Assignment 1

Background Subtraction

Hari Krishnan CK

2021EEY7583

Method – Kernel Density based background subtraction with constant weight (Non Parametric background subtraction)

The main objective of the assignment is to model the background subtraction model which separates the foreground from the background using a non-parametric kernel density approach. Given $x_1, x_2, \ldots x_{N-1}$ previous values of pixel from previous frames we need to subtract the background from the present frame (i.e x_t). The probability that the present pixel value actually belongs to the background is given by

$$P_r(x_t) = \frac{1}{n}\sum_{i=1}^{N} K(x_t - x_i)$$

If K is chosen to be a normal distribution, then,

$$P_r(x_t) = \frac{1}{N}\sum_{i=1}^{N} \frac{1}{(2\pi)^{d/2}\,|\sum|^{1/2}} e^{-\frac{1}{2}(x_t - xi)^T \sum^{-1}(x_t - x_i)}$$

So the present pixel is assigned to the background if the probability value is above a given threshold. This threshold has to be chosen for a particular task and has to be tuned.

The variance is obtained by calculating the median of $|x_i - x_{i+1}|$ from all the previous values say m. It is also assumed that the covariance matrix is a diagonal matrix i.e there is no covariance between the channels. Then the standard deviation for a channel is obtained by

$$\sigma = \frac{m}{0.68\sqrt{2}}$$

False Detection Suppression

The above algorithm could result in a lot of noise due to small movements in the background and hence a false detection. We can reduce this by doing a neighbourhood check also on the pixel. Using the same probability density equation we evaluate the maximum probability for the neighbourhood pixels. The assumption is that small changes to movement of pixels due to situations like leaves moving in the wind, etc could be reduced by checking the probability for the neighbourhood pixels

$$P_N(x_t) = \max_{y \in N(x)} P_r(x_t \mid B_y)$$

Here N refers to the neighbourhood of the pixel x and B_y is the sample pixel in the background.

But the main problem with detection suppression is that foreground could also be subtracted, so we do the suppression only if the current pixel is not part of a big connected component. So we make sure that the entire target component has moved from nearby not only some of the pixels. i.e.

