A Report for Assignment -1 of COL780
Computer Vision

On

Implementation of kernel-density-based method with decaying weights

Submitted to

Anurag Mittal

(Instructor COL 780)

by

Nitesh Dohre (2022MCS2070)

(Y=2)

Department of Computer Science and Engineering

Indian Institute of Technology, Delhi

Session: 2022-23

CONTENTS:

INTRODUCTION:

The kernel density function is a non-parametric technique used in background subtraction to model the distribution of the pixel intensity values in a video stream. It involves estimating the background pixel's probability density function (PDF) using a kernel function, which assigns a weight to each pixel in the image based on its similarity to the surrounding pixels. The resulting PDF represents the background model and can be updated continuously as new frames are processed. The foreground objects are then detected as regions where the pixel intensity values deviate significantly from the background model. This method can handle complex scenes with varying illumination conditions and moving objects, making it a helpful approach for background subtraction in real-world scenarios.

TOOLS AND TECHNOLOGY USED:

1. Jupyter notebook IDE
2. Python3
3. Libraries: numpy, matplotlib, cv2, maths, os
4. Datasets provided

Algorithm of implementation:

Load all images from the input directory into a list of numpy arrays.
Transpose of each image array and store all image arrays in a single numpy array called "mat".
Initialize variables:
1. Set the threshold value "th" to 0.04.
2. Get the number of rows and columns of each image from the shape of "frame_matrix".
3. Create a zero numpy array called "res" with the same number of rows and columns as the images.
For each time step "frame" in the range of 1 to the number of images:
a. For each pixel (i, j) in the image:
i. Calculate the value of "sigma" based on the current time step "k".
ii. Calculate the values of constant factors "outmult" and "inmult".
iii. Normalize the red, green, and blue channels of the first "frame" images.
iv. Apply exponential decay to the red, green, and blue channels with a decay rate of "alpha".
v. Calculate the sum of the exponentials of the squared distances between each channel of the current image and the last image.
vi. Multiply the sum by "outmult" to get the final value for the current pixel.
vii. If the value is greater than or equal to the threshold, set the value to 1, otherwise, set it to 0.
b. Transpose "res", multiply it by 255,
c. Call the 'noise_reduction' function to reduce the noise
d. save it as an image with the current time step index in the output directory.
Repeat steps 4a-4b for all time steps until all images have been processed.
5. End of algorithm

Explanation:

In this assignment, I have implemented a paper named **Non-parametric Model for Background Subtraction by** *Ahmed Elgammal, David Harwood, Larry Davis.*

$$Pr(x_t) = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{(2\pi)^{\frac{d}{2}} \mid \Sigma \mid^{\frac{1}{2}}} e^{-\frac{1}{2}(x_t - x_i)^T \Sigma^{-1}(x_t - x_i)}$$

The above expression is a probability distribution function which calculates the average kernel density for n frames for all 3 channels (red, green, blue).

I_new = alpha*I_curr + (1-alpha)*I_old

the above expression is used to calculate the new intensity value of all the channels (red, green, blue) respectively.

In my implementation, I have considered 99% stable, i.e. 1% decaying rate.

If the intensity value is greater than the threshold, then is a background pixel, otherwise foreground.

Results:

I have compared the output generated by my code and the ground-truth output provided, and we got the following results:

1.IBMtest2 accuracy: 0.9419167877906974
2.HallAndMonitor accuracy: 0.9612861680806356
3. HighwayI accuracy: 0.8034138773413603
4. Caviar accuracy: 0.9312246089721269
5. Candela's accuracy: 0.9485805175852726

