TARGET DETECTION IN CLUTTERED ENVIRONMENTS USING INFRA-RED IMAGES

Bruce McIntosh[1], Shashanka Venkataramanan, Abhijit Mahalanobis

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

ABSTRACT

The detection of targets in infra-red imagery is a challenging problem which involves locating small targets in heavily cluttered environments while maintaining a low false alarm rate. We propose a network that optimizes a "target to clutter ratio"(TCR) metric defined as the ratio of the output energies produced by the network in response to targets and clutter. We show that for target detection, it is advantageous to analytically derive the first layer of a CNN to maximize the TCR metric, and then train the rest of the network to optimize the same cost function. We evaluate the performance of the resulting network using a public domain MWIR data set released by the US Army's Night Vision Laboratories, and compare it to the state-of-the-art detectors such as Faster RCNN and Yolo-v3. Referred to as the TCRNet, the proposed network demonstrates state of the art results with greater than 30% improvement in probability of detection while reducing the false alarm rate by more than a factor of 2 when compared to these leading methods. Ablation studies also show that the proposed approach and metric are superior to learning the entire network from scratch, or using conventional regression metrics such as the mean square error (MSE).

1. INTRODUCTION

The detection of vehicular targets surrounded by natural background clutter in infrared (IR) imagery is a challenging problem [1]. The IR phenomenology differs significantly from the visible band as a result of which algorithms trained on conventional color images cannot be readily incorporated into IR applications. This is further compounded by the fact that there is generally a dearth of labeled IR data for training the algorithms. Although many algorithms have been proposed over the years to address this problem [2, 1, 3, 4] IR target detection at acceptably low false alarm rates remains a difficult problem. Of course, there has been a tremendous advanced in computer vision using convolutional neural networks (CNNs) and deep learning. Hence, there is significant interest in determining if similar performance gains can be achieved by applying these techniques in the IR domain.

To facilitate research in target detection and recognition, a MWIR data set [5] was made available to the research community by the US Army's Night Vision and Electronic Sensors Directorate (NVESD). This data set contains different scenarios with varying levels of difficulty. In fact, there is substantial variation in how well the targets are resolved at different ranges and in the background clutter during day and night conditions. Here "clutter" refers mainly to the terrain (background and foreground) and vegetation throughout the scene. Using this data set, a comparison of an existing method known as the Quadratic Correlation Filter (QCF) [6] and Faster R-CNN [7] was conducted [8] with the latter exhibiting significantly better performance. However in this initial study, there was significant overlap in the background clutter between training and testing images, which lead to optimistic results. Furthermore, this paper did not propose a method for optimizing the TCR metric using a combination of analytically derived filters for the first layer, with training for the rest of network to directly optimize target detection performance. Millikan *et al.*[9] also proposed using the QCF filters as the first layer in a CNN for classifying the 10 different types of targets in this data set, but did not address target detection at long ranges and under difficult clutter conditions. Other researchers [10] have reported the use of Faster R-CNN [7] to find moving targets using multiple frames of both visible band (which is also included in the same data set) and infrared imagery. In contrast, we tackle the more difficult problem of "single frame" detection of *stationary* targets in a single MWIR image frame. Specifically, our goal is to develop a method that is optimized for finding targets in high clutter, and to compare its performance under challenging conditions to that of other state of the art object detection algorithms such as Faster RCNN and Yolo-v3.

2. TCR NETWORK AND OPTIMIZATION

Figure 1 shows a typical MWIR image with a target in natural terrain and the output produced by the TCRNet. It should be noted that the target (indicated by the arrow) is not easy to find in such cluttered scenes. The first layer of the TCR network employs 100 relatively large filters (20×40 in the illustration) which are analytically derived to separate targets from clutter. Since the targets are relatively small and not well resolved at long ranges, smaller filters that attempt to learn the internal structural details do not work well. Therefore, the size of the filters in the first layer is large enough to cover the spatial extent of the targets. The activations produced by the

