Infrared Target Detection in Cluttered Environments by Maximization of a Target to Clutter Ratio (TCR) Metric Using a Convolutional Neural Network

BRUCE MCINTOSH
SHASHANKA VENKATARAMANAN
ABHIJIT MAHALANOBIS
University of Central Florida Orlando, FL USA

Infrared target detection is a challenging computer vision problem which involves detecting small targets in heavily cluttered conditions 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. A TCR-network (TCRNet) is presented in which the filters of the first convolutional layer are composed of the eigenvectors most responsive to targets or to clutter. These vectors are analytically derived via a closed form optimization of the TCR metric. The remaining convolutional layers are trained using a novel cost function also designed to optimize the TCR criterion. We evaluate the performance of the TCRNet using a public domain medium wave infrared dataset released by the US Army's Night Vision Laboratories, and compare it to the state-of-the-art detectors such as Faster regions with convolutional neural networks (R-CNN) and Yolo-v3. The TCRNet 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 two when compared to these leading methods. Experimental results are shown for both day and night time images, and ablation studies are presented which demonstrate the contribution of the first layer eigenfilters, additional convolutional layers, and the benefit of the TCR cost function.

Manuscript received March 24, 2020; revised August 23, 2020; released for publication August 30, 2020. Date of publication September 17, 2020; date of current version February 9, 2021.

DOI. No. 10.1109/TAES.2020.3024391

Refereeing of this contribution was handled by K. P. Judd.

Authors' address: The authors are with the Center for Research in Computer Vision, Department of Computer Science, University of Central Florida, Orlando, FL 32816, USA, E-mail: (bwmcint@gmail.com; shashankv@knights.ucf.edu; amahalan@crcv.ucf.edu). *(Corresponding Author: Abhijit Mahalanobis.)*

0018-9251 © 2020 IEEE

I. 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 [1]–[4], IR target detection at acceptably low false alarm rates remains a difficult problem. Of course, there has been a tremendous advance 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.

An example of an IR image containing a target surrounded by natural background as well as a summary of the detection network is shown in Fig. 1. The location of the target in this image is indicated by the red box and the strongest prediction is indicated by the green x. As is evident, the target is relatively small and is difficult to find amidst the pronounced features of the terrain. For security, surveillance and reconnaissance applications, an automatic target detection system must cue the attention of a human operator to the location of such targets in the scene. To avoid operator fatigue, it is imperative to maintain a very low "false alarm" rate (also known as false positive rate), while detecting the actual targets with high probability. Unlike visible band cameras, IR sensors can be used in both day and night time conditions. However, the background clutter is significantly more pronounced during the day, which negatively impacts the false alarm rate. Different types of vegetation and landscape features can also give rise to challenging background clutter. Furthermore, the appearance of the targets also varies considerably due to changes in weather conditions, their operational state (e.g., engine ON or OFF,etc.), solar effects, and time of day. All of these factors make reliable target detection in high clutter a very challenging problem.

To facilitate research in target detection and recognition, a medium wave infrared (MWIR) dataset [5] was made available to the research community by the US Army's Night Vision and Electronic Sensors Directorate (NVESD). This dataset 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. Using this dataset, a comparison of a method known as the quadratic correlation filter (QCF) [6] and faster R-CNN [7], was conducted [8] with the latter exhibiting significantly better performance under relatively easy conditions (i.e., these early comparisons did not focus on performance under particularly challenging clutter conditions) . It was later observed that the results presented in this article were overly optimistic because of the way the data was collected.

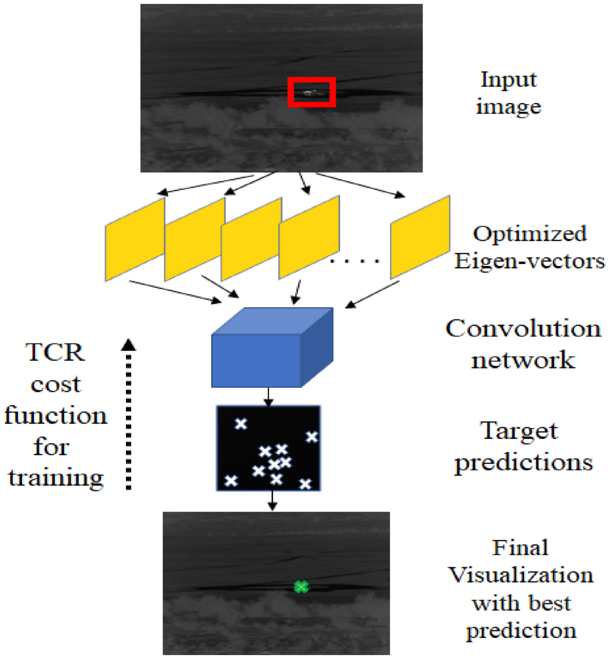

Fig. 1. Detecting small targets in cluttered background by maximizing target to clutter metric.

Specifically, the data consists of videos of targets moving in circles at different ranges. However, at any given range, the background scenery does not change because the camera was stationary. Therefore, although the images of the target class were different between testing and training, simply splitting the data does not adequately separate the background class (clutter) between the test and train sets. This issue is addressed in the current article by training on the images of targets at closer ranges, and testing on those at farther ranges. This ensures, that both the target and background are different, and the robustness of the algorithms is fairly evaluated. Millikan *et al.* [9] also proposed using the QCF filters as the first layer in a CNN for classifying the ten different types of targets in this dataset, but did not address target detection at long ranges and under difficult clutter conditions. In [10], Nasrabadi has shown target detection and recognition using conventional deep CNNs fail when applied to MWIR images in real-world scenarios, particularly when dealing with a limited number of data samples. To address this, they propose a fully convolutional network based on the VGG-16 framework (trained with the cross-entropy loss function) that maps the input forward looking infra red (FLIR) imagery data to a fixed stride correspondingly sized target score map. The potential targets are identified by applying a threshold on the target score map. Finally,the corresponding regions centered at these target points are fed to a second CNN to classify them into different target types while at the same time rejecting the false alarms. Other researchers [11] 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 dataset) and infrared imagery. Unfortunately methods that rely on target motion cannot find stationary (nonmoving) targets. In contrast, we address the detection of stationary targets in a single MWIR image frame when neither target motion nor other spectral bands are available.

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 R-CNN and Yolo-v3. Towards this end, we define a TCR metric and analytically derive the optimum eigenvectors that simultaneously represent targets and discriminate them from clutter. These eigenvectors are used as the filters in the first layer of a CNN. The rest of the network is then trained using a modified version of the TCR metric as the cost function. We refer to this hybrid structure as the TCR Network (or TCRNet) which is illustrated in Fig. 2.

