Machine Learning Engineer Nanodegree

Capstone Proposal

Dominik Söllinger
August 17th, 2018

Foreground-background segmentation using Gaussian mixture models

Domain Background

Background subtraction is a common pre-processing step in many computer vision related tasks. Many applications [1][2] only require information of changes in a scene since such regions typically contain the objects we are interest in. For instance, cars that drive on a highway or pedestrians the walk on a sidewalk. Background segmentation allows us to detect such objects by a pixel-wise segmentation of a frames into foreground and background regions.

Deep learning enthusiast might now suggest the use of novel neural networks like UNET [3] to solve such segmentation tasks. However, there are well-studied algorithms like the one proposed by Stauffer and Grimson [4] that provide an unsupervised learning based solution based on Gaussian mixture models (GMMs) to the segmentation problem. The fact that these GMM based algorithms are well-studied and don't require extensive training data strongly advocate its use for foreground-background segmentation problems.

This is also the reason why I want to implement the GMM based segmentation approach suggested by Stauffer and Grimson myself. So far, I only got in touch with neural network based segmentation approaches. However, it's inspiring to me that similar things can be done using relative simple GMM based solutions without requiring a labeled dataset. Since I haven't seen any code that implements, explains and demonstrates this algorithm in Python it would be a great capstone project and hopefully also help others to get a better intuition of how this algorithm works.

Problem Statement

Goal of this project is to develop a solution for unsupervised foreground-background video segmentation and assess its performance. For performance evaluation we will assess the model's performance on the LASIESTA [5] dataset. A good model should finally be able to segment any video into a foreground or background regions after "seeing" a few frames. Ideally, the model should be able to perform this segmentation task in real-time.

Datasets and Inputs

We will use the LASIESTA [5] dataset to train and test the implemented model. The dataset comprises of eight different scenes captured in different indoor and outdoor environments. These scenes cover a broad spectrum of image distortions we may encounter in real-world meaning that we have to deal with illumination changes, occlusion and shadows.
In each scenario we are given frames of the original image (24bpp BMP) as well as the corresponding labels (ground truth) for every frame.

Label data are given as images where every pixel value uniquely assigns the pixel to a segment:

- Black pixels (0,0,0): Background.
- Red pixels (255,0,0): Moving object with label 1 along the sequence.
- Green pixels (0,255,0): Moving object with label 2 along the sequence.
- Yellow pixels (255,255,0): Moving object with label 3 along the sequence.
- White pixels (255,255,255): Moving objects remaining static.
- Gray pixels (128,128,128): Uncertainty pixels.

Solution Statement

Various research papers propose the use of Gaussian mixture models for foreground-background segmentation. In the project we want to take a closer look at the approach described by Stauffer and Grimson in [4].

The idea is to fit a Gaussian mixture model to a time series of image pixels. In other words, we are given video and take a certain number of frames out of this video. Next, we consider what's called a "pixel process". The "pixel process" is a time series of pixel values, for example, scalars for gray values or vectors for color images.

At any time t we know the history (intensity values) of a certain pixel:

$$\{X_1, \ldots, X_t\} = \{ \ I(x, y, i) : 1 \leq i \leq t \ \} \qquad \text{where } I \text{ is the image sequence}$$

We can now use this time series to estimate the probability of seeing a certain pixel value. Formally, this means that we can estimate the probability of a certain pixel value by fitting a GMM on the recent history of pixel values.

The probability of observing the current pixel value is

$$P(x) = \sum_{I=1}^{K} w_{i,t} \cdot \eta(X_t, \mu_{i,t}, \Sigma_{i,t})$$

where K is the number of distributions, $w_{i,t}$ is an estimate of the weight (what portion of the data is accounted for by this Gaussian) of the i-th Gaussian in the mixture at time t, $\mu_{i,t}$ is the mean value of the i-th Gaussian in the mixture at time t, $\Sigma_{i,t}$ is the covariance matrix of the i-th Gaussian in the mixture at time t, and where η is a Gaussian probability density function.

$$\eta(X_t, \mu_{i,t}, \Sigma_{i,t}) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma|^{\frac{1}{2}}} \cdot e^{-\frac{1}{2}(X_t - \mu_t)^T \Sigma^{-1} (X_t - \mu_t)}$$

Note: K depends mainly on the modality of the background distribution, but for implementation purposes factors like the available computational power and real time requirements have to be considered. In practice, it has been shown that 3 to 5 is a reasonable choice for K.

Background Model estimation

The distribution of recently observed values of each pixel in the scene is now characterized by a mixture of Gaussians. This mixture can now be used to estimate the probability that a certain pixel value belongs to a background or foreground region. The idea is similar to an algorithm referred as Bog-of-words (BoW) classification. Some Gaussians are more likely to represent a background region than others. If we know which Gaussians represents background objects, we can assign new pixel values to background / foreground by calculating its proximity to background Gaussians.

