COL780 Assignment 1

Tamajit Banerjee | Mustafa Chasmai
2019CS10408 | 2019CS10341

September 2021

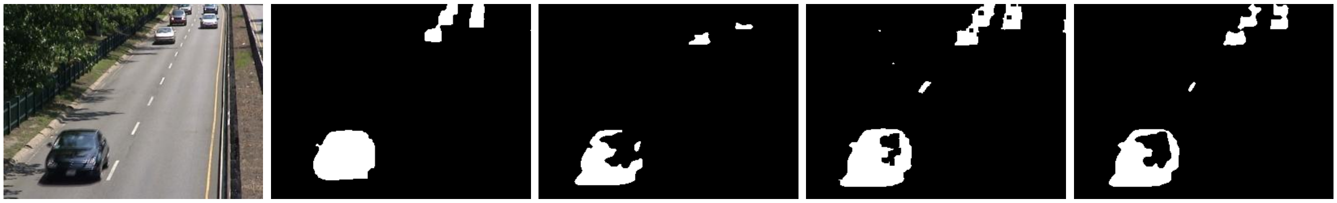
1 Baseline

As a baseline for background subtraction under static background assumptions, we tried the following methods:

1.1 Median and Hysteresis Thresholding (MHT)

Since the baseline contains videos with a static background an no illumination changes, the background model can be constructed simply as the previous frame. To make the model more robust against noise, an average of the last k (tunable parameter) frames was used as the background. In practice, we observed that the median served as a better alternative to average, and thus, in this method, we consider the background of an image to be the median of the last k images.

Once we got the modelled background image, we first applied simple thresholding to the difference of the image from background. Upon fine tuning the threshold, we observed that a high threshold would lead to a lot of foreground being missed, while a lower threshold would lead to erroneous noise. In the provided validation images in particular, the leaves of trees beside the road moved between frames, and contributed to significant noise. Taking a closer look, we observed that the leaves had much lower differences than the cars, and decided to experiment with hysteresis thresholding. Thus, all pixels above a particular threshold were marked as surely foregorund, all below a different threshold were marked as surely background and a pixel between the two thresholds was marked as foreground only if it had atleast one pixel that is surely foreground in a 3 × 3 widow around it.



(a) Input Image (b) Ground Truth (c) High Threshold (d) Low Threshold (e) Hysteresis

Figure 1: Effects of thresholding

1.2 Entropy-Based initial Background Extraction (EBBE)

The algorithm was published in the paper 'A Background Subtraction Algorithm in Complex Environments Based on Category Entropy Analysis' [1] in 2018. It is a background subtraction algorithm based on category entropy analysis that dynamically creates color categories for each pixel in the images. The algorithm uses the concept of a joint category to build background categories that can adapt to the color disturbance of the background. Furthermore, in order to overcome dynamic background environments, the algorithm uses the concept of color category entropy to estimate the number of necessary background categories and establish sufficient and representative background categories to adapt to dynamic background environments.

We implemented the algorithm. But as we started running them on the datasets although the background started to converge in the right direction but it did not scale well. I think it was mainly due to the fact that the parameters were not well tuned and also there was very little information about preprocessing and post processing of the images.

We also added several new heuristics in the above algorithm. Since the background can change , we tried to keep sufficient number of color categories for each pixel and periodically removed some of the color categories to reduce the time complexity of the algorithm.

1.3 Mixture Of Gaussians (MOG)

It is a Gaussian Mixture-based Background/Foreground Segmentation Algorithm. It was introduced in the paper 'An improved adaptive background mixture model for real-time tracking with shadow detection' [3] by P. KadewTraKuPong and R. Bowden in 2001. It uses a method to model each background pixel by a mixture of K Gaussian distributions (K = 3 to 5). The weights of the mixture represent the time proportions that those colours stay in the scene. The probable background colours are the ones which stay longer and more static.

Initially, we need to create a background object using the function, cv2.createBackgroundSubtractorMOG(). It has some optional parameters like length of history, number of gaussian mixtures, threshold etc. It is all set to some default values. Then we have to use backgroundsubtractor.apply() method to get the foreground mask.

The Parameters we tuned are :

1. Length of the history
2. Threshold on the squared Mahalanobis distance between the pixel and the model to decide whether a pixel is well described by the background model. This parameter does not affect the background update.

1.4 K Nearest Neighbors (KNN)

It implements the K-nearest neigbours background subtraction described in the paper 'Efficient adaptive density estimation per image pixel for the task of background subtraction' [4]. It is very efficient if number of foreground pixels is low. The paper consisted of recursive equations that are used to constantly update the parameters of a Gaussian mixture model and to simultaneously select the appropriate number of components for each pixel and it also presented a simple non-parametric adaptive density estimation method.

Initially, we need to create a background object using the function, cv2.createBackgroundSubtractorKNN(). Then we have to use backgroundsubtractor.apply() method to get the foreground mask.