Our principle contributions are as follows.

1) *Formulation and Optimization of a Target-to-Clutter Metric* for infrared target detection in the presence of strong clutter. Defined as a ratio of target and clutter energies produced by the network, the metric also emphasizes target representation while minimizing clutter response.
2) *Modified TCR Cost Function* for training a hybrid architecture where the first layer is fixed, but the rest of the convolutional layers are trained to maximally discriminate between targets and clutter.
3) *New State of the art.* In infrared target detection, TCRNet outperforms state-of-the-art methods on the MWIR dataset with over 30% improvement in probability of detection while reducing the false alarm rate by factor of two.

The rest of the article is organized as follows. In Section II we provide a brief overview of other previous research on infrared object detection using eigenanalysis, and CNNs that use precomputed and engineered features. The formulation of the TCR metric is described in Section III along with a description of the CNN used for maximizing it. This is followed by a derivation of the eigenfilters used in the first layer of the network and the modified TCR cost function used for training the rest of CNN layers. The NVESD dataset, and the training and testing protocols are described in Section IV. In Section V, we outline the details of training the TCRNet, Faster R-CNN, and Yolo-v3. The results of the experiments, performance analysis and comparisons, and ablation studies are given in Section VI. Finally, Section VII concludes this article.

II. OTHER RELATED RESEARCH

We now briefly review some of the prior relevant works on infrared target detection and recognition, including earlier techniques proposed for handling background clutter. In [12], Nasrabadi introduced a powerful method for detecting targets in infra-red images using a combination of a two-dimensional wavelet transform and a well-known multivariate anomaly detector called the Reed Xiaoli (RX) detector.

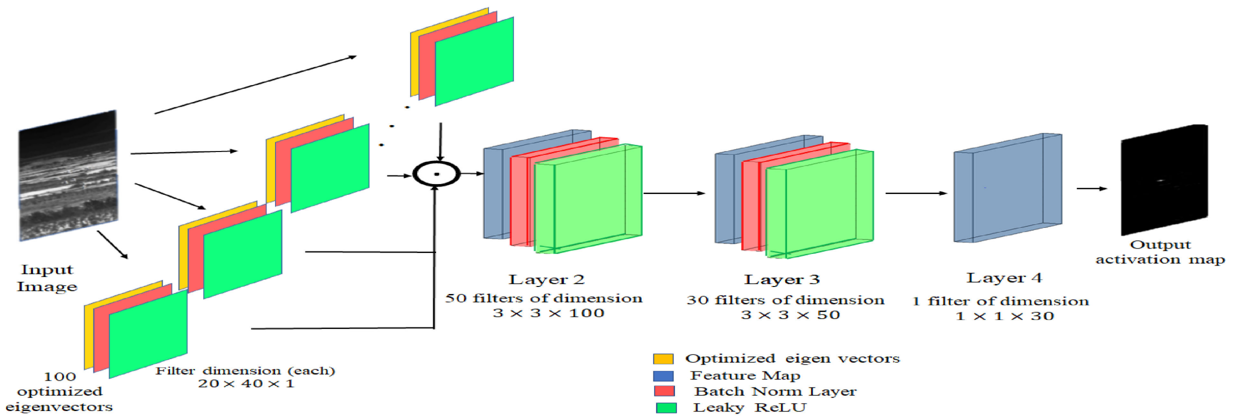

Fig. 2. Architecture of TCR Network. Layer 1 is analytically derived while the rest of the convolutional layers are iteratively learned to optimize the TCR metric.

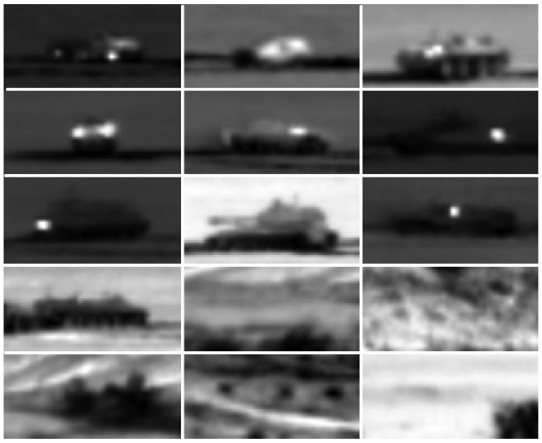

Fig. 3. Examples of the 20 × 40 chips used for training the first layer of the TCRNet. The ten vehicle classes are shown along with five examples of clutter or background chips. Similar images of size 40 × 80 are used for training the rest of the layers of the network.

Essentially, a 2-D wavelet transform is first applied to decompose the input image frame into uniform subbands, and the RX algorithm is then applied to the subband-image cube. Liu *et al.* [13] proposed to detect targets using eigenvectors obtained from a target map function computed for every location in the image, where large values indicate presence of target. Greenberg *et al.* [14] have proposed an automatic region of interest extractor to locate targets in infrared images without any knowledge of their size, orientation, etc. using localized texture and statistical features. Moradi *et al.* [15] utilize the average absolute gray scale difference combined with absolute directional mean difference for small object target detection in infrared images. Qin *et al.* [16] propose filtering of infrared images using facet kernel and random walker algorithm to enhance target pixels and suppress clutter. Bi *et al.* [17] use seven features such as direction, scale, gray-level intensity, spatial distribution, and variation while Qin *et al.* [18] have suggested the use of morphological component analysis based on sparse representation for target detection in heavy cluttered scenes. Han *et al.* [19] introduced a relative local contrast measure to enhance real targets while suppressing different types of interferences simultaneously, following which an adaptive threshold is used for target detection. Qin *et al.* [20] employ the difference of the Gaussian band-pass filter to enhance targets and suppress the background after which the salient map obtained from the novel local contrast measure is used to extract the target region. Demir *et al.* [21] classify image patches of very small aerial images with cloud or sea clutter using normalized correlation filters followed by a single fully connected decision layer. Hu *et al.* [22] deal with aerial views of small ground targets by combining static local binary pattern features and frame differencing features with a CNN. In [23], the authors have shown that precomputed and engineered features can be effectively used in CNNs to detect GAN generated fake images using a combination of cooccurrence matrices and deep learning. Song *et al.* [24] address the detection of land, air, and sea targets using patch similarity propagation for background estimation. Guan *et al.* [25] have shown that small targets can be detected in aerial images using Gaussian scale space, contrast enhancement, maxpooling, and adaptive threshold. Li *et al.* [26] also address the detection of land, air, and sea targets using nonlearning techniques based on local and nonlocal priors. Huang *et al.* [27] have also proposed nonlearning methods for aerial object detection using directional difference of Gaussian filters. Finally d'Acremont *et al.* have described the use of synthetically generated and real infrared data to train CNNs to classify region proposals.

