COL780 IIT Delhi

Background Subtraction

Done By-
Rahul Parmar (2019CS10387)
Navneet Jindal (2019CS10379)

Introduction

Background Subtraction is a well-known technique for extracting foreground masks from the video's background, and it's the basis for this project. This work is completed for various scenarios, including straightforward films, lighting changes that can introduce shadows and modify brightness, videos shot with a shaking camera (jitter), videos with a dynamic background, and Pan-tilt zoom videos. In the sections below, we discuss our methodology and approaches for background subtraction in various contexts. We have implemented the same using OpenCV and python.

Part 1:- Baseline

OpenCV's KNN background subtractor is used to get the foreground mask of different frames in the video. To correctly identify the background, it uses the K-Nearest Neighbour approach to calculate the Euclidean distance between each segment of the current video frame and the history frames. It uses some of the histories to build a statistical model for the image's background before classification each subsequent segmented frame. After applying the KNN model to the given image, we observed the result and improved our results using different methods.

In the given image, it was observed that there was some leaf movement in the background trees. So this case was quite similar to part 4 i.e., moving background. We applied median blurring/filtering to remove this leaf movement as the leaves movement (background movement) would be pretty small and discrete, which is easily removed by median blurring/filtering.

We also observed that the KNN model detects a lot of noisy points and doesn't detect the whole object. To remove this noise we applied erosion and median filtering. To fill up our objects whole we used contour filling and erosion and dilation to remove any noise detected and make detected objects whole.

The above techniques improved our IOU value from about 0.67 to 0.79.

Apart from this, we also tried Temporal median filtering in which we took a median of some initial frames (Not used for evaluation) as preprocessing. Then used this median frame as background and performed background subtraction (absolute difference) for subsequent frames to get the foreground mask. This also provided us with about 0.77 IOU.

Best IOU: 0.7933
Output Link:: [Baseline]

Part 2:- Illumination

Next, we add the feature to handle illumination changes in our video. Here we use the MOG2 background subtraction to get our result instead of KNN Background subtraction as used in Baseline. Because illumination changes alter the Euclidean distance between the learned background model and the frame segments, KNN does not operate well with video illumination variations. Even if other features stay unchanged, if the lighting changes in a region, its Euclidean distance from the previously learned model explodes. It is recognized in the foreground mask, which is incorrect. MOG2 subtractor works well in illumination changes.

This still did not give us a satisfactory result. Hence, we also applied Contrast Limited Adaptive Histogram Equalization (CLAHE) to improve the image's contrast and reduce the effects of variation in illumination across frames. This worked well with the given dataset. We also tried histogram equalization, but it didn't work well as it assumes a similar distribution in the whole frame. We did adaptive histogram equalization to improve this, which divides the frame into multiple segments and does histogram equalization individually. This introduces noise in the output. CLAHE handles this noise by putting a limit on contrast amplification.

After applying CLAHE we applied the MOG2 on the resulting output (similar to Baseline) and achieved an IOU value of 0.532. We see that this is still a low value which is mainly because of the low white region to be detected i.e. small persons moving across in frames. As the total region to detect is low, our IOU value decreases a lot because of minor errors.

Best IOU achieved: 0.532
Output Link:: [illumination]

Part 3:- Jitter

This is a fascinating case. Here the camera is randomly moving due to its unstable setup. When we apply a simple KNN model on a jitter video, then sharp objects or lines are also caught by the KNN model. These are removed by erosion and median filtering. Still, with a large camera shake, such lines acquired are thicker, making it challenging to classify between the foreground object and the thick line. So to remove these acquired lines, we tried to remove/reduce the jitter in-camera by applying the image alignment technique.

We first find the median of some first N frames to get the base background frame to which each frame is aligned. Next, we align every frame to this median image to get a constant orientation of the camera. We apply the KNN Background subtraction model on this correctly orientated image to get the foreground mask to an IOU value of 0.69.

We also tried averaging some consecutive k frames as we do in the sliding window method. Doing so blurs the image and also removes the jitter. Next, we apply the KNN Background subtraction model on these averaged frames. Due to the blurring of the image acquired, IOU w.r.t given ground truth value is low. But we were able to distinguish between foreground and background clearly.

Applying simple KNN like in Baseline with some post-processing modifications gives the best result with IOU value of 0.71. This might be because that removing jitter doesn't improve our IOU value a lot as with the given jitter, our KNN model is not affected a lot by it (KNN model is an excellent robust algorithm), and aligning an image still reduces our image quality, making the detection non-accurate in the image alignment case. Also, there is sometimes blurring of image when the camera shaking is significant, making image alignment less accurate. We tried sharpening the image to reduce this effect, but it still did not increase our IOU value.

Best IOU achieved: 0.71
Output Link: Jitter (IOU value achieved in this is 0.68)

Part 4:- Moving BG

In this case, there are also some slight background movements. So applying simple KNN or MOG2 also detects these background movements. Here we assume that the background movements are random and small. So these can be treated as noise in the image. This results in discrete and small background movement detections.

Here we first apply a gaussian blur so that background movement being small is blurred and escapes detection. This will decrease background movement detection. Also, the background movement detection (due to water or leaves, etc.) will be pretty discrete as there will be similar movements in the spatial neighborhood. So applying a median blur on the detected KNN model will remove most of the background movement detected. The rest of the movement detected is the required foreground mask which is refined using erosion and dilation. Next, we also tried to do this using dense optical flow and thresholding a particular optical flow value to distinguish foreground and background movements. This creates a problem when foreground objects also move slowly.

Best IOU Achieved: 0.5023
Output Link: Moving BG

IOU: 0.5023
IOU: 0.7952 (considering empty frame doesn't contribute to IOU)

Part 5 :- PTZ

In PTZ cameras, there are camera movements and even camera zoom in and out variations. Capturing movements/foreground masks in this type of problem is a lot tougher. What our start idea was to do feature matching and transform the current frame the to previous frame using homography to minimize the changes due to PTZ movements, and then apply KNN model on the transformed frames. But this resulted in a very low IOU value as KNN model takes a set of previous frames to do background subtraction.

To handle this history of transformation we transformed each 10 frames to a specific frame so that we can handle the KNN model. So we have divided all frames in a set of 10 consecutive frames. Now we align each frame in a set to the first frame in the set to get a consistent background orientation for at least 10 frames. We can't have many frames in a set as we also need to update the base image frame on which the whole set is aligned to. Now we apply the KNN model on the aligned frames. This gives a somewhat improved IOU value but still there can be more improvement. Here the issue which arises is that when we suddenly switch the set of consecutive frames the KNN model detects large changes and gives incorrect values for initial set frames and gradually gives the correct result till we reach the end of the set. This means we get the wrong result approx from frame 1 to 5 (the first half of the set) and correct result from frame 6 to 10 (the second half of the set).

To reduce this loss, we apply two background subtractors such that one makes a set from frame i to i+10 and the other makes a set from i+5 to i+15. So we can take the best half of both these background subtractors to get our correct result i.e., we get foreground mask from background subtractor 2 for frame i to i+5 and get foreground mask from background subtractor 1 for frame i+5 to i+10. Modifying these values (size of set and number of background subtractors used) can further increase our IOU value.

Output Link: Pan Tilt Zoom

IOU: 0.1892
IOU: 0.3796 (considering empty frame doesn't contribute to IOU)