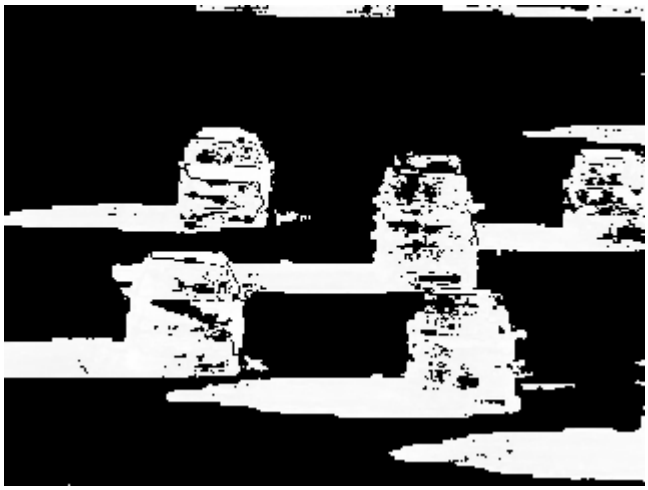
Comparison with another model:

I have compared the output generated by my implementation of Background Subtraction using the kernel density Function with another implementation of Background Subtraction using the kernel density Function and implementation of Background Subtraction using the Gaussian Mixture model and got the following results:

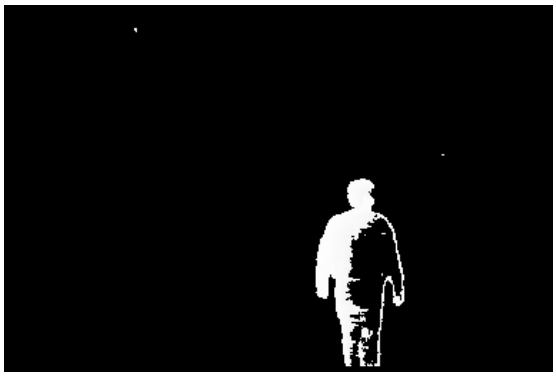
DATASET	My Accuracy of KDE (in percentage)	Another KDE (in percentage)	GMM Model (in percentage)
CANDELA	94.85805175852726	94.85497188740719	95.49788836249043
CAVIAR1	93.12246089721269	91.38	95.20342567704044
IBMtest2	94.19167877906974	94.12740734011624	94.41861979166667
HallAndMonitor	96.12861680806356	93.3025730334164	97.11627863610413
HighwayI	80.34138773413603	81.74215721899223	84.80438587060391

FAILURE CASES:

1. **Not able to remove the shadow in some cases:**
   Our model of Background Subtraction method is not able to detect the shadow, because shadows are also creating the changes in pixel intensity



2. **In Caviar2 Dataset, detecting the person partially:**
   After noise removal from the images of the CAVIAR1 dataset, my model in creating the output where it is speculating at that particular spot, the pixel intensity is not changing.

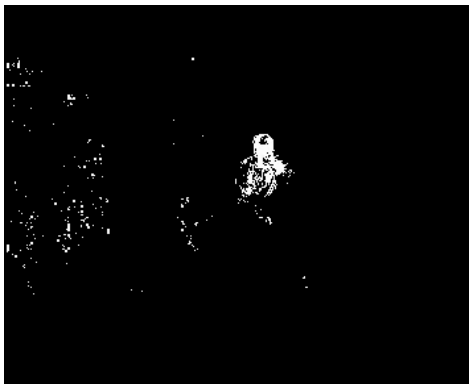


3. **In the Candela dataset, the person is getting disappeared:**

In the Candela dataset, the person is getting disappeared, when the person is sitting on the couch, my model is giving this failure result may be because of the high threshold, and considering a very low change in the pixel intensity.

4. **Noise cleaning:**

   To remove the noise present in the frames, I have implemented the method of integral images, where I have divided the frame in a grid of 3*3 matrix, and detected the noise by considering if less than the half of pixel value is white then, it is a noise and removed it by making them black. But due to trees and ambience changes, it is not able to detect that.



References:

1. Non-parametric Model for Background Subtraction by Ahmed Elgammal, David Harwood, Larry Davis
2. Adaptive background mixture models for real-time tracking by Chris Stauffer W.E.L Grimson
3. Computer Vision: Algorithms and Applications
   Book by Richard Szeliski

OneDrive link of code and videos:

https://csciitd-my.sharepoint.com/personal/mcs222070_iitd_ac_in/_layouts/15/onedrive.aspx?id=%2Fpersonal%2Fmcs222070%5Fiitd%5Fac%5Fin%2FDocuments%2FCOL780%20Assignment1&ga=1