2021 IEEE International Conference on Image Processing (ICIP) | 978-1-6654-4115-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICIP42928.2021.9506700

DETECTION OF SMALL MOVING GROUND VEHICLES IN CLUTTERED TERRAIN USING INFRARED VIDEO IMAGERY

Adam Cuellar, Abhijit Mahalanobis

Center for Research in Computer Vision, University of Central Florida, Orlando, FL 32816-8005

ABSTRACT

The detection of small moving targets in cluttered infrared imagery remains a difficult and challenging task. Conventional image subtraction techniques with frame-to-frame registration yield very high false alarm rates. Furthermore, state of the art deep convolutional neural networks (DCNNs) such as YOLO and Mask R-CNN also do not work well for this application. We show however, that it is possible to train a CNN to detect moving targets in a stack of stabilized images by maximizing a *target to clutter ratio* (TCR) metric. This metric has been previously used for detecting relatively large stationary targets in single images, but not for the purposes of finding small moving targets using multiple frames. Referred to as *moving target indicator network* (MTINet), the proposed network does not rely on image subtraction, but instead uses depth-wise convolution to learn inter-frame *temporal* dependencies. We compare the performance of the MTINet to state of the art DCNNs and a statistical anomaly detection algorithm, and propose a combined approach that offers the benefits of both data-driven learning and statistical analysis.

Index Terms— Detection, Localization, Infrared, CNN

1. INTRODUCTION

High performance DCNN object detectors have been mainly developed for use with visible band color imagery to detect well resolved objects. However, the detection performance for small objects remains a challenging problem, even for conventional RGB imagery. Typically, modern detection networks are benchmarked on common datasets such as the Microsoft common objects in context (MS COCO) [1] dataset. MS COCO defines small, medium, and large objects as those with areas (defined in terms of total pixels) less than 32^2, between 32^2 and 96^2, and greater than 96^2, respectively. Current state-of-the-art (SOTA) networks, such as You Only Look Once (YOLO) and Mask R-CNN, perform inadequately on the small objects in the MS-COCO dataset. The average precision (AP) on large objects is more than twice of that on small objects for each of these networks [2] [3]. Therefore, we ask the question: What improvements can be made to increase the performance of small object detection techniques, and can it be realized on types of imagery other than RGB data?

For surveillance applications using infra-red imagery, the goal is to detect small objects at long ranges based on their movement between successive image frames. The challenge is illustrated in Figure 1, which shows the distribution of pixels on the targets to be detected, and an example of a target at 1km (close) and 4km (far). The bottom right of the figure also shows the difference of two frames indicating the movement of the small target against the background. We aim to tackle this problem by first investigating whether existing object detection algorithms can be repurposed to find small moving targets in infra-red imagery. We then introduce the MTINet which specifically learns temporal inter-frame dependencies that are characteristic of target motion, and maximizes the TCR metric for detection of moving targets. We also propose a strategy for combining it with a statistical anomaly detection algorithm to achieve the best overall results.

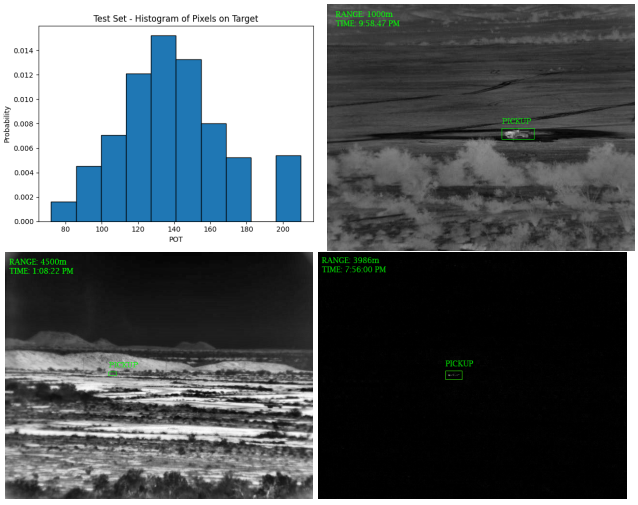

Fig. 1. Histogram of Pixels on Target for the NVESD testing set (top left), example of target at 1Km (top right) and at 4Km (bottom left). Difference image showing target movement at 4Km (bottom right)