$$P_C = \prod_{x \in C} P_{N(x)}$$

and assign pixel to the background only if

$$(P_N(x) > th_1) \wedge (P_C(x) > th_2)$$

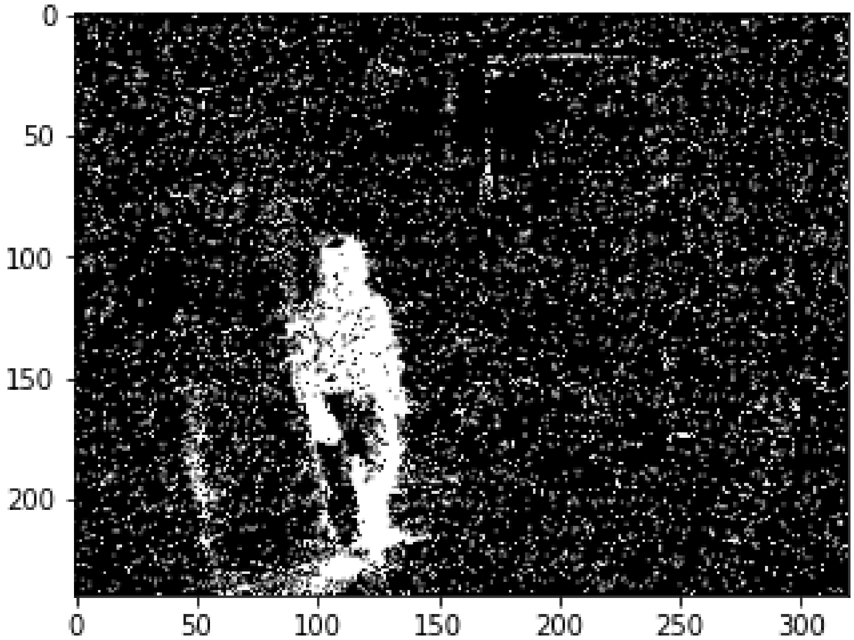

Figure 1: Image without applying suppression of false detection

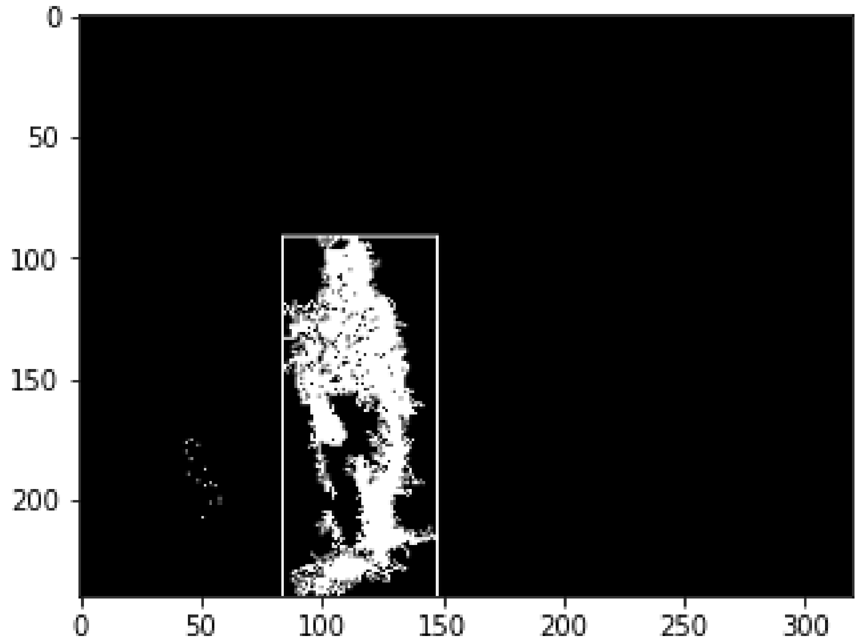

Figure 2: Image after applying suppression to reduce false detection.

Limitations of Kernel based background Subtraction

1.) Sensitive: Although the false detection is suppressed still there exists a lot of sensitivity associated with the model due to which the edges of objects of non moving objects could also come into the picture. These are introduced due to tiny movements in the camera. If any new object comes into the frame there could be lot of noise also associated with it.
Solution: The solution is to optimally tune the threshold th_1 associated with the false detection accurately so that we will be able to detect the movements and remove from the foreground.
2.) Inside of the moving objects: Inside of big objects like car may not be able to be detected since these pixels may not move too much resulting in these pixels being assigned to the background. This does not happen in the parametric model.
Solution: Increasing the number of frames taken into consideration can solve this problem but will increase the computational time

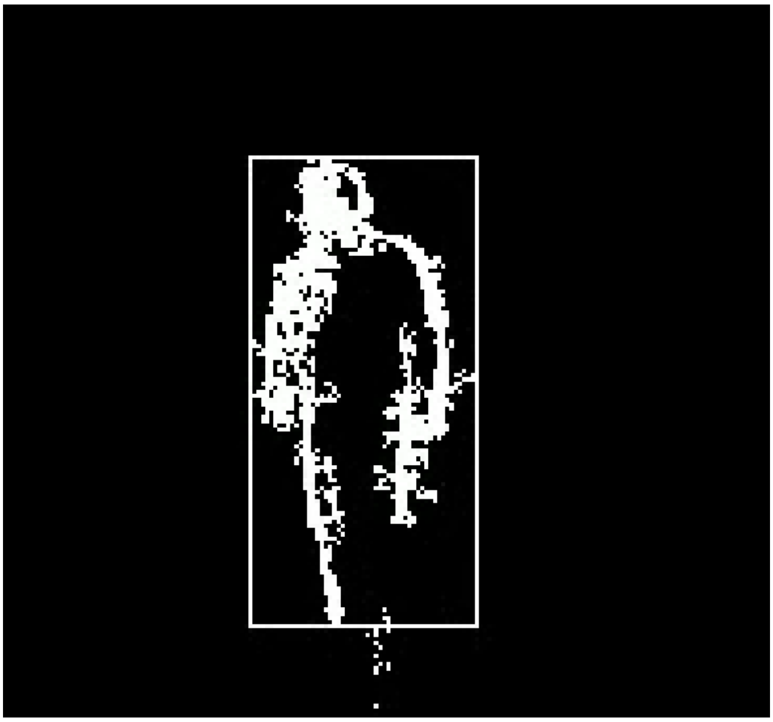

Figure 3. Figure showing the missing of inside objects.

