Detecting Distant Moving Target Using Infrared Sensors

Arisa Kitagishi, Babak Ebrahimi, Dr. Abhijit Mahalanobis
Center for Research in Computer Vision (CRCV), University of Central Florida (UCF)

Abstract

A popular topic in the recent research community is computer vision. And it has left significant results throughout the past several years. However, not many tackle the issues that lies in the motion detection of distant objects especially with videos obtained from the infrared sensors. Thus, we propose a method to accurately predict the moving targets that are more than 4km away from the static camera. This is done by incorporating frame differencing, background subtraction, and fully connected convolutional network (FCN) together to estimate where the motions of the desired targets are. We outperform the state-of-the-art generic object detectors (pre-trained).

1. Introduction

There are many successful object detection research in the community, especially within the past few years. However, many focus on creating the bounding box around the object and not consider the objects that may not be distinguishable. Our method focuses on detecting objects that are almost indistinguishable due to the distance from the camera. This motion is depicted by few pixels in the frames. Thus, making it harder for detection. To solve this problem, we decided to utilize the frame differencing and background subtraction to minimize the false alarm rates and stack the images together to keep the temporal information between the frames before sending them to the network. By doing so, our network can predict the target accurately. Figure 1 shows how we predict the targets and compare it to the groundtruth.

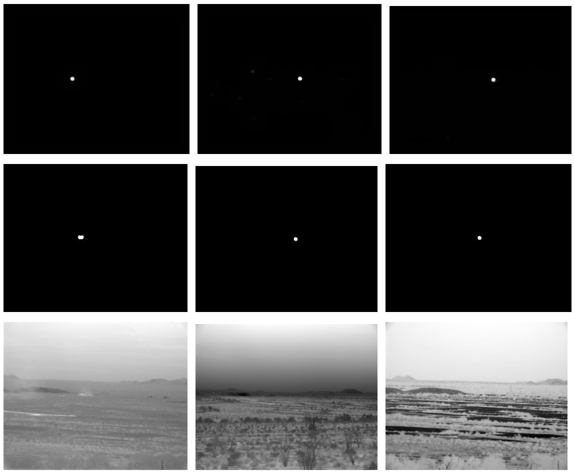

Figure 1: The first row consists of the predictions made by the network. The second row lists the groundtruth. And the last row contains the current frame image where the network tries to predict the target.

2. Related Work

2.1 Frame Differencing & Background Subtraction

We use absolute difference and adaptive threshold to perform frame differencing and background subtraction. There are four different frame differences in one input. Figure 2 illustrates how this is performed. These differences are stored as channels in the input. By doing so, it keeps the temporal information even though our method utilizes 2D convolutional network.

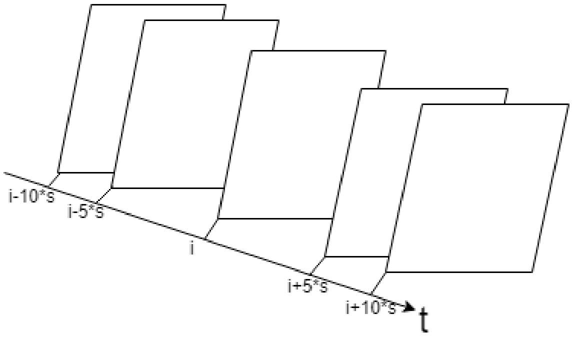

Figure 2: The variable i indicates the current time of the video which specifies which frame we are currently on. The other variable t indicates the time of the video. And the variable s indicates the number of skipped frames when loading the data. During frame differencing, we look at the two upcoming frames and two previous frames that are at 5 or 10 frames away from the current frame. Thus, if the current frame is at i with no skipping, we look at frames that are at i+5 and i+10. If we skipped three frames per frame when loading in the data, we would look at i+5*3 and i+10*3 which means 15 and 30 frames away from the current frame.

2.2 Generic Object Detectors

There are many published methods to detect any types of objects such as faster RCNN[1] and YOLO[2]. Figure 3 shows some of the results from the pre-trained faster RCNN. Because the objects are much smaller in our dataset, faster RCNN and YOLO tends to miss them or detect excessively which results in low recall with high false alarm rate.

3. Our method

3.1 Frame Differencing & Background Subtraction

We perform frame differencing by applying absolute difference between the frames to detect the motion of the target. However, the differences also mark the slight movements of the background which leads to high false alarms. To solve this, we implement adaptive thresholding and normalization to apply background subtraction. By doing so, we successfully eliminate most of the background and correctly detect which of the detections are targets. We perform this frame differencing and background subtraction four times per input where each input will contain a stack of five images. Afterwards, there will be 4 channels where each channel will contain predicted targets. Figure 4 shows how each channel may look like.

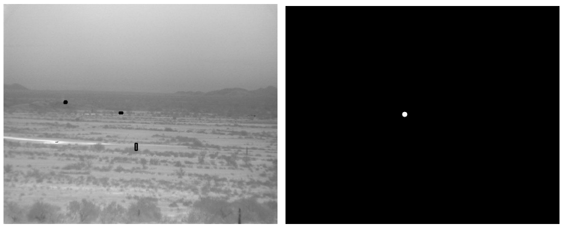

Figure 3: **Right**: the prediction made by the pre-trained faster RCNN indicated by the bounding boxes.**Left**: the groundtruth,

Figure 4: **Top Left:** channel 1 **Top Right:** channel 2 **Bottom Left:** channel 3 **Bottom Right:** channel 4

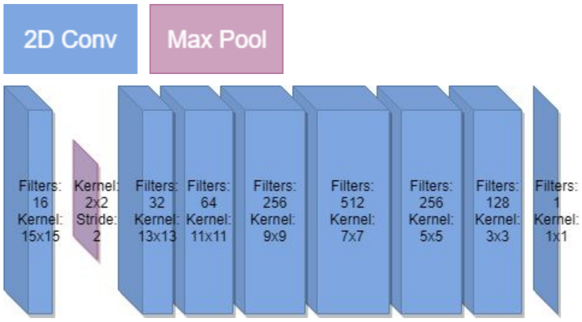

Figure 5: The architecture of FoveaNet. All layers have ReLU activation functions except the last layer which has Sigmoid activation layer to ensure the output is within 0 and 1.