[1]Corresponding author. email at bwmcint@knights.ucf.edu

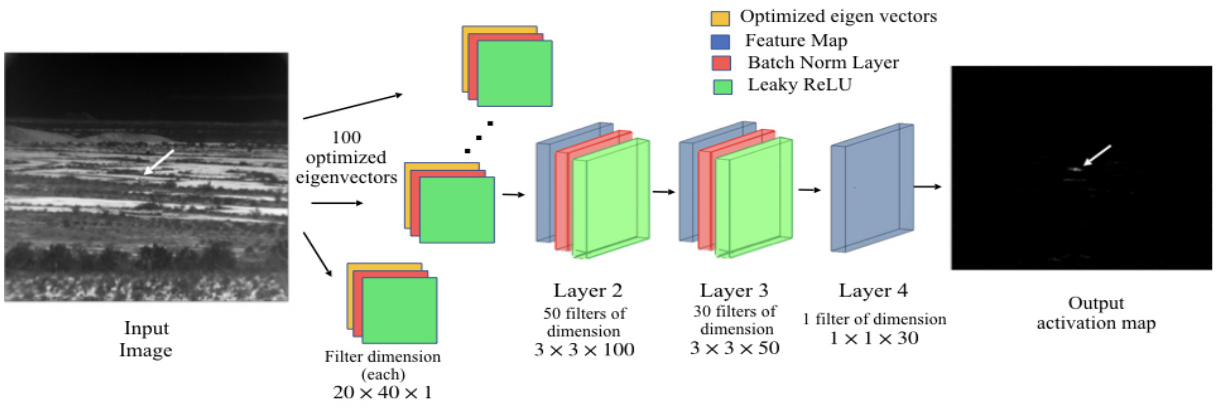

Fig. 1. Layer 1 of the TCRNet is analytically derived while the rest of the convolutional layers are iteratively learned to optimize the TCR metric. The output has a strong intensity at the location of the target in the input image (denoted by the arrow)

first layer are then post processed by two successive convolutional layers with fifty $3 \times 3 \times 100$ and thirty $3 \times 3 \times 50$ filters respectively. Each of these layers is preceded by Batch Normalization [11] and ReLU [12]. The output is produced by a single $1 \times 1 \times 30$ filter. No pooling or stride is used so that the spatial dimensions of the output directly correspond to that of the input image. Detection is performed by searching for local maxima in the output intensity values produced by the network.

2.1. TCR Metric and Derivation of Layer 1 Filters

Although metrics like cross-entropy or regression loss (such as mean square error or MSE) are commonly used to train CNNs, our goal is to maximize the energy in the response of the network to targets, and minimize the same in response to clutter. Consider the energy of the projection of an image vector x on a set of M vectors q_i given by

$$\phi = \sum_{i=1}^{M} |q_i^T x|^2 = \sum_{i=1}^{M} q_i^T (x x^T) q_i \tag{1}$$

In general, we wish ϕ to be as large as possible when x is a target image. However, the problem is that this can occur even if one of the terms in the summation is large, and the rest are small. For effective representation of the target class, we wish the projection of x on all the M basis vectors to be as large as possible. Therefore to make the output of each of the basis functions large in response to the target, we require their joint expectation to be maximized. Assuming independence between the terms, this can be expressed as

$$E\left\{\prod_{i=1}^{M} |x^T q_i|^2\right\} = \prod_{i=1}^{M} E\{|x^T q_i|^2\} = \prod_{i=1}^{M} q_i^T R_1 q_i \tag{2}$$

where $R_1 = E\{x_i x_i^T\}$ is the correlation matrix of the data for the target class. For the clutter class however, minimizing the statistic in eq. 1 will ensure the response of each of the basis functions is also small. Based on this reasoning, the TCR metric we propose is

$$J_{TCR} = \frac{\prod_{i=1}^{M} q_i^T R_1 q_i}{\sum_{i=1}^{M} q_i^T R_2 q_i} \tag{3}$$

where $R_2 = E\{x_i x_i^T\}$ is the correlaton matrix of the data for the clutter class. This metric is different than the original QCF performance criterion, and ensures that all examples of the targets produce a large output response. Taking derivative of eq. 3 with respect to q_i, we obtain

$$\nabla_{q_i} J_{TCR} = \frac{2R_1 q_i \prod_{i \neq j} q_j^T R_1 q_j}{\sum_{i=1}^{M} q_i^T R_2 q_i} - \frac{2R_2 q_i (\prod_i q_i^T R_1 q_i)}{(\sum_{i=1}^{M} q_i^T R_2 q_i)^2} \tag{4}$$

Setting the derivative to zero, and observing that $q_i^T R_1 q_i$ and $\sum_{i=1}^{M} q_i^T R_2 q_i$ are both scalars, we obtain

$$R_2^{-1} R_1 q_i = \left(\frac{q_i^T R_1 q_i}{\sum_{i=1}^{M} q_i^T R_2 q_i}\right) q_i = \gamma_i q_i \tag{5}$$

This clearly shows that q_i are the complete set of eigenvectors of $R_2^{-1} R_1$, and that they all play a role in the maximization of J_{TCR}.