In the above picture inside of the man is missing from the detection of the object as it is only able to detect the edges of the man after some time since the inside pixels remain constant for some time.

3.) Number of hyperparameters: The number of hyperparameters to be tuned are three thresholds (one for background, one for false detection and another for connected components). Since the model is sensitive to the choice of these thresholds, it is important that the thresholds be chosen approximately correctly.

4.) Shadow detection: Another limitation is the detection of the shadow of the object along with the object.
Solution: The solution is instead of using RGB channels use normalized rgb where $r=R/(R+G+B)$, $g=G/(R+G+B)$, $b=B/(R+G+B)$). These chromicized coordinates have the advantage of being non sensitive to illumination and hence will be better if we don not want to detect the shadow in the images.

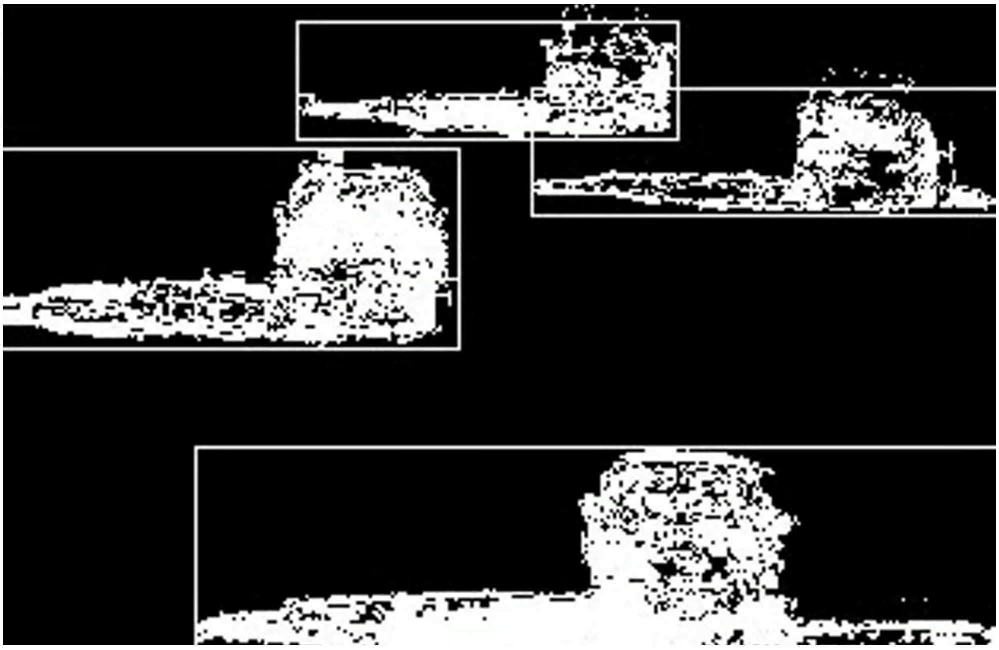

Figure 4. Shadows of the car in the above picture is also detected along with the object

5.) Disappearance of the object detected: Objects may disappear from the foreground if they remain still for some time as it will lead to them being assigned to the background unless they move again