III. TCR NETWORK AND OPTIMIZATION

As shown in Fig. 2, 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 first layer are then post processed by two successive convolutional layers with fifty 3 × 3 × 100 and thirty 3 × 3 × 50 filters, respectively. Each of these layers is preceded by batch normalization [28] and rectified linear unit (ReLU) [29]. 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.

A. TCR Metric and Derivation of Layer 1 Filters

Consider the energy of the projection of an image vector x on a set of M basis vectors q_i given by

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

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 (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 (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}. This observation is based on the fact that the numerator is a product of quadratic terms, making it large by maximizing the Raleigh quotient will require all the terms to be as large as possible. In other words, none of terms in the product in the numerator can be small or zero. In contrast, in conventional formulation (such as the traditional Fisher metric) the numerator is a sum of quadratic terms. This does not always ensure that every term is large (since a sum of positive terms can be large if any of them is large).

Interestingly, the eigenvectors of $R_2^{-1} R_1$ simultaneously diagonalize both R_1 and R_2. In other words, if $Q = [q_1, q_2 ... q_N]^T$, then it can be shown that $QR_2Q^T = I$, and $QR_1Q^T = \Delta$, where I is the identity matrix, and Δ is a diagonal matrix with elements $\gamma_i = q_i^T R_1 q_i$. Based on this, the new metric in (3) can be also written as

$$J_{TCR} = \frac{|Q^T R_1 Q|}{Tr(Q^T R_2 Q)} \tag{6}$$

where $Tr(.)$ represents the trace of the matrix. Therefore maximizing the ratio of the determinant of $Q^T R_1 Q$ to the trace of $Q^T R_2 Q$ is equivalent to maximizing J_{TCR}.

B. 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. A criteria for choosing the number of eigenvectors M to be used as filters in the first layer will be discussed in Section V. In general, the classes will be better separated by the few dominant eigenvectors which will ensure fewer false positives. However fewer eigenvectors will also lead to poorer representation of each class, leading to missed detections of true targets. As M is increased, the dimensionality of the space represented by the eigenvectors increases, but quadratic classifiers (such as the original QCF) can no longer provide good discrimination.

To alleviate this problem, we process the outputs of the eigenfilters (i.e., the output of the first layer) using several convolutional layers and nonlinear activations. In other words, the set of eigenvectors obtained in (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

$$J'_{TCR} = \frac{\frac{1}{N}\sum y_i^T y_i}{\sqrt[N]{\prod x_i^T x_i}} \tag{7}$$

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 (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 cost function 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 the following:

The derivative of this function with respect to each class is

$$\begin{aligned}\log(J'_{TCR}) &= -\log(N) + \log\left(\sum y_i^T y_i\right) - \frac{1}{N}\log\left(\prod x_i^T x_i\right)\\ &= -\log(N) + \log\left(\sum y_i^T y_i\right) - \frac{1}{N}\sum \log\left(x_i^T x_i\right).\end{aligned} \tag{8}$$

The derivative of this function with respect to each class is

$$\begin{aligned}\nabla_{y_i}\log(J'_{TCR}) &= \frac{2y_i}{\sum y_i^T y_i}\\ \nabla_{x_i}\log(J'_{TCR}) &= -\frac{1}{N}\frac{2x_i}{x_i^T x_i}.\end{aligned} \tag{9}$$

Therefore, as training images are presented to the network during the learning process, the gradient supplied to the back-propagation 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.

IV. DATASET, TRAINING, AND TESTING PROTOCOLS

The dataset is a collection of visible and MWIR imagery collected by NVESD that is intended to support the automatic target recognition (ATR) algorithm development community. It contains 207 GB of MWIR imagery and 106 GB of visible imagery along with ground truth and environmental information. Target classes included humans, military, and civilian vehicles. The data were collected during day and night and at range increments of 500m from 500 to 5000 m. The targets moved at constant velocity along a circular track with an approximate diameter of 100 m.

We used the MWIR images between the ranges of 1000 and 3500 m for our tests and evaluations. The size of these images are 512 × 640, with 16-integer pixel values. In earlier experiments reported in [8], different images from the same ranges were used for training and testing. While this is sufficient to ensure that the training and test views of the targets are different, the results were overly optimistic because the sensor was stationary and the background scenery is largely the same in all images obtained at a particular range. Thus the networks were inadvertantly trained and evaluated on the same background clutter.

In order to adequately decorrelate the training and test sets, we create a training set based only on images at the 1000, 1500, and 2000 m ranges while the testing set comprises of images at 2500, 3000, and 3500 m. This results in train and test images with substantially different backgrounds and makes the detection problem more realistic. The training and testing sets are both composed of images sampled at uniform intervals as the target moves around the track, so that the target is seen at various view angles. Each set contains 9720 images (8640 with single targets, and 1080 with two or more targets) with an even balance of target types, ranges, and day/night scenarios. The exact number of images in the training and test sets are shown in the tables in Table I.

TABLE I
Summary of the Experimental Settings used to Evaluate TCRNet on the NVESD Dataset

	Range (in k.m)	Time	Number of classes	Total samples
Training	1.0 - 2.0	Day & night	10	8640 s & 1080 m
Testing	2.5 - 3.5	Day & night	10	8640 s & 1080 m

Acronyms : s: Single Target m: Multiple Targets in an Image.

Each image in the dataset has rich metadata which provides the distance to the target, their orientation of the targets with respect to the sensor, time of day, atmospheric conditions, and so forth. In our experiments, the range information was used to scale the size of the image by a factor s, such that the targets always appear to be at a distance of 2500 m in all cases. The scale factor was computed simply as $s = d/2500$, where d is the distance to the target in a given frame.

V. EXPERIMENTS AND PERFORMANCE ANALYSIS

In this section, we evaluate the performance of the TCR-Net and compare it to that of Faster R-CNN and Yolo-v3. The various networks are trained in their own particular way as described below, but tested in the same way on full images from the test set. Detection is a two class problem, so although some networks provide a classification result, it is not considered, i.e., a detection with an incorrect classification will be treated as a true positive if it is close enough to the ground truth location. For a prediction to be treated as a correct detection, its location was required to be within 20 pixels of the ground truth center. Results are presented in the form of receiver operating characteristic (ROC) curves that show probability of detection (P_d) as a function of false alarm rate (FAR). 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

$$\text{FAR} = \frac{\text{Total number of false positives}}{\text{Total number of frames} \times \text{FOV}} \tag{10}$$

where FOV is the product of the horizontal and vertical fields of view of the sensor. The ROC curves are shown for the entire test set, and also separately for the day and night conditions.

