IMPROVED TARGET DETECTOR FOR FLIR IMAGERY

Lipchen Alex Chan, Sandor Z. Der, and Nasser M. Nasrabadi

U.S. Army Research Laboratory, Attn: AMSRL-SE-SE
2800 Powder Mill Road, Adelphi, MD 20783, USA

ABSTRACT

Algorithms are considered for searching wide area forward-looking infrared imagery for military vehicles. Wide area search has typically been handled by using a simple detection algorithm with low computational cost to search the entire image or set of images, followed by a clutter rejection algorithm that analyzes only those portions of the image that are marked by the detection algorithm. We start with a feature based detector and a eigen-neural based clutter rejecter, and examine a number of architectures for combining these modules to maximize joint performance. The architectures considered include a clutter rejection threshold method and a nonlinear learning-based combination. The performance of the architectures are compared using a set of several thousand real images.

1. INTRODUCTION

Empowered by the advances in computer capability and image processing technology, automatic target recognition (ATR) systems are becoming an essential part of many imaging systems. The development of robust ATR systems must still overcome a number of well-known challenges, including the large number of target classes and aspects, long and varying viewing range, obscured targets, cluttered backgrounds, various geographic and weather conditions, sensor noise, and variations caused by translation, rotation, and scaling of the targets. For the experiments presented in this paper, the input images were obtained by forward-looking infrared (FLIR) sensors. For these sensors, the signatures of the targets within the scene are severely affected by rain, fog, and foliage [1]. Two of the sharper images in our data set are shown in Figure 1, with different fields of view, viewing ranges, and number of targets.

This research was sponsored by the Army Research Laboratory (ARL) and was accomplished under the ARL/ASEE postdoctoral fellowship program, contract DAAL01-96-C-0038. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes not withstanding any copyright notation hereon.

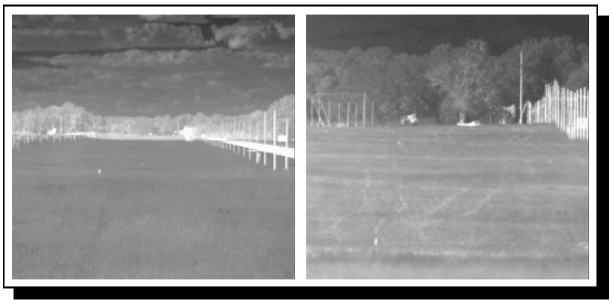

Fig. 1. A pair of FLIR scenes with different characteristics.

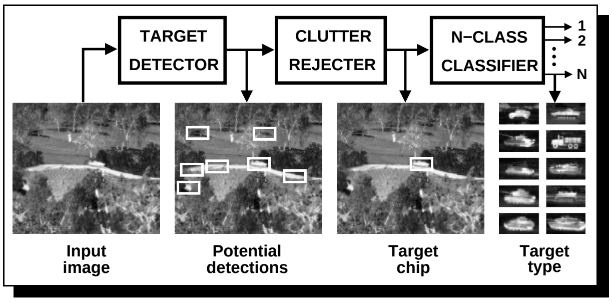

Fig. 2. Detection-recognition modules in an ATR system.

The detection-classification modules, shown in Figure 2, are usually the most important components in an ATR system. The target detector operates on the whole input image and identifies the regions that might contain targets. In order to locate difficult targets, the target detector usually operates at a non-zero false alarm rate. As exemplified by Figure 2, a target detector has found the real target, but has also selected a number of background regions as potential targets. To enhance the performance of the system, a clutter rejecter may be added to reject most of the false-alarms produced by the detector, while eliminating only a few of the targets. Because the number of false-alarms is greatly reduced, the subsequent target classifier may perform its N-class recognition task much more accurately and efficiently. In this paper, we propose a combined automatic target detection/clutter rejection algorithm that maximizes the detection-rejection performance on FLIR imagery.

2. MULTI-STAGE TARGET DETECTOR

Our target detector consists of three distinct stages. The first stage is called range gate detector (RGD), whose function is to detect spatial anomalies based on several localized image features. In the second stage, these anomalies are further examined by a more complicated clutter rejecter that we refer to as EIGMLP in this paper. Finally, the information produced by the RGD and EIGMLP are combined in the third stage by an evidence integrator.

2.1. First Stage: Range Gate Detector

This detector is named *range gate* because the detection process is performed over a set of overlapping gates or zones along the line of sight. The local features incorporated in the RGD were selected by intuition, which are computationally simple and functionally monotonic in nature. The features that we use in RGD are maximum gray level, contrast box, average gradient strength, and local variation.