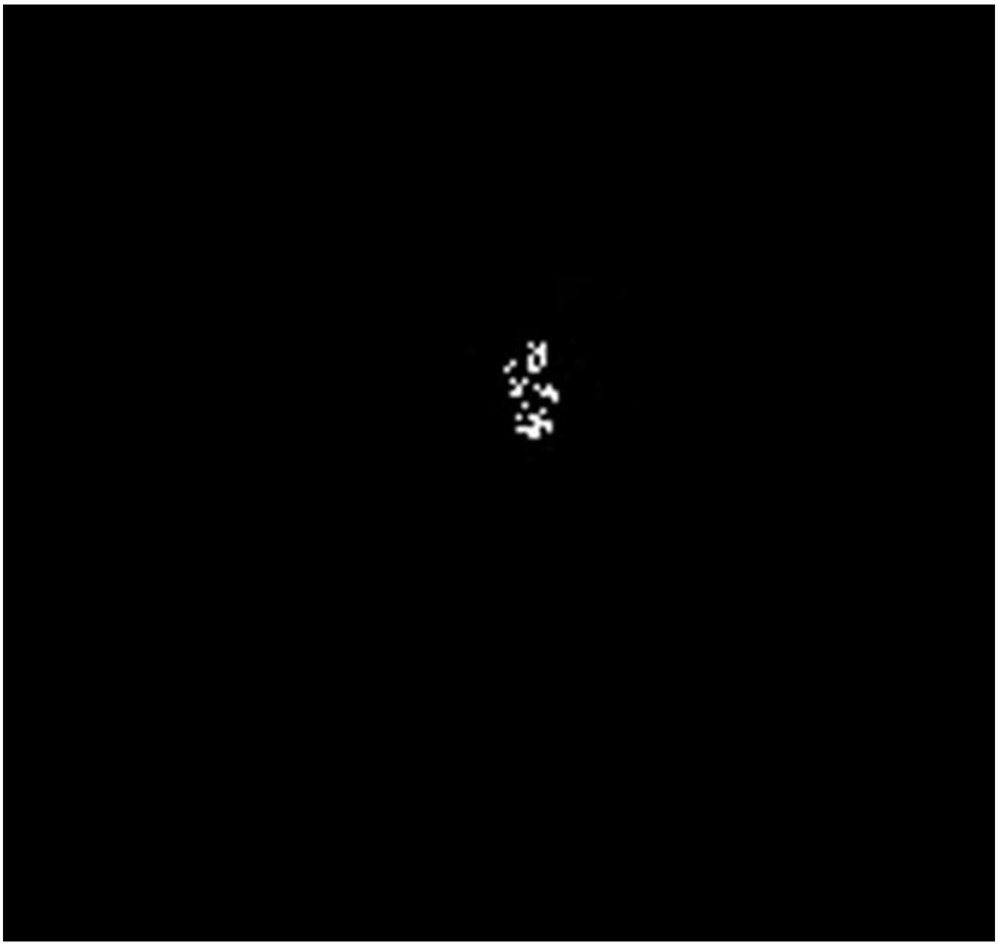

Figure 5. The object almost disappears for some time(2-3 frames) in the above picture