A. TCR Network

The TCR Network is composed of four layers as shown in Fig. 2. The first layer has 100 20 × 40 filters, which are set

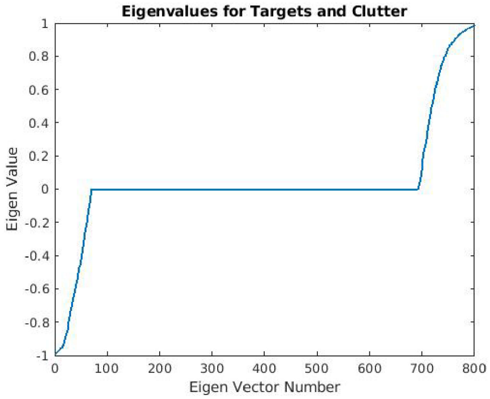

Fig. 4. This figure shows the eigenvalues for the 800 eigenvectors. Vectors with positive eigenvalues show a stronger response for targets while those with negative values respond strongly to clutter. The most responsive eigenvectors are chosen to be the filters in the first layer of the network.

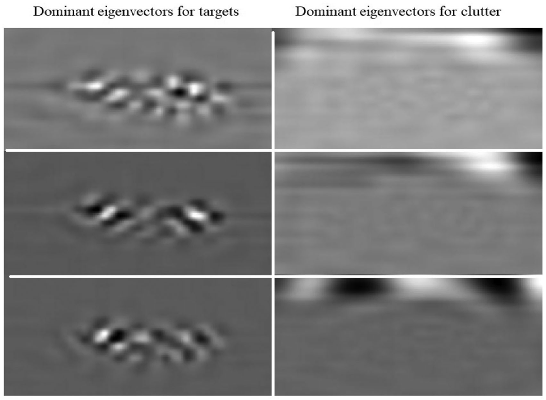
Dominant eigenvectors for targets
Dominant eigenvectors for clutter

Fig. 5. This figure visualizes the three most dominant eigenvectors for detecting targets (left) and clutter(right).

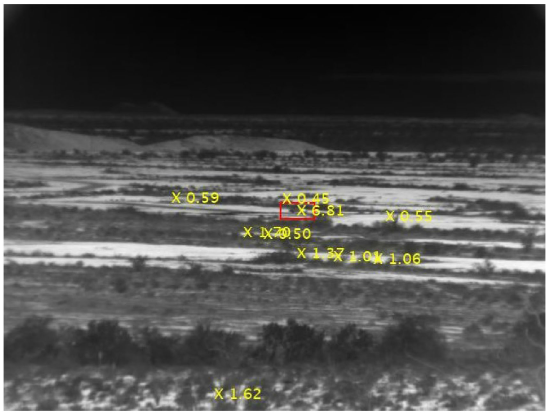

Fig. 6. Detection responses for the TCR Network. The x's mark the detections along with the response value. The response that falls within the ground truth bounding box has a value of 6.81 which is substantially higher than any of the other predictions.

as eigenfilters as described below, and remain fixed during training. This layer is followed by two CNN layers with filters of dimensions 3 × 3 × 50 and 3 × 3 × 30 that are randomly initialized. The output of all these layers goes through batch normalization and uses a Leaky ReLU for nonlinearity. The final stage is a convolution with a single 1 × 1 × 30 filter.

The optimum eigenfilters for the first layer are created as described in Section III.A, 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 20 × 40 pixel region of the image is cropped out and treated as a "clutter" or background example. Examples of some of the target and clutter training chips are shown in Fig. 5. 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 (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 difficult to choose suitable eigenvectors. For this reason, the eigenvalues γ_i are remapped into a range $[-1, +1]$ as

$$\lambda_i = \frac{\gamma_i + 1}{\gamma_i - 1}. \tag{11}$$

The value of λ_i now ranges between $[-1, +1]$. Fig. 4 shows these remapped eigenvalues of the TCR basis set. The eigenvectors that correspond to positive values of λ_i in this plot contain more target information than clutter, while the ones corresponding to the negative values of λ_i are more representative of clutter. Furthermore, the eigenvectors corresponding to $\lambda_i = 0$ contain no useful information at all, and can be readily discarded. The images of the top three "target-specific" eigenfilters are shown in the left column in Fig. 5. These correspond to the three largest values of λ_i in Fig. 4. It is interesting to note the features contained in these eigenfilters are "target-like" in appearance, and occur in the middle of the filter. Similarly, the three most "clutter specific" eigenfilters are shown on the right column in Fig. 5, and correspond to the three most negative values of λ_i. Unlike the target specific ones, these eigenfilters appear to contain a void in the middle with larger values along the boundaries of the filter. This is due to the fact that clutter surrounds the boundaries of the targets, but does not overlap with it.

Based on the values of λ_i in Fig. 6, 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.

This network was implemented, trained, and tested in Matlab. Holding the first layer as constant and allowing the others to learn, the network was trained with image chips of size 40 × 80, which are otherwise similar to those in Fig. 5 (i.e., they contained either a target in the center or clutter). The cost function described in (9) was used for training over 25 epochs with the RMSprop optimizer [30], a batch size of 100, and initial learning rate of $1e^{-5}$.

B. Faster-R-CNN, Yolo-v3, and Wavelet RX

The number of training images available in the dataset is too few to train deep networks from scratch. Therefore we use transfer learning to adapt state-of-the-art methods to the IR target detection problem. The faster-R-CNN [31] uses a Resnet-50 [32] pretrained on Imagenet [33] as its backbone and finetuned on our dataset. We evaluate this network in MATLAB with the full sized (512 × 640) images (not image chips) using the four step training procedure. The training data are gray scale and therefore single channel, while the pretrained faster-R-CNN was based on three channel RGB data. To make this compatible, the training image was replicated into each of the three channels. It was run for 50 epochs with a batch size of one and learning rate of $1e^{-3}$

We also evaluate Yolo-v3 [34] and compare its performance with our proposed TCRNet algorithm. For Yolo-v3, we use the weights pretrained on MS-Coco [35], and finetune it on our dataset. Pytorch [36] was used to implement our model using the original 512 × 640 image size. Similar to the preprocessing techniques used in Faster-R-CNN, the input images to the network were replicated to make it into three channels. We trained the model using a learning rate of $1e^{-3}$ for total of 100 epochs with batch size of four.