The maximum gray level, $F^0_{x,y}$, is the highest gray level within a roughly target-sized rectangle centered at the pixel (x, y). Its value at pixel (x, y) is defined as:

$$F^0_{x,y} = \max_{(m,n)\in N_{\text{in}}(x,y)} f(m,n) \; , \tag{1}$$

where $f(m, n)$ is the gray level value of the pixel in the m-th column and n-th row, $N_{\text{in}}(x, y)$ is the neighborhood surrounding the pixel (x, y) and delimited by a rectangle capable of enclosing the largest target in the data set.

The contrast-box feature, F^1, measures the average gray level over a target-sized region, and compares it to the average gray level of the local background. This feature is computed as:

$$F^1 = \frac{1}{n_{\text{in}}} \sum_{m,n}^{N_{\text{in}}} f(m,n) - \frac{1}{n_{\text{out}}} \sum_{m,n}^{N_{\text{out}}} f(m,n) \; , \tag{2}$$

where N_{in} is the same target-sized neighborhood defined above, N_{out} is a larger rectangle that engulfs but excludes all the pixels of N_{in}, while n_{in} and n_{out} is the number of pixels in N_{in} and N_{out}, respectively.

The gradient strength feature, F^2, was chosen because man-made objects tend to show sharper internal detail than natural objects, even when the average intensity is similar. This feature is calculated as:

$$F^2 = \frac{1}{n_{\text{in}}} \sum_{m,n}^{N_{\text{in}}} G_{\text{in}}(m,n) - \frac{1}{n_{\text{out}}} \sum_{m,n}^{N_{\text{out}}} G_{\text{out}}(m,n) \; , \tag{3}$$

$$\begin{aligned} \text{where } G_{\text{in}}(m,n) &= |f(m,n) - f(m,n+1)| \; , \\ &+ \; |f(m,n) - f(m+1,n)| \; , \end{aligned} \tag{4}$$

and $G_{\text{out}}(m, n)$ is defined similarly.

Finally, the local variation feature, F^3, was chosen because man-made object often show greater variation in temperature than natural objects. It is computed as:

$$F^3 = \frac{L_{\text{in}}}{n_{\text{in}}} - \frac{L_{\text{out}}}{n_{\text{out}}} \; , \tag{5}$$

$$L_{\text{in}} = \sum_{m,n}^{N_{\text{in}}} |f(m,n) - \mu_{\text{in}}| \; , \tag{6}$$

$$\mu_{\text{in}} = \frac{1}{n_{\text{in}}} \sum_{m,n}^{N_{\text{in}}} f(m,n) \; , \tag{7}$$

and L_{out} and μ_{out} are defined similarly for the N_{out}.

For each input image, the strength of each feature is first computed for each pixel location. Then the feature values are normalized across the image, so that the feature value at each pixel represents the number of standard deviations that the pixel stands apart from the values for the same feature across the image. The normalized k-th feature value at pixel (x, y) is calculated as:

$$\bar{F}^k_{x,y} = \frac{F^k_{x,y} - \mu_k}{\sigma_k} \; , \tag{8}$$

$$\text{where } \mu_k = \frac{1}{S} \sum_{\forall m,n} F^k_{m,n} \; , \tag{9}$$

$$\sigma_k = \frac{1}{S} \sum_{\forall m,n} (F^k_{m,n} - \mu_k)^2 \; , \tag{10}$$

where S is the size of the feature image.

After the normalization, all features images are linearly combined into a confidence image:

$$\Phi_{x,y} = \sum_{\forall k} w_k \bar{F}^k_{x,y} \tag{11}$$

where the feature weights w_k are determined using a linear fitting algorithm. The confidence value of each pixel is then mapped by a monotonic scaling function $\psi()$ into a value between 0 and 1 for convenience:

$$\psi(\Phi_{x,y}) = 1 - e^{\alpha \Phi_{x,y}} \; , \tag{12}$$

where α is a constant.

Figure 3 shows the internal feature images for all the local features described above, as well as the resulting confidence image, when the RGD operates on the left image in Figure 1. To determine the detection locations from the scaled confidence image $\psi(\Phi)$, the pixel location associated with the maximum confidence value is first chosen and designated as the first detection. Then the confidence values of all pixels situated within a target-sized neighborhood around the peak pixel are set to zero, so that the search for subsequent detections will not choose a pixel location corresponding to the same target again. The process is then repeated for a predefined number of detections.