The authors gratefully acknowledge the support of the US Army Night Vision and Electronic Sensors Directorate for this research

2. DATA SET AND BASELINE EXPERIMENTS

The experiments reported in this paper are conducted using a public domain dataset released by the US Army Night Vision and Electronic Sensors Directorate (NVESD) [4]. This data set contains a collection of mid-wave infrared (MWIR) imagery of vehicular targets moving at a constant velocity along a circle with a diameter of about 100 meters. The size of the images is 512x640, and frames were captured at 30Hz. The data was collected during the day and night, and at different ranges between 1000m and 5000m in increments of 500m. There are ten different types of vehicles in this data set. Figure 1 shows a day-time image of a typical target at a range of 4000m. Note that the target is relatively small and difficult to find in the surrounding cluttered terrain. The dataset was split into a training and testing sets using the provided range information. To focus on the ability to detect small objects, we use the targets at ranges 4000 to 4500 meters for training, and the 5000m range for testing. Every fifth frame of the original videos was selected to allow sufficient movement of the target between successive images. The effective frame rate is therefore 6Hz. In total 10484 images were used, with 7090 images allocated for training and 3394 images for testing. The distribution of the number of pixels on target (POT) is also shown in Figure 1. We note that the maximum size of the targets meets the criteria of a small object with the area $< 32^2$, as defined in MS COCO.

The key performance metrics are the probability of detection P_d (which is the number of targets detected divided by the total number of targets) and the false alarm rates (FAR) defined as the number of false positives per frame. Although our goal is to achieve high probability of detection while keeping the false alarm rate low, we are particularly interested in detection performance at very low false alarm rates between 0 to 0.5 false alarms per frame.

2.1. Results on YOLOv3 and Mask RCNN

We evaluated YOLOv3 for target detection using both single and multiple image frames. We initialize YOLOv3 with pretrained weights from the MS COCO dataset, and then fine-tune on the NVESD dataset until the training loss no longer decreased. Using single frames, YOLOv3 achieved a mean average precision of approximately 95.87% on the training set; however, the mean average precision on the testing set was just 3.52%. For the training set, the P_d was greater than 90% but when evaluating the testing set, we saw a maximum P_d of approximately 7%. The test set P_d vs FAR receiver operating characteristic (ROC) curve obtained using YOLOv3 single frame case is shown in Figure 2. We also trained and evaluated a modification of YOLOv3 with LSTM layers; however, the performance was similar to that of the original YOLOv3.

Although several image frames can be simultaneously processed as a multi-channel input by YOLOv3 and Mask R-CNN, neither network is designed to directly learn movement of objects between frames. Therefore, to enable these networks to detect a moving target, four *difference images* were produced using a stack of five images. These were obtained by taking the magnitude of the difference between the third image and each of the other ones. Using a stack of four such difference images, YOLOv3 achieved a mean average precision 20.68% on the testing sets, while the P_d also improved by approximately 23% as shown by the corresponding ROC curve in Figure 2 .

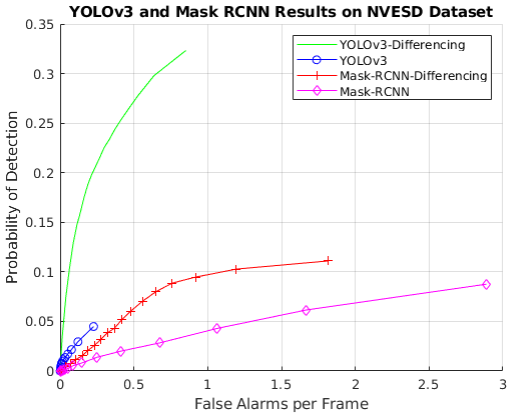

Fig. 2. YOLOv3, Mask R-CNN results on NVESD testing set

We also evaluated the Mask R-CNN network with ResNeSt backbone. This was also initialized with pretrained weights from the MS COCO dataset, and fine-tuned on the NVESD dataset until the training loss no longer decreased. We chose ResNeSt due to its state-of-the-art attention mechanism [5]. Using single frames, Mask R-CNN achieved a mean average precision of 93.67% and 1.34% on the training and testing set, respectively. On the training set, Mask R-CNN accurately detected targets with a maximum probability of detection of 90%; however, the false alarm rate was significantly higher than the results of YOLOv3. Figure 2 shows the performance of Mask R-CNN on the test set for both the single frame and difference images cases. Similar to YOLOv3, Mask R-CNN performs slightly better on the stack of difference images compared to the single image frames. Specifically, while the overall P_d improved slightly, the FAR at a $P_d \approx 0.9$ was reduced from ≈ 3.0 to 0.75 false alarms per frame.