The wavelet RX algorithm does not require any training, and was implemented following the method described in [12]. A four stage decomposition was done on each input image using *Daubechies wavelets* to produce a block of sixteen subband images. The RX detector [37] finds anomalies in the scene by computing the *Mahalanobis distances* of the feature vector at each pixel with respect to the nominal background, where the features are the values in the different subband images at the same pixel. Thus the wavelet RX algorithm is also a quadratic detector, and serves as a baseline for performance comparison in our experiments.

VI. RESULTS AND PERFORMANCE ANALYSIS

As indicated in Section IV, all algorithms were trained on images at ranges of 1, 1.5, and 2.0 km, and tested on images at ranges of 2.5, 3.0, and 3.5 km. For our experiments, we had 10 800 training images of targets, and 10 800 training images of clutter, each. Since it is not possible to train very deep networks from scratch on such relatively small datasets, we used our dataset to finetune pretrained versions of Faster R-CNN and YOLO. Therefore, any potential benefit of the extensive training of these networks on the standard larger datasets (such as ImageNet and MS-COCO) is built into the pretrained version. It should be noted that the mini batch size determines the number of iterations per epoch. All training images are presented to the algorithm during an epoch, whether it is in batch sizes of 1, 4, or 100. Therefore all three algorithms learn from all training images during an epoch. However, the number of epochs and the size of the minibatch (as well as other hyper parameters) were selected to get the best possible performance from each network model. We used 9750 images (of size 512 × 640) for testing in total. 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 2.5 km. The entire image is input to the TCRNet, and the filters of the first layer are convolved across the entire 512 × 640 test image. Since there are 100 such filters, the output of the first layer is a 512 × 640 × 100 dimensional tensor. This is the input to the upper CNN layers of the TCRNet. Examples of the detections produced by the TCRNet in a day-time MWIR image are shown in Fig. 6. A yellow “x” indicates where potential targets may be located in the scene. The numerical score associated with each detection is indicated at each location. The red box marks the ground truth location of the target in the scene. It is noteworthy that the correct detection within the truth box has a score which is an order of magnitude greater than that of the false detections elsewhere in the scene. Thus, the majority of the false detections are readily eliminated by thresholding the detection score. When tested on the entire test set including day and night time images, the TCRNet shows the best performance with substantial margin of 14, 29, and 9% over Yolo-v3, Faster-R-CNN, and the Wavelet RX algorithms, respectively. The ROC curves are shown in Fig. 7. Qualitatively, the TCRNet not only detects the most number of targets but also achieves the highest probability of detection at the lower false alarm rates on the left side of ROC curves.

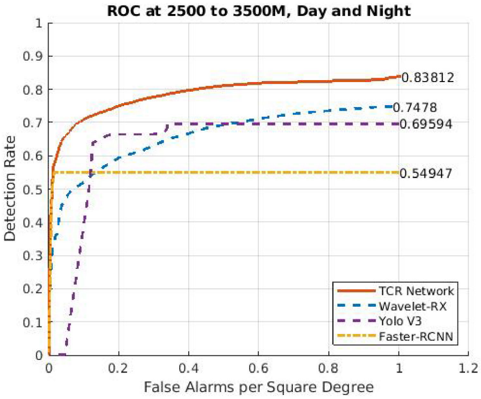

Fig. 7. ROC curves for the three networks on the full test set (day and night). The TCR Network shows the best performance with a substantial margin of 14% , 29%, and 9% over Yolo-v3, Faster-R-CNN and wavelet RX, respectively.

TABLE II
FAR Comparison

	max P_d	FAR	TCR FAR
Faster-RCNN	0.549	0.015	0.012
Yolo-v3	0.695	0.339	0.08
Wavelet-RX	0.748	0.943	0.197

The TCR Network shows a substantially lower false alarm rate. The first two columns show the maximum detection rate and associated false alarm rate for each detector. The third column shows the false alarm rate for the TCR network for the given detection rate.

Table II compares the false alarm rate of the TCRNet to that of the other approaches at the same detection scores. In fact, the max P_d column shows the highest detection achieved by each algorithm while the FAR column shows their corresponding false alarm rate. The last column shows

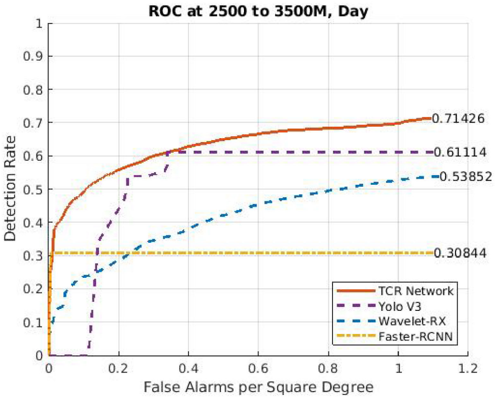

Fig. 8. ROC curves for the three networks on the more challenging day time images. The TCR Network shows the best performance with a substantial margin of 10, 41, and 17% over Yolo-v3, Faster-R-CNN and wavelet RX respectively.

the false alarm rate for the TCR Network at the same detection rate. This shows that at the maximum detection of 54.9% achieved by the Faster R-CNN (which is the lowest among all the algorithms), the TCRNet is slightly better in terms of false alarm rate. However, at the maximum detection score of 69.5% for Yolo-v3, the TCRNet has a significantly lower false alarm rate of 0.08 versus 0.34FA/sq degrees. The same is also true for the wavelet RX algorithm, where the TCRNet achieves a false alarm rate of 0.197 versus 0.94 FA/sq at a detection value of 78%.

In infrared images, the background and clutter intensity is often greater during the day time. Thus detection of targets in daytime clutter is the most challenging detection problem. In contrast, targets with hot engines are relatively easy to find during the night time. From the same test described above, the results for the day time images were separated out and the ROC curve is shown in Fig. 8. The TCRNet shows the best performance with a margin of 10, 41, and 15% over Yolo-v3, Faster-RCNN, and the wavelet RX algorithms, respectively. Notably, the Yolo-v3 network in unable to find any target below a false alarm rate of 0.1 Fa/sq degree. It is clear that these object detection algorithms struggle with finding the relatively small targets in a cluttered background.

For completeness the results for night time are also shown in Fig. 9. In this comparatively easier scenario the TCRNet achieves near perfect results with 97% accuracy. Although their performances improve compared to the day-time, the Yolo-v3 and Faster-RCNN still perform poorer than the TCRNet even at night, with final detection scores of 80%. Interestingly under these easier conditions, the wavelet RX method achieves the same final detection as the TCRNet (i.e., 97%) at the right end of the ROC curve, but is outperformed by the TCRNet at lower false alarm rates below 0.4 FA/sq degree.