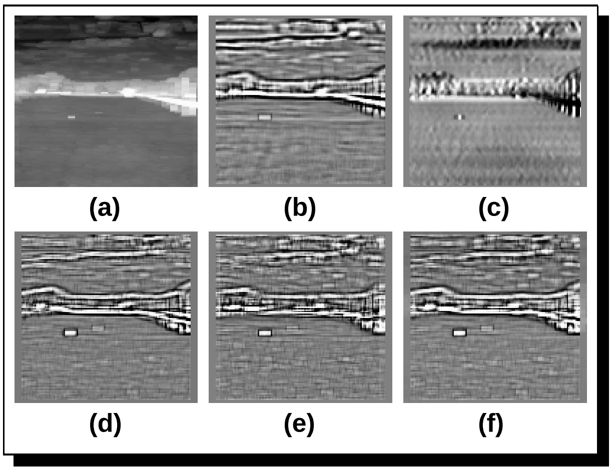

Fig. 3. The internal RGD representations of the left image in Figure 1, which correspond to (a) F^0, (b) large F^1, (c) small F^1, (d) F^2, (e) F^3, and (f) linearly combined map of these five features.

2.2. Second Stage: EIGMLP Clutter Rejecter

The function of a clutter rejecter is to further examine those locations indicated by the detector. The clutter rejection algorithm can be more computationally intensive, because it examines only a small portion of the input image. Because of the limited diversity of the training set, dimensionality reduction is essential to achieve good performance by the subsequent neural classifier. Without dimensionality reduction, a typical learning algorithm would over-fit training data, resulting in a sharp difference between training and testing performance. Our eigen-neural-based clutter rejecter, EIGMLP, uses an eigenspace transformation to perform the dimensionality reduction and a back-end MLP (BMLP) to reject clutter and accept targets. As shown in Figure 4, we implement the whole EIGMLP as a neural network structure by concatenating the eigenspace transformation to the input layer of the BMLP.

The principal component analysis (PCA) [2] is one of the most well-known eigenspace transformation methods. Based on statistical properties of vector representations, the PCA is aptly used as the initial method of eigenspace transformation in our EIGMLP clutter rejecter. We first compute the PCA eigenvectors by performing linear algebraic operations on the covariance matrix of the vectorized regions of interest identified by the RGD. The resulting eigenvectors are then used to initialize the transformation layer of the EIGMLP, while the BMLP is initialized with small random weights. After a quick training on the BMLP with a frozen transformation layer, the EIGMLP is further optimized by allowing changes on the transformation layer based on the error signals back-propagated from the BMLP classifier. The purpose of this optimization is to incorporate class information in the design of the transformation layer. During the joint optimization process, all the weights in EIGMLP are gradually adjusted, using a variety of gradient descent-based algorithms, so that the overall error is reduced at the output node of the BMLP.

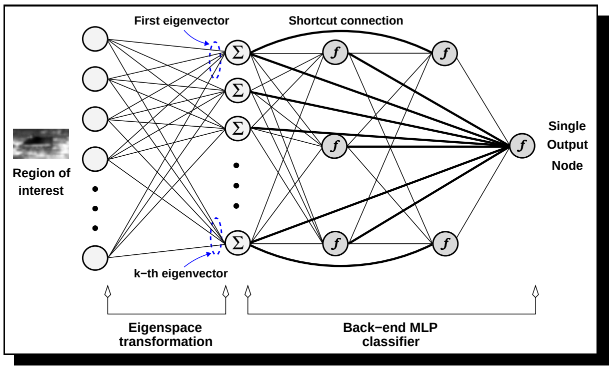

Fig. 4. An EIGMLP that consists of a transformation layer and a back-end MLP.

2.3. Third Stage: Evidence Integrator

The third and final component of our multi-stage detector is an evidence integrator, whose function is to produce the final detection output by combining the outputs from the previous two stages. These outputs include the internal feature values, as well as the output values of the RGD and EIGMLP. Several internal feature values of the RGD and the EIGMLP output value comprise the best feature or evidence set for the integrator, in terms of reaching the highest detection performance on our data set.

The features may be combined in a number of ways. One of the simplest architectures uses the EIGMLP output as a threshold switch to the RGD output. Whenever the EIGMLP output for a region of interest identified by RGD is higher than a predefined threshold value, the corresponding RGD output passes and emerges as the final detection result. Otherwise, that RGD detection is deemed as unreliable and discarded from further consideration.

In addition, we combine these features non-linearly via an evidence-combining MLP (ECMLP). The ECMLP is just another simple MLP, whose structure is very similar to that of the BMLP. Suppose we use the best five feature values from the RGD and the output value of the EIGMLP, then the ECMLP would need only 6 input nodes plus a bias input. Such a small MLP is easy to train, provided the two classes (target versus clutter) in the training data are adequately separable. The output of the ECMLP is the final target-likelihood assessment on the given input chip for our target detector.