3. PROPOSED APPROACH

Although difference images were found to improve the performance of SOTA DCNNs, it is preferable to allow the network to directly learn how to exploit the joint spatio-temporal information contained in the stack of input image frames. To properly exploit the spatial and temporal information, the

network must utilize features from each image in the block of frames. The architecture of the proposed MTINet shown in Figure 4 seeks to allow the network to emphasize cross-dependencies by using depthwise convolutions in its first two layers. This is followed by a variant of the attention modules from the Convolutional Block Attention Module [6]. To include the concept of attention, without suppressing necessary features, we modify the implementation to emphasize both the spatial and inter-channel relationship of features. Unlike the Convolutional Block Attention Module implementation in [6], we utilize a pointwise convolution on the depth-wise concatenated features extracted from the pooling operations. It is anticipated that the pointwise convolution allows for the combination of information, through the reduction of the number of filters, in a manner that accentuates the independent features of each frame, and thereby develops the necessary cross-dependencies between them.

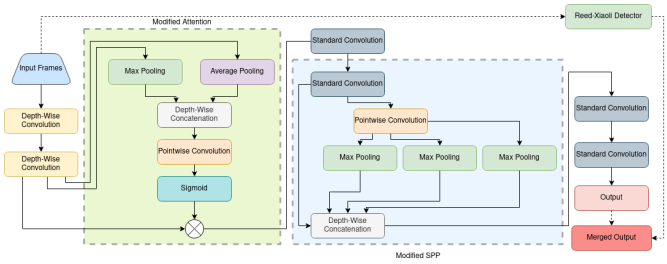

Fig. 3. Architecture combining MTINet-DW and the RX algorithm. MTINet-DW as approximately 1.3M parameters.

To further improve the performance of the network, we introduce the Spatial Pyramid Pooling block which has shown a promising increase in performance in state-of-the-art detectors such as YOLOv3 [5]. The purpose of this block is to increase the range tolerance of the network as well as emphasizing high frequency features. We include this block into the stem of the network.

To extract each region of interest, we locate the coordinates and value of the maximum response from the output of the network. We replace all values in a 20×20 window around the maximum response with value of the minimum response to ensure the detection is not counted more than once. We continue this process for up to ten regions of interest. The values extracted from the regions of interest are then normalized and used as the confidence score of the respective detection.

TCR Cost Function: Conventional regression-based training methods minimize the MSE loss between the actual and ideal desired response of the network. However, we observed that this cost function does not work well for our application where the shape of the desired response is not important. Rather, it is essential to produce a strong response at the true location of the targets, while attenuating the output of the network produced in response to clutter. This is achieved by using the Target to Clutter Ratio (TCR) cost function which was originally developed by McIntosh et al for finding large stationary targets in a single image. The TCR cost function aims to maximize the response of the target while minimizing the response of clutter [7]. This is done by minimizing the ratio $J_{TCR} = \frac{\frac{1}{N}\sum {y_i}^T y_i}{\sqrt[N]{\prod {x_i}^T x_i}}$, where y_i is the output at the location of background clutter, and x_i is the output at the location of true targets. Hence the TCR metric is the ratio of the arithmetic mean of the energy produced in response to clutter and the geometric mean of the energy of the output in response to the target [7]. The derivative of this function with respect to clutter and targets is $\nabla_{y_i} \log(J^{'}_{TCR}) = \frac{2y_i}{\sum y_i^T y_i}$, and $\nabla_{x_i} \log(J^{'}_{TCR}) = -\frac{1}{N}\frac{2x_i}{x_i^T x_i}$. While McIntosh et. al used separate training exemplars for targets and clutter, we compute the gradient for both target and clutter simultaneously for each training sample. For a input stack of five images, the true position of the target is specified as its ground-truth location in the third (or middle) image frame. During training, the gradient supplied to the back-propagation algorithm is $\nabla_{y_i} log(J^{'}_{TCR})$ for clutter regions of the image, and $\nabla_{x_i} log(J^{'}_{TCR})$ for region where the target is actually located.