The Parameters we tuned are :

1. Length of the history
2. Threshold on the squared distance between the pixel and the sample to decide whether a pixel is close to that sample. This parameter does not affect the background update

1.5 Other image processing techniques used in Baseline

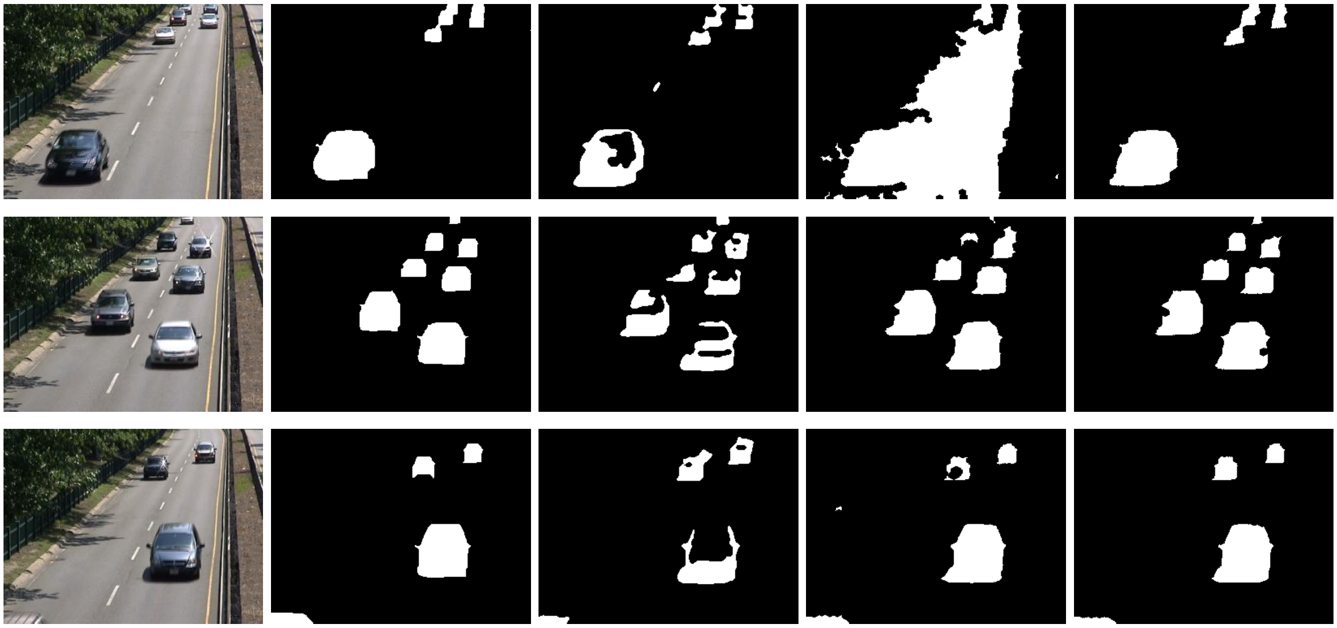
1. In the Baseline final processing, we also used the Dilation and Erosion functions for processing the image. We used a general kernel which is MORPH_ELLIPSE of 5 × 5 size whose matrix representation is:

$$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

2. We also used Contour detection, and removed objects whose area was less than some threshold. This allowed us to filter out some random noise, since most of the noise was in the form of small isolated patches.

1.6 Performance on Validation dataset

Applying our baseline methods which we explained above , the best results were given by KNN and post processing the mask with dilation, erosion and contour detection. We were able to obtain a mean IOU of **0.8035**. A qualitative analysis of some sample images of the validation data can be seen in Fig 2. The link to the output masks for baseline is baseline results



(a) Input Image (b) Ground Truth (c) MHT (d) MoG (e) KNN (baseline)

Figure 2: Qualitative comparison of different methods for baseline on some sample images

2 Illumination

Changes in the illumination pose quite a few challenges. The most common approach to image based background subtraction systems is to make use of the difference in intensity of a particular pixel to classify it as being foreground or background. Now with drastically changing illumination, the intensities of true background pixels would also be changing significantly, leading them to be mis-classified as foreground. We observed this phenomenon when we used our illumination method on the illumination validation data provided.

To remedy these effects of large variations in illumination, we used histogram equalisation and obtained more stable grayscale images. On these equalised imaged, applying the illumination model with some fine-tuning allowed us to improve the performance of our model from **12.86% to 55%**. Histogram equalisation is explained in short in Section 2.1. A clear validation of the usefullness of histogram equalisation can be seen in Fig 3.

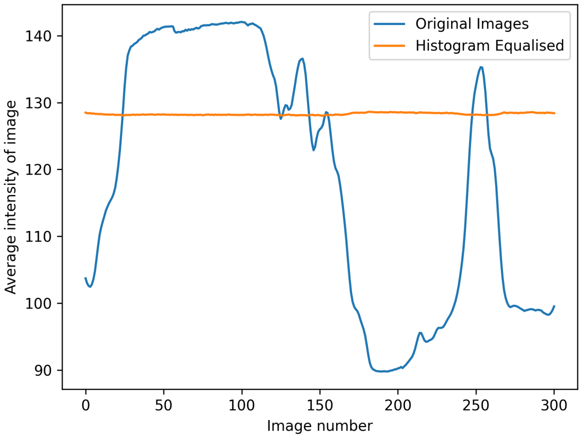


Figure 3: Variation of average intensities of the validation images, before and after histogram equalisation.