To understand this, consider the accumulation of supporting evidence and the relatively low variance for the "background" distributions when a static, persistent object is visible. In contrast, when a new object occludes the background object, it will not, in general, match one of the existing distributions which will result in either the creation of a new distribution or the increase in the variance of an existing distribution. Also, the variance of the moving object is expected to remain larger than a background pixel until the moving object stops. [4]

To perform this classification for a pixel we order the Gaussians by the value of ω/σ. This value increases both as a distribution gains more evidence and as the variance decreases. This ordering of the model is effectively an ordered, open-ended list, where the most likely background distributions remain on top and the less probable transient background distributions gravitate towards the bottom.

Now, the first B distributions are chosen as background model that account for a predefined fraction of the evidence P.

$$B = \operatorname{argmin}_b \sum_{k=1}^{b} w_k > T$$

Classification of pixel values

As we know which Gaussian best represent background regions, we can now use them to classify new pixel values. A pixel value is considered as "close" to the Gaussian if the value is not more than 2.5 standard deviations away from its mean. If this is the case, we classify the point as background pixel.

Such a proximity measure is the **Mahalanobis distance**. It measures how many standard deviations away a point is from the mean μ.

$$D(x) = \sqrt{(x-\mu)^T \Sigma^- 1 (x-\mu)}$$

Evaluation Metrics

A good way to assess the quality of the developed model is the F1-score [6] [7]. It's defined as the harmonic average of the precision and recall, where an F1 score reaches its best value at 1 and worst at 0.

$$F_1 = 2 \cdot \frac{\text{Precision}\cdot\text{Recall}}{\text{Precision}+\text{Recall}}$$

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive}+\text{False Positive}}$$

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive}+\text{False Negative}}$$

In our problem domain *True Positive*, *False Positive*, *False Negative* and *False Positive* can be understood as:

- **False Positive:** Classified as background pixel, but it's a foreground pixel
- **False Negative:** Classified as foreground pixel, but it's a background pixel
- **True Positive:** Classified as background pixel and it's a background pixel
- **True Negative:** Classified as foreground pixel and it's a foreground pixel

Benchmark Model

Fortunately, benchmarks are provided on the website of the LASIESTA dataset. This benchmarks illustrate the obtained F1-score for eight different background subtraction algorithms including the algorithm suggested by Stauffer. Scores are provided for individual scenario as well as an average F1-score.

Unfortunately, the website is not clear on how these score were obtained meaning that it's not fully clear whether they first trained the GMM on the whole scenario to get a background reference image or if the updated the model framewise while computing the score. As the second approach would be similar to a real-world setting where we gradually have to update our model, it seems the better approach for evaluating this algorithm.

Project Design

Keeping the model up-to-date

We already discussed the basic idea in the "Solution Statement" section. However, so far, we haven't discussed how to fit the Gaussian mixture model to the different frames.
A standard method for finding the parameters of the GMM is expectation maximization (EM) [8] algorithm. However, as the pixel process varies over time due to further frames we capture, implementing an exact EM algorithm on a window of recent data would be costly. Instead, we implement an **on-line k-means**

approximation.

Every new pixel value, X_t, is checked against the existing K Gaussian distributions until a match is found (within the range of 2.5 standard deviations of the distribution). If none of the K distributions match the current pixel value, the least probable distribution is replaced with a distribution with the current value as its mean value, an initially high variance, and low prior weight.

The prior weights of the K distributions at time t, ω_k, t, are adjusted as follows:

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

where α is the learning rate and $M_{k,t}$ is 1 for the model which matched and 0 for the remaining models. After this approximation, the weights are renormalized.

Furthermore, also the parameters of the distribution which matches the new observation have to be update:

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

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

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

One of the significant advantages of this method is that when something is allowed to become part of the background, it doesn’t destroy the existing model of the background. The original background color remains in the mixture until it becomes the K-th most probable and a new color is observed.

Processing pipeline

The following steps have to be performed for every pixel value X_t of a new frame:

1. Sort the existing distributions according to ω/σ
2. Selects the first B distributions that account for a predefined fraction of the evidence $B = \operatorname{argmin}_b \sum_{k=1}^{b} w_k > T$ and consider them as background distributions
3. Checks if the incoming pixel value X_t can be assigned any distribution

    - If the pixel can be assigned to a background distribution, label it as background pixel
    - If the pixel can be assigned to a background foreground, label it as foreground pixel
    - If the pixel does not match to any distribution, label it as foreground

4. Update model

References

[1] Cheung S, Kamath C. Robust background subtraction with foreground validation for Urban Traffic Video. J Appl Signal Proc, Special Issue on Advances in Intelligent Vision Systems: Methods and Applications (EURASIP 2005), New York, USA, 2005; 14: 2330-2340

[2] Carranza J, Theobalt C, Magnor M, Seidel H. Free-Viewpoint Video of Human Actors, ACM Trans on Graphics 2003; 22(3): 569-577

[3] Olaf Ronneberger, Philipp Fischer, Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. CoRR, May 2015

[4] Stauffer C, Grimson W. Adaptive background mixture models for real-time tracking. Proc IEEE Conf on Comp Vision and Patt Recog (CVPR 1999) 1999; 246-252.

[5] http://www.gti.ssr.upm.es/data/LASIESTA

[6] https://en.wikipedia.org/wiki/Precision*and*recall

[7] https://en.wikipedia.org/wiki/F1_score

[8] A Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society, 39 (Series B):1-38, 1977.