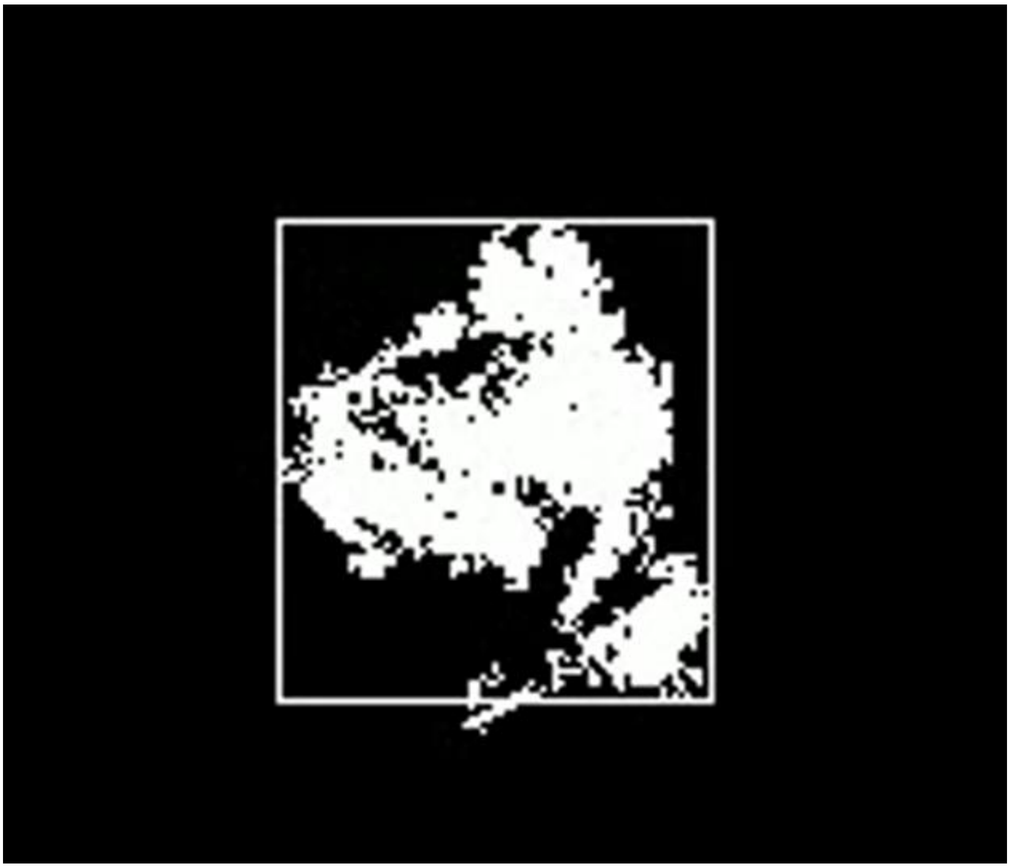

Figure 6. The object appearing after disappearing from the foreground after 5 frames.

The above two pictures are of a man resting in a sofa. In the first figure he became background as he had stop moving and in the second image he appears again once more. But compared with the parametric model this effect is very small.

Comparison with Parametric Gaussian Mixture models

1.) Ghosting effect: This happens mostly in parametric models when a pixel becomes part of a foreground and the object moves and the pixel becomes background parametric model may still consider it as a foreground unless another object occupies the pixel. This does not happen in non-parametric models as these are highly sensitive and depends only on the very recent frames compared to parametric models.

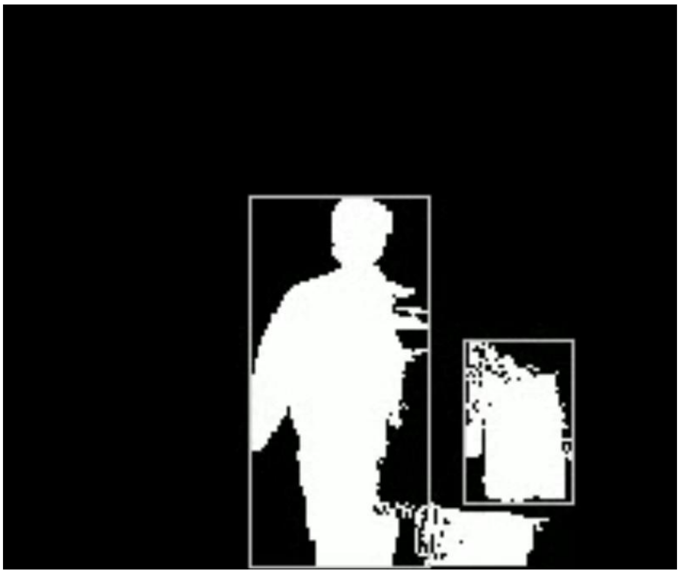

Figure 7. The right box detected is ghosting effect of the man entering the frame.