2.1 Histogram Equalisation

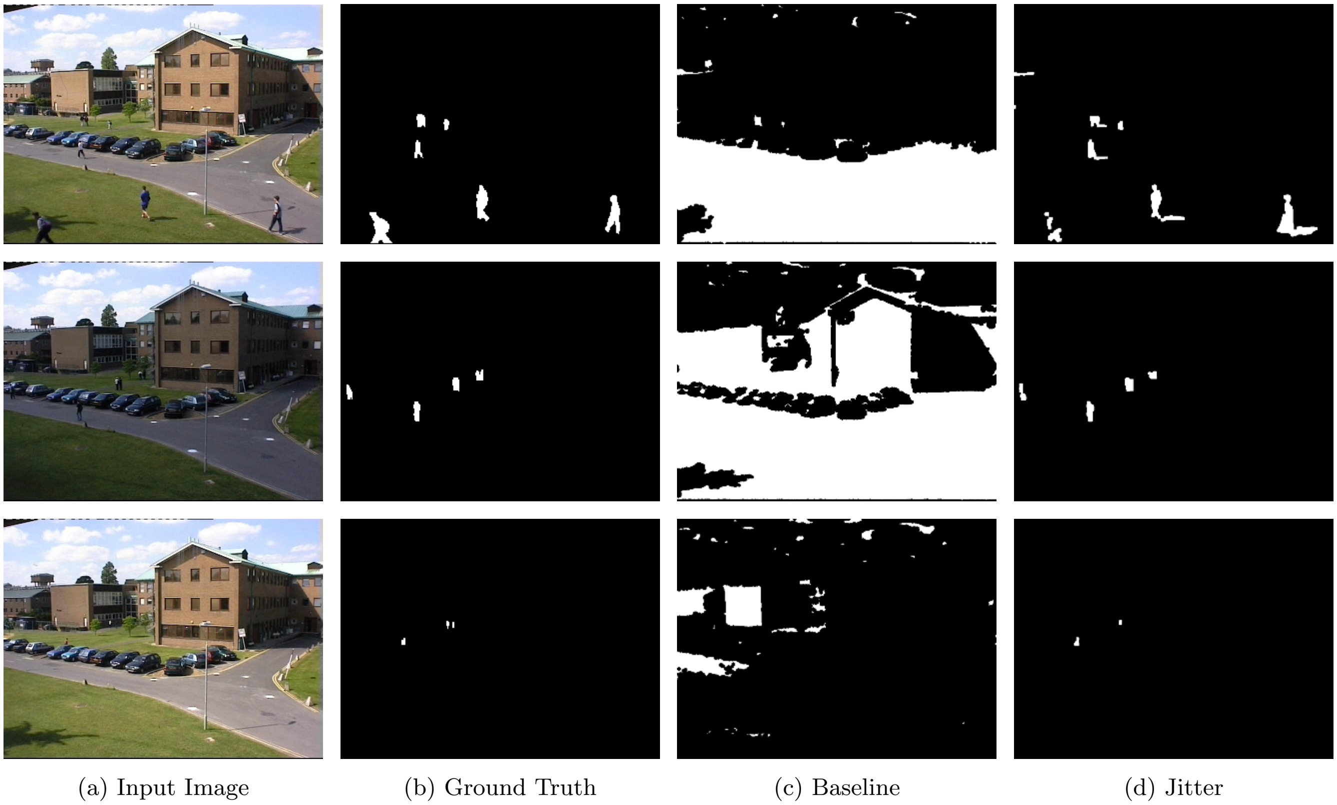
Histogram equalization is a simple image processing method of contrast adjustment using the image's histogram. Through this method, the intensities can be better distributed on the histogram utilizing the full range of intensities evenly. Often implemented using a Lookup Table, histogram equalisation is a widely used method, and OpenCV has an easy to use implementation of it.

2.2 Changes from Baseline

1. Histogram Equalisation
   (a) Instead of RGB images used in the illumination, grayscale images were used here.
   (b) The grayscale images were passed through a histogram equaliser
   (c) OpenCV's cv2.equalizeHist() function was used for the same
2. Reduction in History of KNN model
   (a) History is the number of frames previously seen by the model that are used to construct the background.
   (b) Decreasing history made the model more robust to illumination changes in the background, decreasing the false positive noise.
3. Hyper parameter tuning (specific to data)
   (a) The illumination data contained images with larger foreground objects compared to the illumination data. With the large contour threshold used in the illumination, many humans that were far from the camera were missed. Thus, a smaller contour threshold was used here.
   (b) The number of iterations for dilation and erosion were tuned slightly to obtain better results.

2.3 Performance on Validation dataset

Applying our baseline method to the data having illumination changes, we were able to obtain a mean IOU of 0.1286. With histogram equalisation and shorter history, we obtained a mean IOU of 0.5339, and after some more fine-tuning, we obtained a mean IOU of **0.5533**. A qualitative analysis of some sample images of the validation data can be seen in Fig 4. The link to the output masks for illumination is illumination results



(a) Input Image (b) Ground Truth (c) Baseline (d) Jitter

Figure 4: Qualitative comparison of different methods for jitter on some sample images

2.4 Discussion

As can be seen in Fig 8c and also in the IOU score of 0.1286, the baseline method performs quite poorly on this data. Since the changes in illumination are quite significant (Fig 3), the baseline model has a hard time differentiating backgrounds with different illumination from moving foreground. This is particularly prominent when there is a sudden change in the illumination, as present in the validation data. By redistributing the intensities, we can obtain a much smoother variation of intensities, and a much better performance (mIOU 0.5533). With small fine-tuning of parameters, our method should be able to generalise well to other datasets having varying illumination.