2.2. Modified TCR cost function for training CNN

The analytical optimization of J_{TCR} yields a set of eigenvectors that maximize the representation of targets while minimizing the effect of clutter. The set of eigenvectors obtained in eq. 5 are treated as the input layer of a CNN. The rest of the layers are then adapted using a variant of the TCR metric suitable for learning via gradient descent.

The modified TCR metric is obtained as follows. Let us assume that we have N labeled samples for the target and

clutter classes which produce the final outputs of the network denoted by $\{x_1, x_2...x_N\}$ and $\{y_1, y_2...y_N\}$, respectively. Our objective is to maximize the energy in the output when targets are present, and minimize the same in response to clutter. This is accomplished by minimizing the ratio

$$J_{TCR}^{'} = \frac{\frac{1}{N}\sum y_i^T y_i}{\sqrt[N]{\prod x_i^T x_i}} \tag{6}$$

which is the ratio of the arithmetic mean of the energy of the clutter samples to the geometric mean of the energy of the target samples. Minimizing this ratio will make the numerator of $J_{TCR}^{'}$ small, which in turn ensures that all the terms in the summation $\frac{1}{N}\sum y_i^T y_i$ are small. Similarly the denominator of $J_{TCR}^{'}$ must be large to minimize the ratio, which implies that $\sqrt[N]{\prod x_i^T x_i}$ is large, which in turn ensures that all terms in the product are large.

It should be noted that this cost function is consistent with the optimization criterion in eq. 4 used for obtaining the generalized eigenvectors that best separate target and clutter. Therefore by minimizing $J_{TCR}^{'}$ the CNN layers will learn the decision boundary between the two classes using a cost function that is similar to the TCR metric used for finding the filters in the first layer of the network. Since $J_{TCR}^{'}$ is always positive, it is simpler to minimize its logarithm given by

$$\begin{aligned}\log(J_{TCR}^{'}) &= -\log(N) + \log(\sum y_i^T y_i) - \frac{1}{N}\log(\prod x_i^T x_i) \\ &= -\log(N) + \log(\sum y_i^T y_i) - \frac{1}{N}\sum \log(x_i^T x_i)\end{aligned} \tag{7}$$

The derivative of this function with respect to each class is

$$\begin{aligned}\nabla_{y_i}\log(J_{TCR}^{'}) &= \frac{2y_i}{\sum y_i^T y_i}, \\ \nabla_{x_i}\log(J_{TCR}^{'}) &= -\frac{1}{N}\frac{2x_i}{x_i^T x_i}\end{aligned} \tag{8}$$

Therefore, as training images are presented to the network during the learning process, the gradient supplied to the backpropagation algorithm is either $\nabla_{y_i}\log(J_{TCR}^{'})$ for clutter images), or $\nabla_{x_i}\log(J_{TCR}^{'})$ for target images. It should be noted that for one training image considered at a time, the gradient expression for the two classes reduce to $\nabla_{y_i}\log(J_{TCR}^{'}) = \frac{2y_i}{y_i^T y_i}$ and $\nabla_{x_i}\log(J_{TCR}^{'}) = -\frac{2x_i}{x_i^T x_i}$ which are simply the energy normalized outputs produced by the training images of the respective classes.

3. EXPERIMENTS AND PERFORMANCE ANALYSIS

In this section, we evaluate the performance of the TCRNet and compare it to that of Faster RCNN and Yolo-v3. Results are presented in the form of ROC curves that show probability of detection (P_d) as a function of false alarm rate (FAR).

Fig. 2. Examples of the image chips used for training the TCRNet. The 10 vehicle classes are shown along with 5 examples of clutter or background chips.

We define P_d as the ratio of number of true targets detected to total number of true targets in the test data. The FAR is defined as FAR $= \frac{\text{Total number of false positives}}{\text{Total number of frames} \times FOV}$ where FOV is the product of the horizontal and vertical fields of view of the sensor. For this data set, the infra-red camera had a 3.4 degrees x 2.6 degrees, and therefore *FOV* = 8.84 square degrees. Thus FAR is reported in units of "false alarms per square degree", abbreviated as "FA/sq degree".