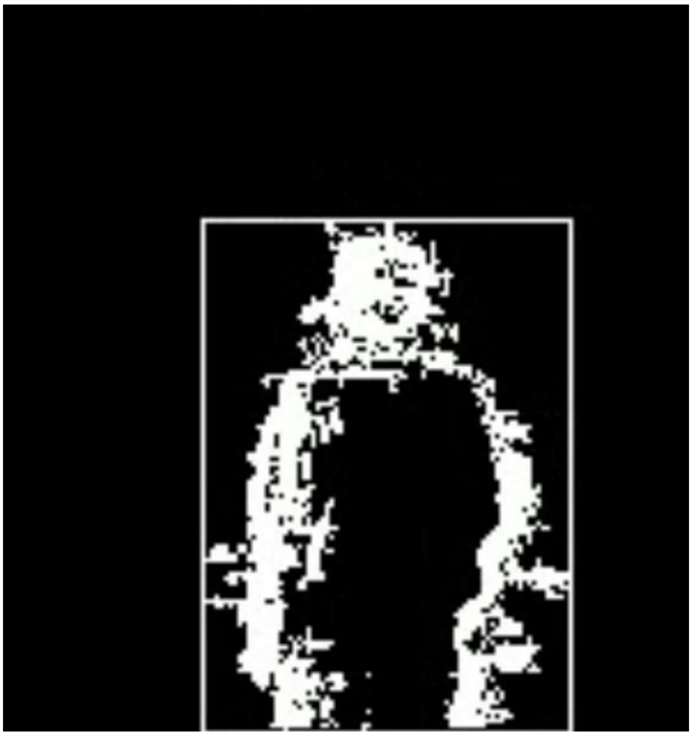

Figure 8. The same picture while using non-parametric model does not show the ghosting effect

The first picture shows the ghosting effect in case of parametric models where the object has already moved from that position but it is still detected (right small box) whereas this box does not appear in the second picture for the non-parametric models

2.) Disappearance of objects : As discussed in the limitations although the objects could disappear from the background in non-parametric models, this problem is much higher and happens for a longer time in parametric models and objects could disappear completely from the foreground. This normally does not happen in non-parametric models. One can easily visualize this in the example of Candela_m1.10 dataset where the man sitting in the couch disappears for a long time in parametric case while he disappears only for 2-3 frames in non-parametric case.

Figure 9. The object completely disappears from the frame compared to Figure 5 in case of parametric model

In the above picture the object completely disappears while in Figure 5. It does not completely disappear from the detection.

3.) Starting Frames: The learning at the start of the video seems to be much more better in parametric case rather than the nonparametric one. This is because the non-parameter model will take at least N frames to learn while the parametric model with decaying weights will learn very quickly and hence there won't be any delay for detection while the non-parameter model will have a lot of noise in the starting frames.

Figure 10. Starting frames in Parametric model

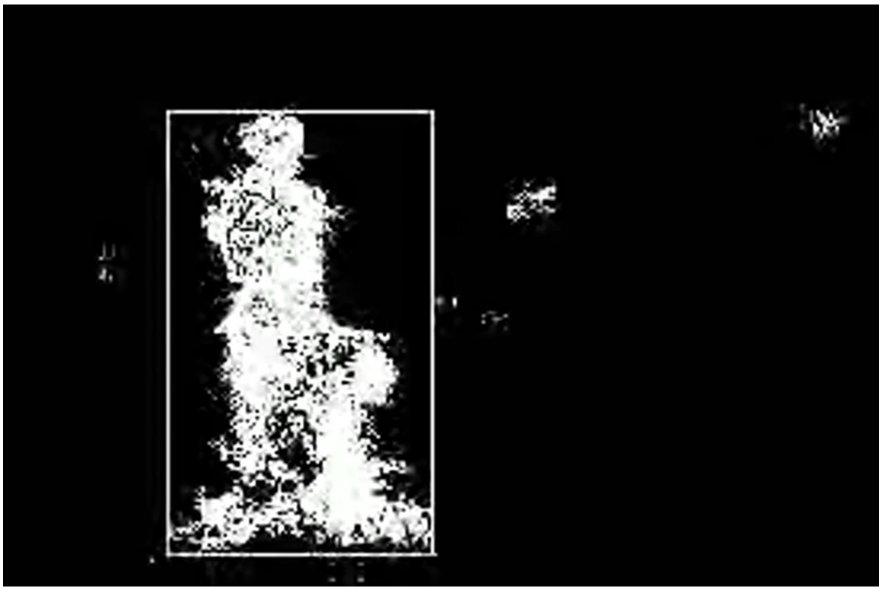

Figure 11. Starting frames in a non-parametric model

The pictures above are the starting frames from Hall and Monitor dataset. As it is visible from them the non-parametric model has a lot of noise in the beginning of detection compared to the parametric model.