3 Jitter

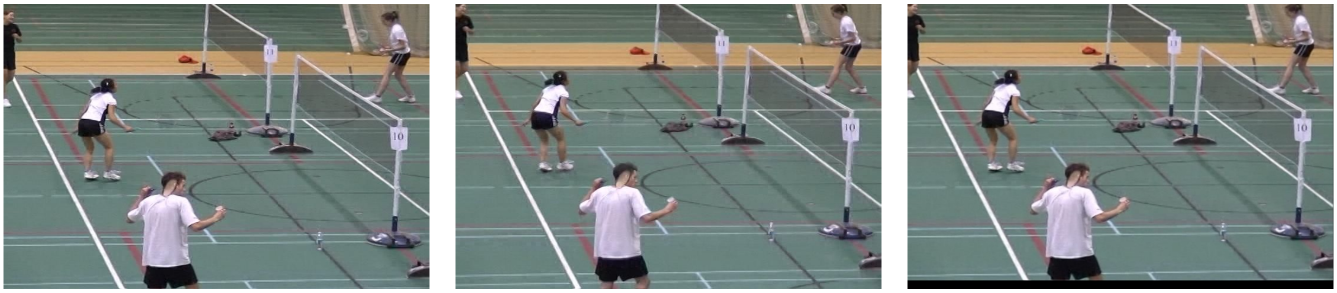
Camera Shake or jitter is indeed a challenge for background subtraction systems. It is non trivial to differentiate between the movement of objects in the image and the movement of the camera itself. A standard background subtractor with static camera assumptions will flag the perceived movement of features in the background as the same as the movement of foreground objects of interest. We observed this in the validation dataset provided, where the baseline model was detecting lines in the tennis court as foreground.

To handle camera shake more robustly, we first try to 'align' every image with its corresponding background image. We then update the background model with this aligned image. This aligning step is basically finding a transformation between the two images. Since we only need to consider camera shake, we assume simple geometric transformations, and in particular, only translation. Thus, we estimate a translation matrix to transform from an image to the background, then apply the background model, and then obtain the final mask by applying a reverse transformation to the predicted mask. This makes our model more robust to jitter. The method to estimate this translation matrix for alignment is described in Section 3.1 below.

3.1 Enhanced Correlation Coefficient Maximization (ECC) [2]

This method effectively estimates the geometric transformation between an input and template image. Given a particular transformation model (translation, affine etc), it tries to estimate the transformation parameters to best fit the two images. Although most other algorithms attempt to compute the parameters by minimizing the difference or the dissimilarity of the two images, this method devises a new enhanced correlation coefficient to measure the performance of their model. With this measure, they use either search based or gradient based methods to obtain optimal sets of parameters, maximising the correlation. Once the optimal parameters are obtained, they can simply be plugged into the transformation model, and a transformation matrix can be obtained.

OpenCV has an easy to use, openly available implementation of this ECC method. The cv2.findTransformECC() function allowed us to estimate the transformation matrix. The function accepts parameters that can be tuned for optimal performance. The most relevant parameters are: 1) transformation model (translation, euclidean, affine and homography) and 2) termination criteria (setting stricter criteria gave better results, but took longer to compute). In our case of jitter, we used a translation model and obtained the translation matrix for aligning a current image with the background image estimated by our background model. With the translation matrix, we warped the current image, then applied the background model on it, and finally warped the obtain masked again with the inverse transformation. An example of the forward transform can be seen in Fig 5.



(a) Input Image

(b) Refernce Image

(c) Aligned Input Image

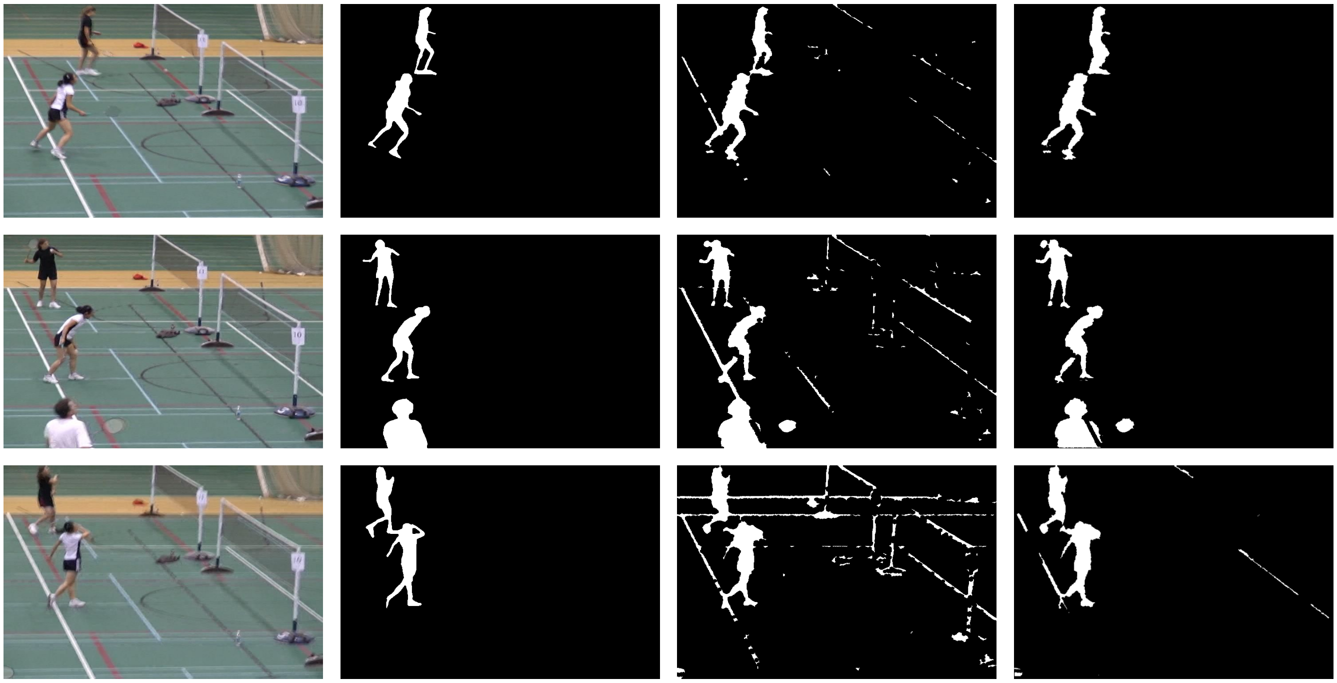
Figure 5: Image Alignment Using ECC algorithm