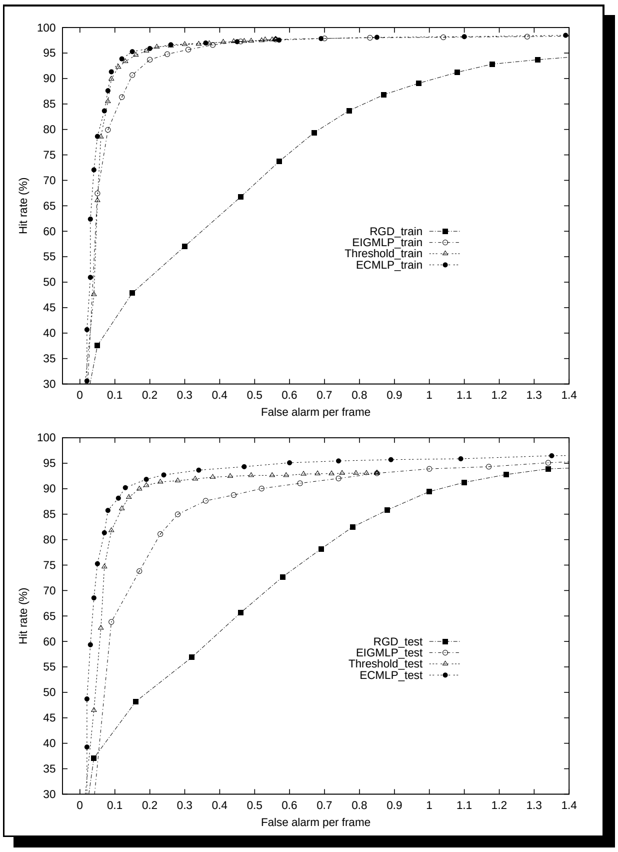

Fig. 5. The detection performance for the training (above) and testing (bottom) set.

3. EXPERIMENTAL RESULTS

We tested the proposed multi-stage detector on a challenging set of FLIR images, two of which are shown in Figure 1. The images were taken with different fields of view, so the size of the targets varies greatly across the frames, occupying areas that covered anything from a few pixels to a third of the frame. We randomly selected 1,756 and 878 frames as the training and testing set, respectively.

First, we used the RGD, operating with the five features shown in Figure 3, to perform the detection task. For each frame, the RGD identifies 10 most potential target areas. These detection results were compared to the ground-truth locations of the corresponding targets and categorized as either a hit or a false alarm. By varying the threshold for acceptable confidence values, thus changing the hit rate and false alarm rate, we were able to plot the receiver operating characteristic (ROC) curves for the training and test sets. As shown in Figure 5, the horizontal axis of these plots represents the average number of false alarms per frame, while the vertical axis represents the percentage of all legitimate targets in the data set that were correctly detected at various confidence thresholds. Based on the ROC curves of the RGD, the first stage detector correctly located about 90 percent of all legitimate targets at 1 false alarm per frame rate, in either the training or the testing set.

The second set of experiments involved the training of EIGMLP clutter rejecters based on the regions of interest identified by the RGD. Using the RGD-detected areas on the training frames, we trained several EIGMLP clutter rejecters that were configured differently in terms of the number of eigenspace projections that the BMLP might consider. We chose the configuration trained with the top-20 projection values, based on the trade-off between computational cost and performance. Based on the resulting ROC curves plotted in Figure 5, we can observe marked improvements of EIGMLP over the RGD results for both the training and testing sets, especially at the operating zone that permits very few false-alarms.

In the third set of experiments, we studied the performance of the evidence integration stage. As a simple approach to combine the evidences produced by the RGD and EIGMLP modules, we accepted any RGD detection that resulted in an EIGMLP output value of 0.5 or higher. Otherwise, that RGD detection was discarded. As shown in Figure 5, the shortened ROC curves of this integration are even better than those of the EIGMLP, especially for the testing set. For example, at 0.1 false alarm per frame rate, the training hit rate achieved by RGD, EIGMLP, and thresholding integration is 44.69, 83.59, and 91.11 percent, respectively. On the testing set, their respective hit rate is 45.36, 64.43, and 82.90 percent at the same false alarm rate.

Finally, we trained an ECMLP that non-linearly combined the five RGD features and the output of EIGMLP to form the final detection decisions. The ROC curves in Figure 5 show that the detection performance of the ECMLP is the highest among all of the architectures tested here, at all confidence values. The training and testing hit rate achieved by ECMLP at 0.1 false alarm per frame rate is 91.88 and 87.54 percent, respectively. It is evident from the testing curves that the ECMLP not only outperformed the thresholding scheme at low false alarm rates, it also had the ability to find those hardest targets if higher false alarm rates could be tolerated.

4. REFERENCES

[1] B. Bhanu and T. Jones, "Image Understanding Research For Automatic Target Recognition," *IEEE Aerospace Electronic System Magazine*, 15–22, Oct. 1993.

[2] I. T. Jolliffe, *Principal Component Analysis*, Springer-Verlag: New York, 1986.