4.) Inside of objects: Another noticeable difference is in detecting the inside of a big object. The parametric model seems to do well in this case as it is able to detect the inside of objects pretty well compared to the non-parametric models as these are highly sensitive and the inside could become part of background. This is illustrated in the images given below.

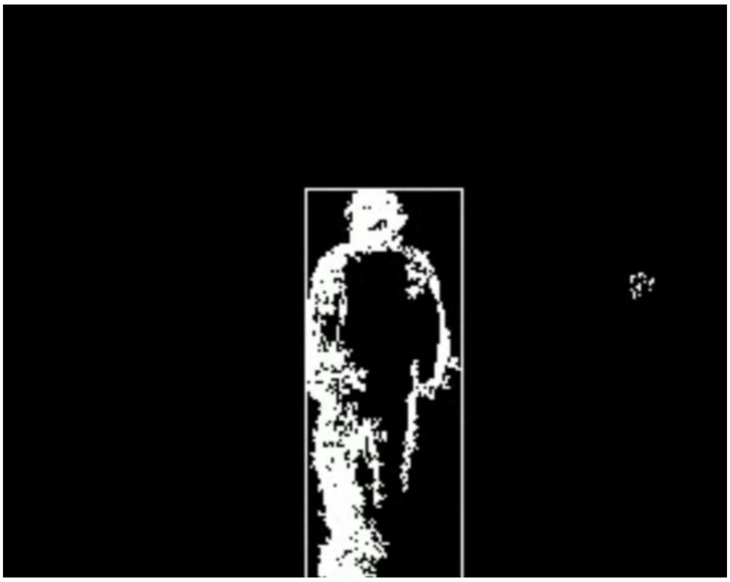

Figure 12. Inside of the object has become part of background in non-parametric case

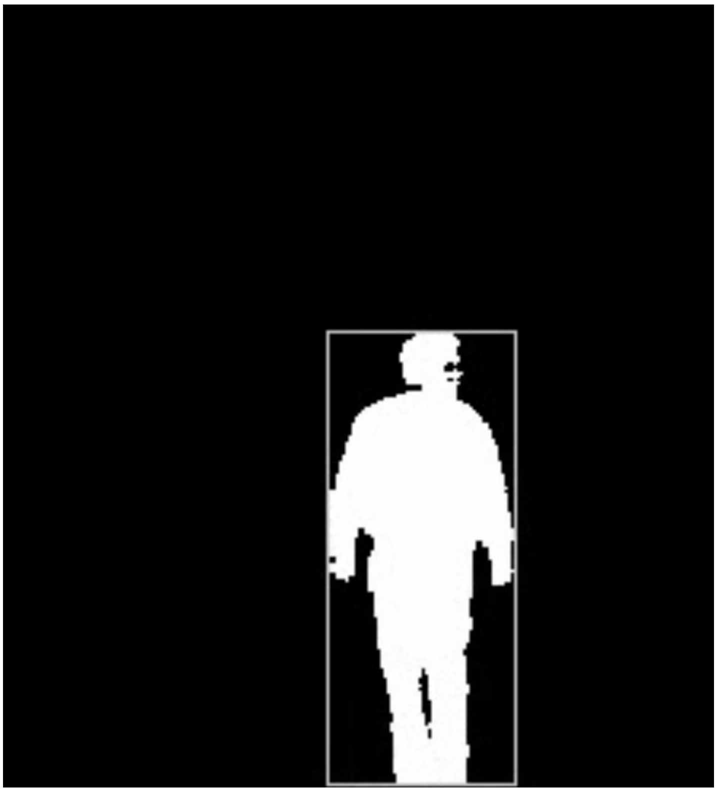

Figure 13. Inside of the object is still part of the object being detected in foreground

Link of the Results and Code

[COL780 Assignment 1]

Conclusion

Non-parametric model is a sensitive model compared to parametric one and is better for detecting objects in a security camera where sensitivity is a problem of concern. Another advantage is that it avoids the problem of ghosting in images and hence reduces the false detection to a large extend and hence could be used for object tracking. Although there are some limitations to the non-parametric model which was mentioned earlier, these could be avoided to a large extend by following the solutions mentioned and thus could be made to a powerful tool.