3.2 Changes from Baseline

1. Image Alignment
   (a) A transformation matrix with a translation model is obtained using ECC
   (b) Instead of the raw image, a translated image is applied to the background model
   (c) After obtaining the mask from the background model, the mask is again transformed back to the original using an inverse transformation.
2. Hyper parameter tuning (specific to data)
   (a) The threshold used by the background model was tuned to obtain better results
   (b) The number of iterations for dilation and erosion were tuned slightly to obtain better results.

3.3 Performance on Validation dataset

Applying our baseline method to the validation data having jitter, we were able to obtain a mean IOU of 0.6597. With ECC alignment, we obtained a mean IOU of 0.72, and after some more fine-tuning, we obtained a mean IOU of **0.7609**. A qualitative analysis of some sample images of the validation data can be seen in Fig 6. The link to the output masks for jitter is jitter results



(a) Input Image (b) Ground Truth (c) Baseline (d) Using ECC alignment

Figure 6: Qualitative comparison of different methods for jitter on some sample images

3.4 Discussion

From the IOU score of 0.6597, it can be seen that the baseline method performs reasonably well on this data. However, as can be seen in Fig 6c, the predicted masks tend to have a lot of noise. Since in this particular dataset that noise is in the form of thin lines, it does not lead to a significant deacrease in the IOU. However, the noise is clearly visible upon analytical inspection, and so, the baseline model has scope for improvement. Contrasting with the masks obtained using ECC alignment, one can see that there is significant improvement. By explicitly modelling shaking of the camera using a translation matrix, the method becomes robust against camera shake and jitter, obtaining a mean IOU of **0.7609**. With small fine-tuning of parameters, our method should be able to generalise well to other datasets having varying illumination.

4 Moving Backgrounds

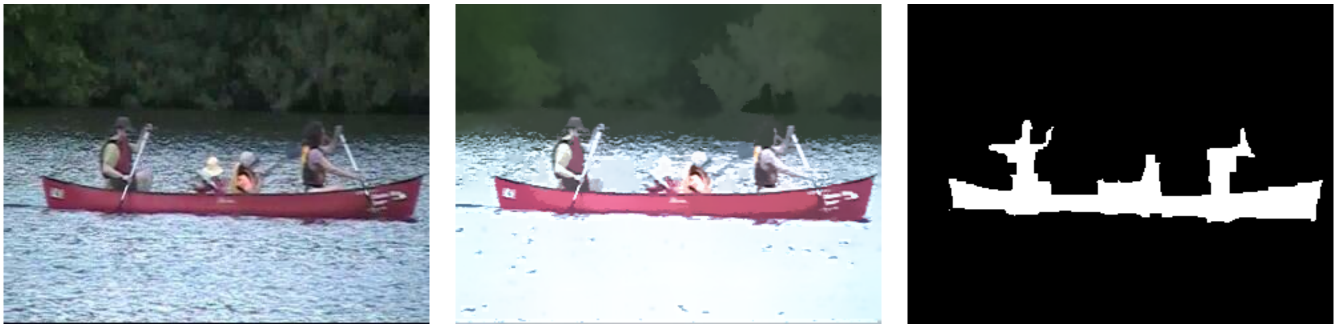
Background subtraction in the case of dynamically changing background is very difficult and it is difficult to detect if the changes in the background is due to a foreground object or a background object. A background subtractor with static background assumption will go detect changes in the background as a foreground object.

To handle the background changes more robustly, we devised two ways one is to increase the contrast of the image by multiply each pixel value by a constant value and second is to apply pyrMeanShiftFiltering() which is Pyramidal Mean Shift Filtering and it performs the initial step of meanshift segmentation of an image. The image contrast was increased so that some close features in the foreground can be distinguished from the background and then the Pyramidal Mean Shift Filtering to club all the similar value pixels in the neighbourhood together.

An example of the image processing done by the above two methods can be seen in Fig 7 .