A. Ablations

In this section we explore the importance of the various aspects of the network to understand their contribution to its performance. The first test was to replace the custom TCR

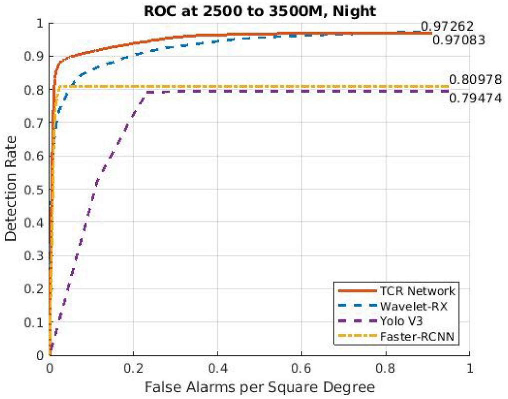

Fig. 9. ROC curves for the three networks on the less challenging night time images. The TCR Network achieves 97% with a margin of 17% over Yolo-v3 and Faster-RCNN, respectively. It also outperforms the Wavelet RX method at false alarm rates below 0.4 FA/sq dergee.

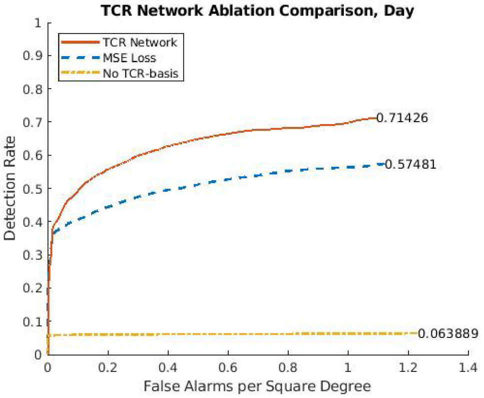

Fig. 10. Ablation testing on day time images. The mse loss plot shows the result when the TCR Network's cost function is replaced with mean squared error loss. The *No TCR-basis* plot shows the result for the TCR Network if the first layer filters are learned instead of loaded with the eigenfilters.

loss function in (8) with a conventional mean square error loss function. The ideal response is specified to be a narrow Gaussian blur centered at the ground-truth location of the target, and zero everywhere else. For the challenging day time images, a large degradation in performance can be seen in Fig. 10. This shows that under challenging conditions, the optimization of the TCR metric substantially improves the TCRNets ability to detect targets in difficult clutter. For the easier night time images, both losses work very well as shown in Fig. 11.

Also shown on these plots is the importance of the eigenfilters in the first layer of the network, which optimize the TCR metric in (3) or (6). The yellow lines show the results when these filters are removed and the entire network is trained from scratch. Instead of loading these filters and freezing them, the first layer is initialized randomly and the network is allowed to learn these weights using the TCR cost function gradients in (9). Although the night time performance is somewhat better than in daytime conditions, it is clear that the analytically derived eigenfilters are essential for achieving the results shown in Fig. 7.

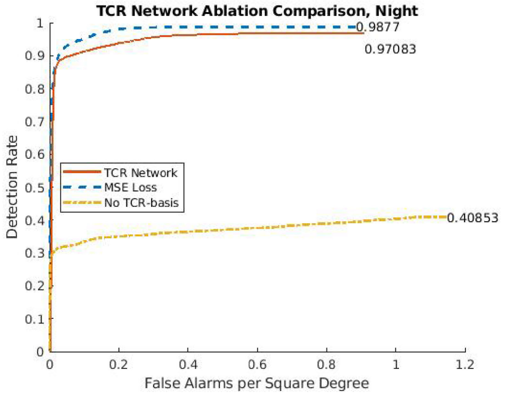

Fig. 11. Ablation testing on night time images. The mse loss plot shows the result when the TCR Network's cost function is replaced with mean squared error loss. The No QCF plot shows the result for the TCR Network if the first layer filters are learned instead of loaded with the eigenfilters.

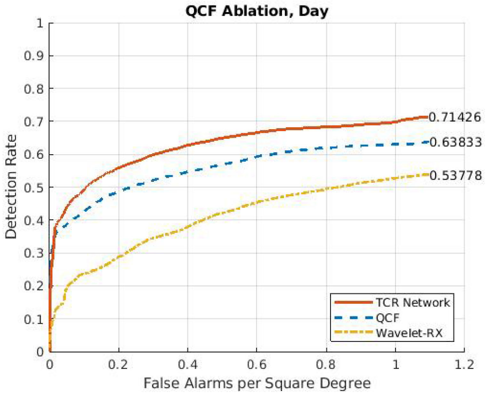

Fig. 12. Ablation testing on day time images. The QCF plot shows the result if only the eigenfilters and a quadratic statistic are used without the later learned layers.

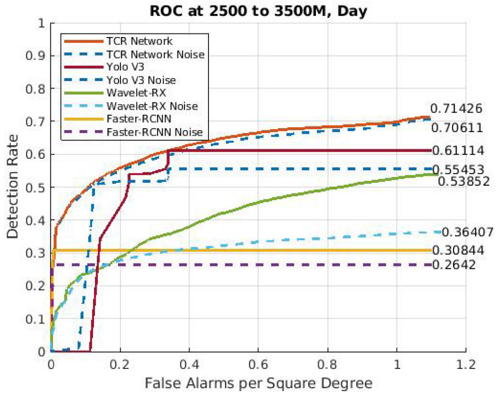

Fig. 13. Noise Comparison. This shows the response of the different detectors when Gaussian noise is added to each test image so that the signal to noise ratio is 8. It can be seen that the TCR network is affected less by noise.

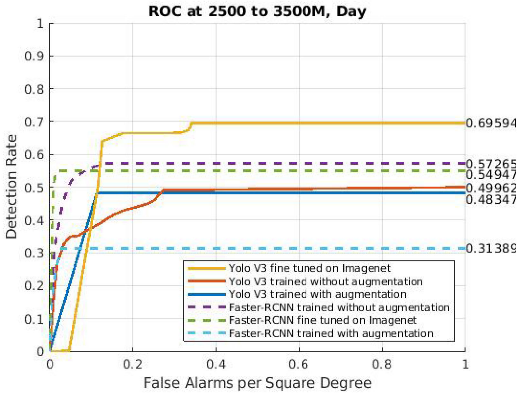

Fig. 14. This figure shows the effect of using transfer learning from the RGB domain. Fine tuning was performed for 10 epochs on networks pretrained on Imagenet images. The other curves show the results of training from scratch for 20 epochs with and without simple data augmentation.