3.2 FoveaNet

To detect objects that are significantly minor, we decided to apply FoveaNet[4] as it performs outstandingly when detecting small objects that are far away. This CNN also keeps the temporal information much better than 3D convolutional network by stacking images that consists of current frame, two previous frames, and two following frames, and then sending it to the network along with frame differencing and background subtraction. The structure can be seen in figure 5.

4. Experimental Results

4.1 Without CNN

Without the CNN implies that the results in figure 6 is obtained with only background subtraction and frame differencing. There was no deep learning applied to the result, but it was able to achieve over 80% of recall. However, it reaches high false alarm rate in order to achieve the high recall. To lower the false alarm rate, we applied the FoveaNet[4] to this method. This plot can be seen in figure 7.

4.2 With CNN

As seen in figure 7, our best model was able to achieve a recall of over 89% with false alarm rate of 0.008. Comparing figures 6 and 7, our method significantly shows improvement from the other.

4.3 Other Object Detectors

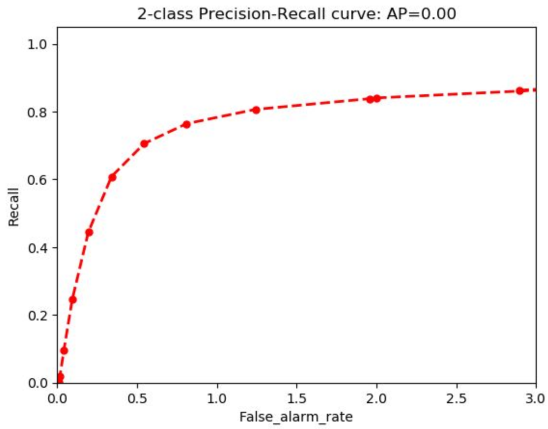

Figure 6: The plot is obtained with only background subtraction and frame differencing.

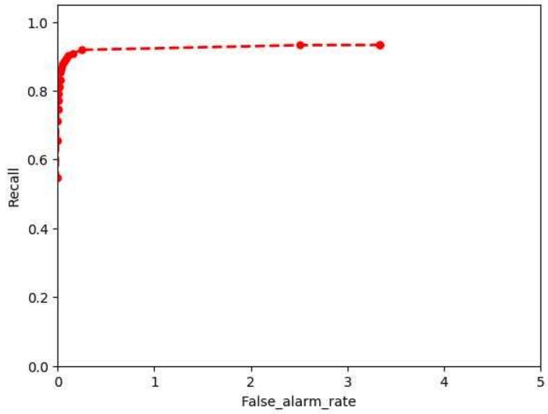

Figure 7: The plot is obtained from our method. It was able to achieve a recall of 89.4% with false alarm rate of 0.0084.

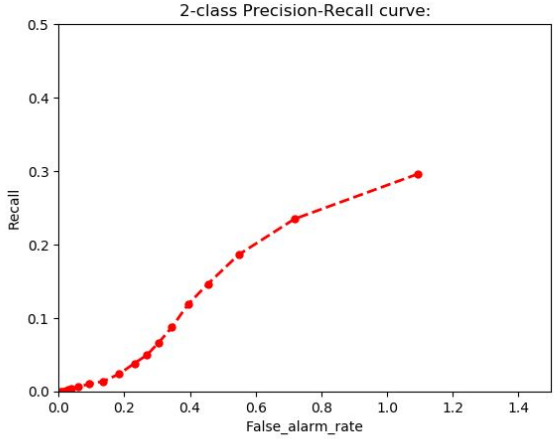

Figure 8: The plot is obtained from pre-trained Faster RCNN from the torchvision detection models which are trained with the COCO dataset[3].

Figure 8 shows a plot obtained from a pre-trained faster RCNN[1] for better comparison with the other plots. The pre-trained YOLOv3[2] was only able to achieve 0.995% of recall while faster RCNN was able to achieve a bit more than 30% of recall. They did not perform as well as our method, but this is most likely due to the fact that they are pre-trained to a different dataset such as COCO[3] that contains much larger and closer objects.

5. Conclusion

Frame differencing and background subtraction provides great support in detecting objects at long ranges. Currently, FoveaNet[4] is able to provide decent results for the dataset we are using which is ATR Dataset. However, we still need more training and testing to see possible improvements for the network. We would also like to train the other object detectors to properly compare their results to our method. There is a possibility that we may be able to combine some of the features form the other detectors to improve efficiency and accuracy of the current network. We are also looking to find out if we could analyze the vehicle's speed and distance to know how far it is from the camera as well as if we could detect the object from long ranges.

7. References

[1] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 36(6):1137–1149, June 2017.

[2] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 779–788, June 2016. Las Vegas, NV.

[3] T. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollar, and C. L. Zit- ´ nick. Microsoft COCO: common objects in context. arXiv e-prints, arXiv:1405.0312 [cs.CV], 2014

[4] R. LaLonde, D. Zhang, and M. Shah. Clusternet: Detecting small objects in large scenes by exploiting spatio-temporal information. In CVPR, 2018.