4.1 Changes from Baseline

1. Increasing Image Contrast
   (a) Multiplied each pixel by a constant factor and then reduce each of the pixel's value to 255 if it was above 255.
2. Pyramidal Mean Shift Filtering
   (a) The function implements the filtering stage of meanshift segmentation, that is, the output of the function is the filtered "posterized" image with color gradients and fine-grain texture flattened.
   (b) The parameters tuned were :
      i. The spatial window radius (similar to sigma of gaussian filtering for the neighbouring pixels).
      ii. The color window radius (similar to sigma of gaussian filtering for the neighbouring pixel values)
      iii. Maximum level of the pyramid for the segmentation.
3. Hyper parameter tuning (specific to data)
   (a) The number of iterations for dilation and erosion were tuned slightly to obtain better results.



(a) Input Image (b) Processed Image (c) Output Mask

Figure 7: Image Processing

4.2 Performance on Validation dataset

Applying our baseline method to the data having illumination changes, we were able to obtain a mean IOU of 0.441. With contrast enhancement and pyramidal mean shift filter, we obtained a mean IOU of **0.515**. A qualitative analysis of some sample images of the validation data can be seen in Fig 8. The link to the output masks for moving background is moving_bg results

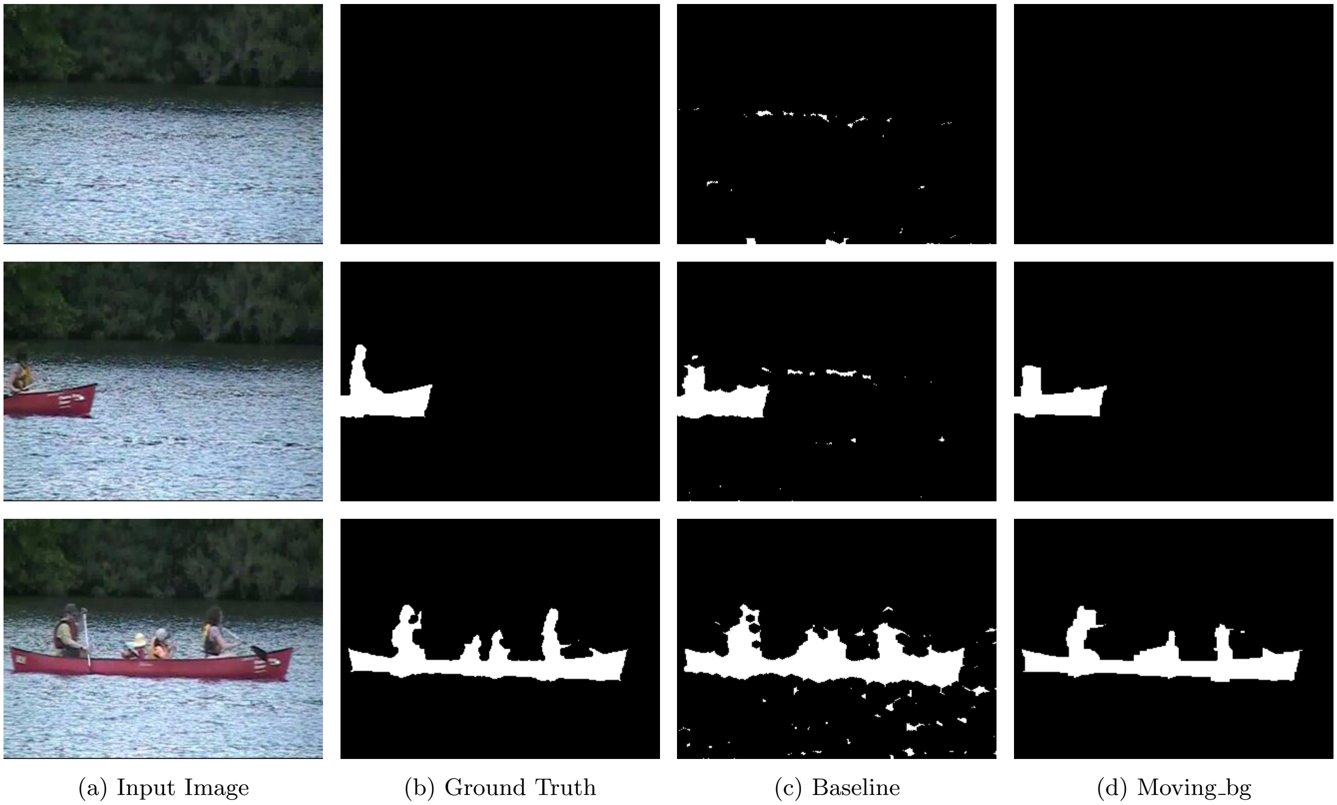


Figure 8: Qualitative comparison of different methods for moving backgrounds on some sample images