All algorithms were trained on images at ranges of 1Km, 1.5Km, and 2.0Km, and tested on images at ranges of 2.5km, 3.0km, and 3.5km. Since the range (or distance) to the targets is given, this information was used to resize all images (both for train and test) to an apparent range of 2.5Km. While the test set has 9360 images, the training set has 10,800 images each for targets and clutter. However, the number of training images available in our data set is too few to train deep networks from scratch (which are pre-trained on much larger datasets). Therefore we use transfer learning to adapt state of the art methods to the IR target detection problem. We compare TCRNet with Faster-RCNN [13] which uses a Resnet-50 [14] pretrained on Imagenet [15] as its backbone and finetuned on our dataset. For Yolo-v3, we use the weights pretrained on MS-Coco [16], and finetune it on our dataset.

The optimum eigenfilters for the first layer are created as described in Section 2.1, using 20×40 image 'chips' or 'patches' derived from the training set. Specifically for each image in the training set, a 20×40 pixel region centered on the target is cropped out and used as a positive target example. Another randomly selected 20x40 pixel region of the image is cropped out and treated as a 'clutter' or background example. These 20×40 images are flattened into 800×1 dimensional vectors and used for estimating the 800×800 dimensional matrices R_1 and R_2. The resulting 800×1 eigenvectors q_i in eq. 5 are then reshaped into 20×40 filters that serve as the eigenfilters in Layer 1 of the TCRNet. The eigenvalues of the $R_2^{-1}R_1$ are strictly positive, but not bounded which makes it

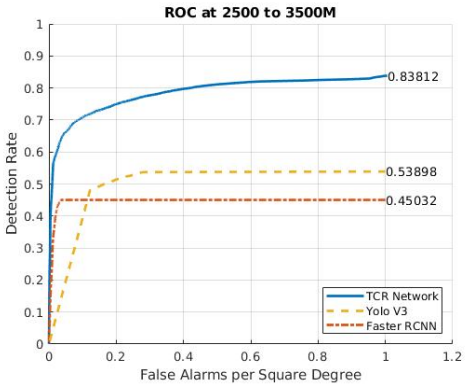

Fig. 3. ROC curves for the three networks. The TCR Network shows the best performance with a substantial margin of 30% and 38.8% over Yolo-v3 and Faster-RCNN respectively.

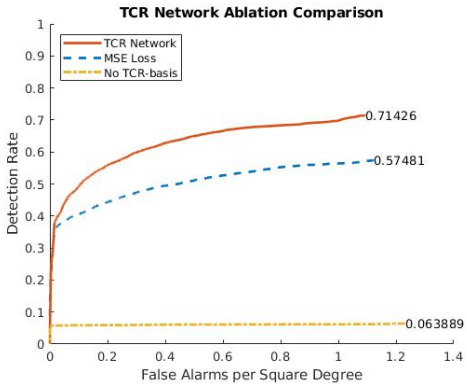

Fig. 4. Ablation tests show that replacing the TCR metric by the MSE regression loss, and training the entire network from scratch lead to poorer performance for the TCRNet

difficult to choose suitable eigenvectors. For this reason, the eigenvalues γ_i are remapped into a range $[-1, +1]$ as

$$\lambda_i = \frac{\gamma_i + 1}{\gamma_i - 1} \tag{9}$$

The eigen-vectors with eigen-values close to 1.0 contain more information about targets, whereas those corresponding to eigen-values closer to -1.0 are more representative of clutter. Eigenvectors with smaller eigenvalues do not distinguish well between the two classes, and therefore are discarded. Based on the values of λ_i , we selected the 70 eigenvectors that represent targets (corresponding to eigenvalues $730 - 800$), along with the 30 eigenvectors for clutter (corresponding to eigenvalues $1 - 30$). Together, they form the 100 20×40 filters for the first layer of the TCRNet. Once the first layer was derived, these filters were held fixed, while the remaining layers were trained. The network was trained with image chips of size 40×80 versions of the same training images used for deriving the layer 1 filter. The cost function described in eq. 7 was used for training over 25 epochs with the RMSprop optimizer [17], a batch size of 100, and initial learning rate of $1e^{-5}$.