To see the effect of upper layers of the network, Fig. 12 compares the performance of the original QCF, the TCR-Network, and the wavelet RX under day time conditions. The blue dashed line is obtained by directly computing the statistic in (1) using the output of the eigenfilters in Layer 1. Since this is a quadratic metric, this is akin to target detection using a QCF. Once again, we see that under challenging day time conditions, the TCRNet performs better, whereas the two were found to be comparable in night conditions. Specifically, the TCRNet detects 71.4% whereas the QCF metric detects 64.3%. At a detection rate of 64.3%, the QCF metric achieves a false alarm rate of 1.15 FA/sq degree, whereas the TCRNet achieves the same detection at 0.45 FA/sq degree. In other words, the false alarm rate of the TCRNet is less than one-half (1/2) that of the QCF metric in heavy clutter conditions. We also see that under these conditions, the wavelet RX is outperformed by both the basic QCF and the TCRNet.

In Fig. 13, we study the robustness of the TCRNet to additive white Gaussian noise (AWGN). It is anticipated that using eigenfilters in the first layer of the network will have a "denoising" effect. For this experiment, noise was added to each input image, such that the *signal-to-noise ratio* (SNR) (defined as the ratio of the signal and noise standard deviations) was equal to 8. The figure shows the ROC curves for all four methods, both with and without noise. It is evident that the ROC curve of the TCRNet is the least perturbed by the noise, while those of the FasterRCNN, Yolo-v3, and the Wavelet RX are more noticeably affected.

Finally, in Fig. 14, we compare three different methods for training the Faster-RCNN and Yolo-v3 networks. In addition to finetuning networks pretrained on ImageNet, we also trained each network from scratch on the infrared data, both with and without data augmentation. We see that the best results were obtained using the pretrained networks fine tuned on the infrared data. We therefore used this approach for training the FasterRCNN and Yolo-v3 in all other experiments reported in the article.

VII. CONCLUSION

The paradigm for contemporary deep CNNs is highly data driven. The availability of large amounts (perhaps millions) of training data is necessary to robust learn the features. To date, most of the large datasets use RGB images. However, this is not always available in some applications infrared imagery for surveillance in natural environments (or in the medical field). Our method shows, that analytically deriving the first layer when limited data is available, and then training the upper layers using the same cost function is better than fine-tuning pretrained deep networks.

The TCR Network proposed in this article 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 first layer of the network uses analytically derived eigenfilters, while the later layers are learned via gradient descent. 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. The TCRNet's performance was evaluated using the MWIR image dataset 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, in both day and night conditions. 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, and the baseline QCF. Several ablation studies were also conducted which show that the Layer 1 eigenfilters in combination with the modified TCR cost function are the most effective in combating daytime clutter.

REFERENCES

[1] J. A. Ratches
Review of current aided/automatic target acquisition technology for military target acquisition tasks
Opt. Eng., vol. 50, no. 7, 2011, Art. no. 072001.

[2] E. Gundogdu, A. Koç, and A. A. Alatan
Automatic target recognition and detection in infrared imagery under cluttered background
in *Target and Background Signatures III*, Bellingham, WA, USA: SPIE, 2017, vol. 10432, Art. no. 104320J.

[3] H. Deng, X. Sun, M. Liu, C. Ye, and X. Zhou
Infrared small-target detection using multiscale gray difference weighted image entropy
IEEE Trans. Aerosp. Electron. Syst., vol. 52, no. 1, pp. 60–72, Feb. 2016.

[4] A. Berg
Detection and tracking in thermal infrared imagery
Ph.D. dissertation, Dept. Elect. Eng., Linköping Univ., Linköping, Sweden, 2016.

[5] *ATR algorithm development image database*, DSIAC. [Online]. Available: https://www.dsiac.org/resources/research-materials/cds-dvds-databases-digital-files/atr-algorithm-development-image

[6] A. Mahalanobis, R. R. Muise, S. R. Stanfill, and A. Van Nevel
Design and application of quadratic correlation filters for target detection
IEEE Trans. Aerosp. and Electron. Syst., vol. 40, no. 3, pp. 837–850, Jul. 2004.

[7] S. Ren, K. He, R. Girshick, and J. Sun
Faster R-CNN: Towards real-time object detection with region proposal networks
In *Advances neural information processing systems*, 2015, pp. 91–99.

[8] A. Mahalanobis and B. McIntosh
A comparison of target detection algorithms using DSIAC ATR algorithm development data set
Proc. SPIE, vol. 10988, 2019, Art. no. 1098808.

[9] B. Millikan, H. Foroosh, and Q. Sun
Deep convolutional neural networks with integrated quadratic correlation filters for automatic target recognition
In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2018, pp. 1222–1229.

[10] N. Nasrabadi
Deep Target: An automatic target recognition using deep convolutional neural networks
vol. 55, no. 6, pp. 2687–2697, 2019.

[11] S. Liu and Z. Liu
Multi-channel CNN-based object detection for enhanced situation awareness
2017, *rXiv:1712.00075*.

[12] A. Mehmood and N. M. Nasrabadi
Wavelet-RX anomaly detection for dual-band forward-looking infrared imagery
Appl. Opt., vol. 49, no. 24, pp. 4621–4632, 2010.

[13] R. Liu, E. Liu, J. Yang, T. Zhang, and Y. Cao
Point target detection of infrared images with eigentargets
Opt. Eng., vol. 46, no. 11, 2007, Art. no. 110502.

[14] S. Greenberg, S. R. Rotman, H. Guterman, S. Zilberman, and A. Gens
Region-of-interest-based algorithm for automatic target detection in infrared images
Opt. Eng., vol. 44, no. 7, 2005, Art. no. 077002.

[15] S. Moradi, P. Moallem, and M. F. Sabahi
Fast and robust small infrared target detection using absolute directional mean difference algorithm
Signal Process., vol. 177, 2020, Art. no. 107727.

[16] Y. Qin, L. Bruzzone, C. Gao, and B. Li
Infrared small target detection based on facet kernel and random walker
IEEE Trans. Geosci. Remote Sens., vol. 57, no. 9, pp. 7104–7118, Sep. 2019.

[17] Y. Bi, X. Bai, T. Jin, and S. Guo
Multiple feature analysis for infrared small target detection
IEEE Geosci. Remote Sens. Lett., vol. 14, no. 8, pp. 1333–1337, Aug. 2017.