4.3 Discussion

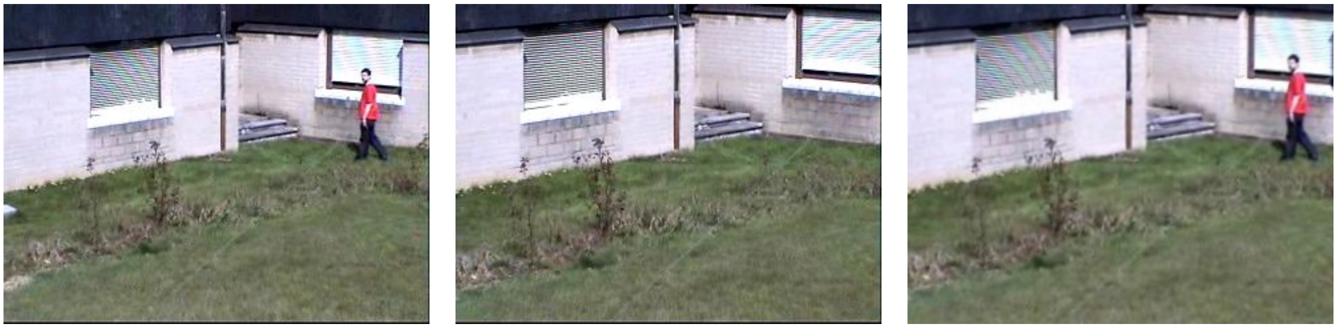
As can be seen in Fig 8 and also in the IOU score of 0.441, the baseline method does not perform good in this data. Since the changes in background gets detected as foreground changes and this decreases the IOU score a lot. The baseline model has a hard time differentiating backgrounds from moving foreground. This is particularly prominent when there is a sudden change in the background due to the waves, as present in the validation data. By increasing the contrast and using the pyramidal mean shift filter, we can obtain a much better performance (IOU 0.515). With small fine-tuning of parameters, our method should be able to generalise well to other datasets having changing background.

5 Pan-tilt-Zoom

Pan–tilt–zoom (PTZ) cameras are capable of remote directional and zoom control allowing them to dynamically modify their field of view. They have many practical applications, particularly in surveillance. They allow increasing the resolution of moving targets and adapting the sensor coverage, thus enabling to focus the attention on automatically selected areas of interest. Thus, background subtraction systems performing well with these kinds of cameras are quite relevant.

The PTZ camera has two axes of rotation (pan and tilt), along with scaling. As the camera makes these movements, in the captured image, it is perceived as if the background itself is moving. Thus, differentiating between the moving foreground of interest becomes quite challenging. This scenario can be considered to be very similar to that of jitter, but with a more complex camera movement. Thus, we solve the problem in a very similar fashion, with a more complex movement model.

With two axes of rotation, the PTZ camera can rotate to any orientation in 3D space. Since OpenCV motion models do not support this transformation, we use two different motion models and take their combination (composition or cascading of the transformations). Since we also need to model zoom, we combine one of the transformation with scaling. Thus, in effect, out of the available models, we use a combination of one **Euclidean** (translation + rotation) and one **Affine** (translation + rotation + scale + shear) transformations. Since Euclidean is a weaker model, we use this model to obtain a transformation between the image and the background. Now on this transformed image, we again obtain a transformation using the Affine model. We apply the background on this image and then do inverse transformations (first affine and then euclidean) to get the final mask. An example can be seen in Fig 9.



(a) Input Image

(b) Refernce Image

(c) Aligned Input Image

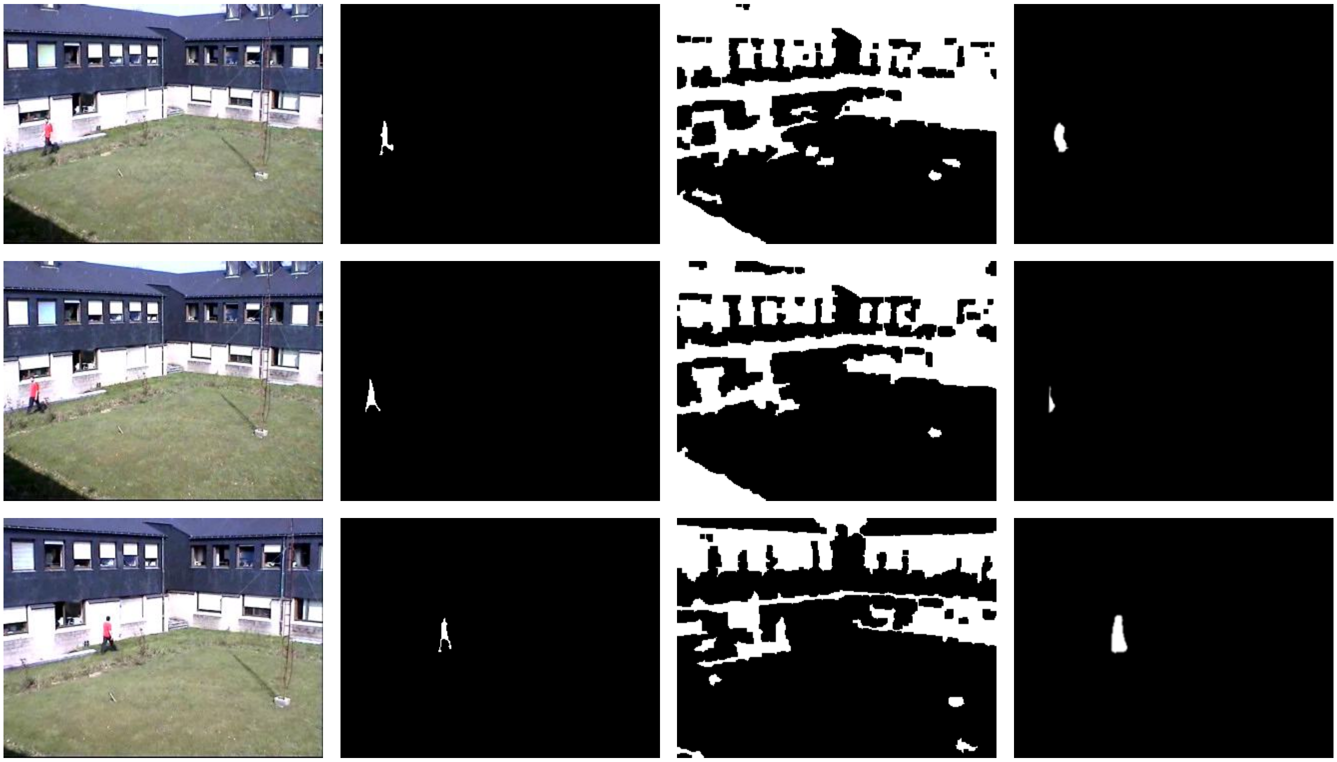
Figure 9: Image Alignment with 2 transformation using ECC algorithm on sample images. Note in the aligned image how the person from zoomed out input image is preserved and visible at the same scale as reference image.