Reed-Xiaoli Algorithm: The Reed-Xiaoli (RX) detector is a constant False Alarm Rate algorithm originally developed for detecting anomalous pixels in hyperspectral data [8]. We re-purpose the RX detector to find moving targets in a stack of registered image frames to find anomalies in the temporal dimension. It assumes the temporal noise at each pixel follows a Gaussian distribution, and uses Log-likelihood Ratio Test to identify temporal anomalies that have a low likelihood of having occurred due to noise. More specifically, given a a block of D image frames, the algorithm computes the Mahalanobis distance at each pixel as $RX(x_i) = (x_i - \mu)^T K^{-1}_{D\times D}(x_i - \mu)$ where x_i is a $D\times1$ column vector representing the $i-th$ pixel across all frames, and μ and $K_{D\times D}$ are its mean vector and covariance matrix respectively, estimated over a pre-defined window surrounding the pixel [9]. Large values of $RX(x_i)$ indicate potential moving objects at the location of the $i-th$ pixel.

4. RESULTS

We conduct experiments using the NVESD test set, to evaluate the ability of the MTINet and the RX algorithms to find small moving targets in challenging background clutter. Although more targets can be detected at higher FAR, we cmpare P_d at low FAR below 0.5 false alarms per frame. First, we separately evaluate MTINet and the RX detector's performance. Then we evaluate the combination of both MTINet and the RX algorithm. The effect of optimizing the TCR metric is shown in Figure 4. We note that the output activation of the MTINet is sharply focused on the target's location, whereas that of YOLOv3 is not. This enables the MTINet to

accurately detect the small targets compared to SOTA CNNs. Figure 5 shows the results of MTINet on the NVESD testing data. We see that P_d reaches 95% at higher false alarm rates. However, at a lower FAR of 0.5, the network achieves a P_d of approximately 74%, as shown in the zoomed plots in Figure 6. Figure 5 also shows the performance of the RX algorithm achieves a maximum P_d of approximately 85% which is lower than that of the MTINet. However, its P_d at the lower false alarm ranges outperforms the MTINet, as shown in Figure 6. Specifically, at a false alarm rate of 0.5 the RX algorithm achieves a similar performance to MTINet with a probability of detection of about 75%. However, at a false alarm rate of 0.1, the algorithm reaches a probability of detection of 60% which is higher than MTINet-DW by about 10%.

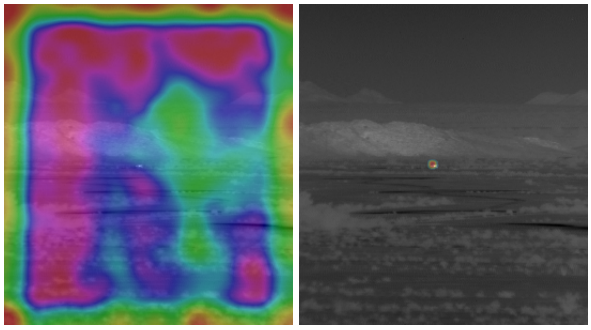

Fig. 4. Final activation map MTINet (right) is sharply focused on the target's location, whereas the YOLOv3 (left) is not.

We observe that the trained CNN is better at finding more difficult targets at higher false alarm rates, but the RX detector finds the relatively easier targets with a lower false alarm rate. This is because while training enables the CNN to detect weaker targets missed by the RX algorithm, it may also give up some performs on brighter targets which can be easily detected using temporal intensity variations induced by movement. To realize the best of both approaches we use the RX algorithm concurrently with the MTINet, and combine their outputs. We do so by normalizing the scores, and merging the output detection lists from both algorithms. Figure 5 also compares the overall ROC curve of the combined algorithm versus the individual performance of each on the NVESD testing set, while the zoomed version of the same at low FAR is also shown in Figure 6. These figures shows the combined algorithms offers the best of both by detecting as many targets as the MTINet while mitigating drop in P_d at lower values of FAR.