For calculating detection accuracy, local maxima are used for the TCR Network while the centers of predicted bounding boxes are used for Yolo-v3 and Faster-RCNN. Non-max suppression is applied to these predictions, and then the distance is measured to the center of the ground truth bounding box. If the prediction is within a distance threshold (20 pixels in this case) it is counted as a correct detection. The ROC curves obtained on the test set for all the algorithms are shown in Figure 3. The test images are of size 640 x 512 and represent the full field of view of the sensor. The TCRNet shows the best performance with substantial margin of 30% and 38.8% over Yolo-v3 and Faster-RCNN respectively. It is clear that these object detection networks struggle with finding the relatively small targets in a cluttered background. We see that the FasterRCNN achieves a maximum detection of 45% with a FAR of 0.035 FA/sq degree. At this P_d the TCR Network has a substantially lower false alarm rates of 0.0072 FA/sq degree. Similarly the TCRNet has a FAR of 0.0116 vs. 0.28 FA/sq degree for Yolo-v3 at its maximum detection of 53.9%. Furthermore, at very low FAR (say 0.01 FA/sq degree), the TCRNet has much higher P_d compared to either of the other two detectors. Figure 4 is an ablation test for the TCRNet. The blue curve shows the results of training and testing the network with the MSE loss function (instead of the proposed TCR metric), while the yellow curve represents the results of training the entire network (including the first layer) from scratch. These plots show that both the TCR metric and the analytical derivation of the first layer enable the TCRNet to significantly outperform all other methods by learning to detect targets using relatively few training images.

4. SUMMARY

The TCR Network proposed in this paper is specifically designed for the detection of relatively small targets in infrared imagery under difficult and challenging clutter conditions. The network optimizes a TCR metric defined as the ratio of the energies produced at the output of the network in response to targets and clutter. The TCR metric not only ensures that clutter energy is minimized, but also emphasizes representation of targets in order to achieve high probability of detection. In general, there is also a dearth of large labeled data sets for training very deep networks from scratch. To address this problem, the first layer of the TCRNet uses analytically derived eigenfilters, while the later layers are learned via gradient descent. The TCRNet's performance was evaluated using the MWIR image data set released by NVESD, and compared to that of the Faster RCNN and Yolo-v3. It was shown that the TCRNet outperforms these other state of the art methods. Specifically, the TCRNet not only achieves a substantially higher P_d, but also delivers considerably lower FAR when compared at the maximum P_d achieved by the Faster RCNN, Yolo-v3.

5. REFERENCES

[1] James A Ratches, "Review of current aided/automatic target acquisition technology for military target acquisition tasks," *Optical Engineering*, vol. 50, no. 7, pp. 072001, 2011.

[2] Erhan Gundogdu, Aykut Koç, and A Aydın Alatan, "Automatic target recognition and detection in infrared imagery under cluttered background," in *Target and Background Signatures III*. International Society for Optics and Photonics, 2017, vol. 10432, p. 104320J.

[3] He Deng, Xianping Sun, Maili Liu, Chaohui Ye, and Xin Zhou, "Infrared small-target detection using multiscale gray difference weighted image entropy," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 52, no. 1, pp. 60–72, 2016.

[4] Amanda Berg, *Detection and Tracking in Thermal Infrared Imagery*, Ph.D. thesis, Linköping University Electronic Press, 2016.

[5] DSIAC, "Atr algorithm development image database," .

[6] Abhijit Mahalanobis, Robert R Muise, S Robert Stanfill, and ALAN Van Nevel, "Design and application of quadratic correlation filters for target detection," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 40, no. 3, pp. 837–850, 2004.

[7] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[8] Abhijit Mahalanobis and Bruce McIntosh, "A comparison of target detection algorithms using dsiac atr algorithm development data set," in *Automatic Target Recognition XXIX*. International Society for Optics and Photonics, 2019, vol. 10988, p. 1098808.

[9] Brian Millikan, Hassan Foroosh, and Qiyu Sun, "Deep convolutional neural networks with integrated quadratic correlation filters for automatic target recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 1222–1229.

[10] Shuo Liu and Zheng Liu, "Multi-channel cnn-based object detection for enhanced situation awareness," *arXiv preprint arXiv:1712.00075*, 2017.

[11] Sergey Ioffe and Christian Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[12] Vinod Nair and Geoffrey E Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807–814.

[13] Ross Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[15] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al., "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.

[16] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[17] T. Tieleman and G. Hinton, "Lecture 6.5—RmsProp: Divide the gradient by a running average of its recent magnitude," COURSERA: Neural Networks for Machine Learning, 2012.