5.1 Changes from Baseline

1. Image Alignment
   (a) A transformation matrix with a Euclidean model is obtained using ECC
   (b) Another transformation matrix with an Affine model is obtained using ECC on the image obtained from the previous step, aligning again with the background
   (c) After obtaining the mask from the background model, the mask is again transformed back to the original using two inverse transformations in the reverse order.
2. Hyper parameter tuning (specific to data)
   (a) The threshold used by the background model was tuned to obtain better results
   (b) The number of iterations for dilation and erosion were tuned slightly to obtain better results.

5.2 Performance on Validation dataset

Applying our baseline method to the validation data having jitter, we were able to obtain a mean IOU of 0.0276. With ECC alignment, we obtained a mean IOU of **0.3231**, with a significant boost in performance. A qualitative analysis of some sample images of the validation data can be seen in Fig 10. Pls note that the mIOU score reported above was after evaluating on a subset of the entire dataset. The evaluation frames taken were: 500 to 815 instead of the given 500 to 1130. The mIOU on the exact given evaluation frames was 0.1617. The reason the frames 815 to 1130 were removed were: 1) frames 815 to 1000 have incorrect ground truths (input image has human, but ground truth empty); and 2) frames 1000 to 1130 have Moire patterns in the background, which lead to significant noise. Since Moire patterns or aliasing is particular to the dataset, and not a general problem in PTZ cameras, we chose to not work on it, and provide evaluations without these images as a better indication of the method. The link to the output masks for ptz camera is full results



(a) Input Image (b) Ground Truth (c) Baseline (d) Using ECC alignment

Figure 10: Qualitative comparison of different methods for PTZ camera on some sample images

5.3 Discussion

The performance of the baseline model on PTZ validation data is very poor (0.0276 mIOU). In almost all frames there is very significant changes in the background and the baseline model is not able to handle these effectively. The performance boost obtained by using ECC and aligning is very significant (more than 15 times).

6 Conclusion

We observe, that the Baseline code gave good performance in the case where the background was static, and there was no change in illumination conditions between the video frames.For different cases, we had to add different algorithms or processes in our code to overcome the challenges.

We found out that changing some of the hyper parameter improves our mIOU values. This hyper parameter will mainly depend on the kind of foreground objects we want to extract. For example, if we are extracting a foreground object whose size is big we can detect contours of small sizes and remove them whereas if our final object is small in size we have to be careful with contour detection and also try to dilate the image to detect the objects better.

Irrespective of the hyper parameters used, we believe that the methods used are robust in their corresponding scene conditions and thus can be generalised to different datasets with the same scene conditions.
A short summary of the different scene conditions and corresponding methods used is presented in the table below.

Scene Conditions	Baseline mIOU	Method mIOU	Key points
Baseline	N.A.	0.8035	KNN, Dilation, Erosion, Contour Detection
Illumination	0.1286	0.5533	Histogram Equalisation, Reduction in History of KNN model
Jitter	0.6597	0.7607	Image Alignment using ECC (Translation)
Moving Background	0.441	0.5147	Increasing Image Contrast, Pyramidal Mean Shift Filtering
Pan-tilt-Zoom	0.0276	0.3229	Image Alignment using ECC (both Translation and Rotation)

References

[1] Sheng-Yi Chiu, Chung-Cheng Chiu, and Sendren Sheng-Dong Xu. “A Background Subtraction Algorithm in Complex Environments Based on Category Entropy Analysis”. In: *Applied sciences* 8.6 (2018), p. 885.

[2] Georgios D Evangelidis and Emmanouil Z Psarakis. “Parametric image alignment using enhanced correlation coefficient maximization”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30.10 (2008), pp. 1858–1865.

[3] Pakorn KaewTraKulPong and Richard Bowden. “An improved adaptive background mixture model for real-time tracking with shadow detection”. In: *Video-based surveillance systems*. Springer, 2002, pp. 135–144.

[4] Zoran Zivkovic and Ferdinand Van Der Heijden. “Efficient adaptive density estimation per image pixel for the task of background subtraction”. In: *Pattern recognition letters* 27.7 (2006), pp. 773–780.