Network	Max PDet	FAR @ Max PDet	PDet @ 0.1	PDet @ 0.5 FAR
YOLOv3	0.32	**0.85**	0.15	0.27
Mask R-CNN	0.11	1.78	0.03	0.07
MTINet	**0.95**	5.0	0.51	0.73
MTINet + RX	**0.95**	5.0	**0.59**	**0.75**

Table 1. Comparing results of best performance experiments

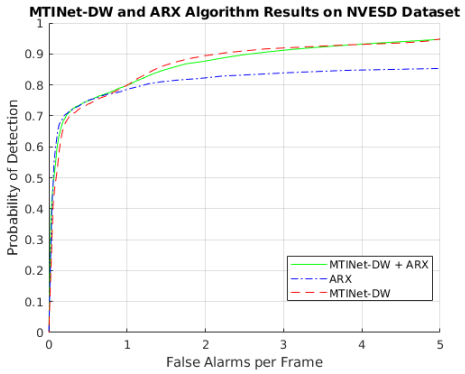

Fig. 5. Results of MTINet-DW and the ARX algorithm on the NVESD testing set

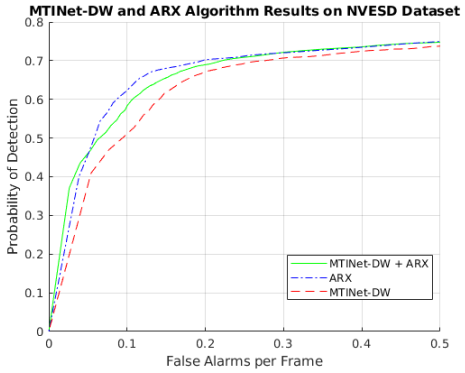

Fig. 6. Results of MTINet-DW and the ARX algorithm on the NVESD testing set within a false alarm rate of 0 to 0.5

5. CONCLUSION

We have proposed the MTINet, a new network for detecting small moving targets in infra-red imagery. Using a publicly released IR dataset, we showed that the proposed algorithms outperform other SOTA networks (e.g YOLOv3 and Mask R-CNN) for such applications. Table 1 compares their performance to the MTINet and its combination with the RX algorithm. Although YOLOv3 achieves the lowest false alarm rate, its maximum P_d is merely 32%. In contrast, the MTINet detects as many as 95% of targets, albeit at a higher FAR. Even at FAR=0.1, the MTINet detects approximately 51% of targets, which is approximately 2.5 times more than the YOLOv3. This performance increase is attributed to the architecture of the MTINet which exploits temporal information, and the optimization of the TCR cost function which enables the network to discern between targets and clutter. Finally, the combination of the MTINet with RX anomaly detection not only achieves the maximum P_d of 95%, but also yields the highest P_d=0.59 at the low FAR value of 0.1.

6. REFERENCES

[1] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C.L. Zitnick, and P. Dollár, "Microsoft coco: Common objects in context," *Springer*, vol. 8693, pp. 740–755,.

[2] Joseph Redmon and Ali Farhadi, "Yolov3: An incremental improvement," *CoRR*, vol. abs/1804.02767, 2018.

[3] P. Dollár K. He, G. Gkioxari and R. B. Girshick, "Mask r-cnn," *CoRR*, vol. 1703, no. 06870.

[4] Defense Systems Information Analysis Center, "ATR Algorithm Development Image Database," https://www.dsiac.org/resources/available-databases/atr-algorithm-development-image-database/, Last accessed on 2020-08-01.

[5] H. Zhang, C. Wu, Z. Zhang, Y. Zhu, Z. Zhang, H. Lin, Y. Sun, T. He, J. Muller, R. Manmatha, M. Li, and A. Smola, "Resnest: Split-attention networks," arXiv preprint, 2020.

[6] W. Sanghyun, J. Park, J.-Y. Lee, and I.S. Kweon, "Cbam: Convolutional block attention module," in *Computer Vision - ECCV*, p. 3–19. Springer International Publishing.

[7] B. McIntosh, A. Mahalanobis, and S. Venkataramanan, "Infrared target detection in cluttered environments by maximization of a target to clutter ratio (tcr) metric using a convolutional neural network," *IEEE Transactions on Aerospace and Electronic Systems*, pp. 1–1,.

[8] I.S. Reed and X. Yu, "Adaptive multiple-band cfar detection of an optical pattern with unknown spectral distribution," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 38, no. 10, pp. 1760–1770,.

[9] C.-I. Chang and C. Shao-Shan, "Anomaly detection and classification for hyperspectral imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 40, no. 6, pp. 1314–1325,.