mathbf{b}_k} \qquad \sigma_k^2 = \frac{\sum \mathbf{b}_k (\mathbf{x} - \mu_k)^2}{\sum \mathbf{b}_k} \qquad \phi_k = \frac{1}{N} \sum \mathbf{b}_k$$
9. We may repeat these steps until converge. That could be up to a point where parameters' updates are smaller than a given tolerance threshold. At each iteration, we update our parameters so that it resembles the true data distribution.

Result Obtained:

We used a small video clip (link of the video clip mentioned in code comment) as our data set which consist of moving car and humans along with their shadows.

The final result was separation of background and moving objects. Results are attached with the zip file.

Accuracy:

For calculating accuracy of the model , built-in function in opencv libarary (cv.BackgroundSubtractorMOG2) was used. And the last frame of output of this function was compared with last frame output of our model to calculate accuracy keeping threshold as 5 for difference in each pixel. The accuracy was obtained to be nearly 85%.

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.
2. github.com/Mainak1792/Backround_Subtraction/blob/main/assets/umcp.mpg
3. https://towardsdatascience.com/how-to-code-gaussian-mixture-models-from-scratch-in-python-9e7975df5252
4. Thierry Bouwmans, Fida El Baf, Bertrand Vachon. Background Modeling using Mixture of Gaussians for Foreground Detection - A Survey. Recent Patents on Computer Science, Bentham Science Publishers, 2008, 1 (3), pp.219-237. ⟨hal-00338206⟩