[18] H. Qin *et al.*
Infrared small moving target detection using sparse representation-based image decomposition
Infrared Phys. Technol., vol. 76, pp. 148–156, 2016.

[19] J. Han, K. Liang, B. Zhou, X. Zhu, J. Zhao, and L. Zhao
Infrared small target detection utilizing the multiscale relative local contrast measure
IEEE Geosci. Remote Sens. Lett., vol. 15, no. 4, pp. 612–616, Apr. 2018.

[20] Y. Qin and B. Li
Effective infrared small target detection utilizing a novel local contrast method
IEEE Geosci. Remote Sens. Lett., vol. 13, no. 12, pp. 1890–1894, Dec. 2016.

[21] H. S. Demİr and E. Akagündüz
Filter design for small target detection on infrared imagery using normalized-cross-correlation layer
Turkish J. Elect. Eng. Comput. Sci., vol. 28, no. 1, pp. 302–317, 2020.

[22] X. Hu, X. Wang, X. Yang, D. Wang, P. Zhang, and Y. Xiao
An infrared target intrusion detection method based on feature fusion and enhancement
Defence Technol., vol. 16, no. 3, pp. 737–746, 2020.

[23] L. Nataraj *et al.*
Detecting GAN generated fake images using co-occurrence matrices
Electron. Imag., vol. 2019, Jan. 2019, pp. 532–1–532–7.

[24] Q. Song, Y. Wang, K. Dai, and K. Bai
Single frame infrared image small target detection via patch similarity propagation based background estimation
Infrared Phys. Technol., vol. 106, 2020, Art. no. 103197.

[25] X. Guan, Z. Peng, S. Huang, and Y. Chen
Gaussian scale-space enhanced local contrast measure for small infrared target detection
IEEE Geosci. Remote Sens. Lett., vol. 17, no. 2, pp. 327–331, Feb. 2019.

[26] W. Li, M. Zhao, X. Deng, L. Li, L. Li, and W. Zhang
Infrared small target detection using local and nonlocal spatial information
IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens., vol. 12, no. 9, pp. 3677–3689, Sep. 2019.

[27] S. Huang, M. Li, X. Wang, X. Zhao, L. Yang, and Z. Peng
Infrared small target detection with directional difference of Gaussian filter
In *Proc. 3rd IEEE Int. Conf. Comput. Commun.*, 2017, pp. 1698–1701.

[28] S. Ioffe and C. Szegedy
Batch normalization: Accelerating deep network training by reducing internal covariate shift
2015, *arXiv:1502.03167*.

[29] V. Nair and G. E. Hinton
Rectified linear units improve restricted Boltzmann machines
In *Proc. 27th Int. Conf. Mach. Learn.*, 2010, pp. 807–814.

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

[31] R. Girshick
Fast R-CNN
In *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1440–1448.

[32] K. He, X. Zhang, S. Ren, and J. Sun
Deep residual learning for image recognition
In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[33] O. Russakovsky *et al.*
Imagenet large scale visual recognition challenge
Int. J. Comput. Vis., vol. 115, no. 3, pp. 211–252, 2015.

[34] J. Redmon and A. Farhadi
YOLOv3: An incremental improvement
2018, *arXiv:1804.02767*.

[35] T.-Y. Lin
et al. Microsoft COCO: Common objects in context
In *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 740–755.

[36] A. Paszke *et al.*
Automatic differentiation in PyTorch
In *NIPS Autodiff Workshop*, 2017. [Online]. Available: https://openreview.net/forum?id=BJJsrmfCZ

[37] I. S. Reed and X. Yu
Adaptive multiple-band CFAR detection of an optical pattern with unknown spectral distribution
IEEE Trans. Acoust., Speech, Signal Process., vol. 38, no. 10, pp. 1760–1770, Oct. 1990.

Bruce McIntosh received the B.S. degree in electrical engineering from the University of Alabama, Huntsville, AL, USA, in 1987 and the M.S. degree in computer science from the University of Central Florida, Orlando, FL, USA, in 2020. He is currently working toward the Ph.D. degree in methods for target detection from the Center for Research in Computer Vision (CRCV) with the University of Central Florida (UCF), Orlando, FL, USA, under the advisement of Dr. Mahalanobis.

His research interests include computer vision, deep learning, and target detection.

Shashanka Venkataramanan received the B.E. degree in electrical engineering from the University of Mumbai, Mumbai, India, in 2017 and the M.S. degree in computer science from the University of Central Florida, Orlando, FL, USA, in 2020. He is currently working toward the Ph.D. degree in network compression for efficient object detection with the LinkMedia team at INRIA Rennes - Bretagne Atlantique, Rennes, France, under the supervision of Dr. Yannis Avrithis.

His research interests include computer vision and developing compressed models for accelerated deep learning.

Abhijit Mahalanobis received the B.S. degree (Hons.) from the University of California, Santa Barbara, CA, USA, in 1984, the M.S. and Ph.D. degrees in electrical and computer engineering from the Carnegie Mellon University, Pittsburgh, PA, USA, in 1985 and 1987, respectively.

He is currently an Associate Professor with the Center for Research in Computer Vision (CRCV) with the University of Central Florida (UCF), Orlando, FL, USA. Prior to joining UCF, he was a Senior Fellow with the Lockheed Martin, Orlando, FL, USA. He has also worked previously with Raytheon, Tucson, and was a Faculty with the University of Arizona, Tucson, AZ, USA, and the University of Maryland, College Park, MD, USA. He has over authored or coauthored 170 journal and conference publications. He also holds four patents, coauthored a book on pattern recognition, contributed several book chapters, and edited special issues of several journals. His research interests include video/image processing for target detection and recognition, computer vision, and computational imaging.

Dr. Mahalanobis was elected a fellow of SPIE and OSA, in 1997 and 2004, respectively, for his work on optical pattern recognition and automatic target recognition. He was also recognized as the 2006 Scientist of the Year by Science Spectrum Magazine, a publication of the Career Communication Group, Inc. He was the recipient of the 2006 Scientist of the Year by Science Spectrum Magazine, a publication of the Career Communication Group, Inc. He served as an Associate Editor for *Applied Optics* from 2004 to 2009. He was as an Associate Editor for the *Journal of the Pattern Recognition Society* from 1994 to 2003. He served on OSA's Science and Engineering council in the capacity of Pattern Recognition Chair from 2001 to 2004, and as Technical Group Chair for Information Acquisition, Processing and Display on OSA's Board of Meetings from 2012 to 2015. He also serves on the organizing committees for the SPIE conferences and OSA's annual and topical meetings.