Automatic target recognition using deep convolutional neural networks

Nasser M. Nasrabadi, Hadi Kazemi, Mehdi Iranmanesh

Nasser M. Nasrabadi, Hadi Kazemi, Mehdi Iranmanesh, "Automatic target recognition using deep convolutional neural networks," Proc. SPIE 10648, Automatic Target Recognition XXVIII, 106480I (30 April 2018); doi: 10.1117/12.2304643

SPIE.

Event: SPIE Defense + Security, 2018, Orlando, Florida, United States

Automatic target recognition using deep convolutional neural networks

Nasser M. Nasrabadi, Hadi Kazemi, and Mehdi Iranmanesh

West Virginia University, Morgantown, USA

ABSTRACT

In this paper, we propose a new Automatic Target Recognition (ATR) system, based on Deep Convolutional Neural Network (DCNN), to detect the targets in Forward Looking Infrared (FLIR) scenes and recognize their classes. In our proposed ATR framework, a fully convolutional network (FCN) is trained to map the input 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, corresponding regions centered at these target points are fed to a DCNN to classify them into different target types while at the same time rejecting the false alarms. The proposed architecture achieves a significantly better performance in comparison with that of the state-of-the-art methods on two large FLIR image databases.

Keywords: Automatic Target Recognition (ATR), target detector, deep learning, Deep Convolutional Neural Network (DCNN), FLIR imagery

1. INTRODUCTION

Automatic Target Recognition (ATR) is an important element of many computer vision applications in the civilian area, such as air traffic control,[1] pedestrian tracking,[2,3] animal tracking,[4] sports,[5] and military area.[6,7] It generally refers to classification of the targets and characterization of their attributes such as orientation or sub-class, in a scene of interest without any human intervention. The information about imaged scenes usually is captured using different types of sensors, e.g., Synthetic Aperture Radar (SAR) or FLIR. Generally, all end-to-end ATR systems are comprised of three distinct stages, namely detection, low-level classification (clutter rejection), and high-level classification.[8] Passing the input image through the first two stages, the input scene is classified into targets and clutters (objects that are not targets of interest) regions. Thereafter, the high-level classifier receives the potential targets and classifies them into different target types.

There is a wide range of ATR algorithms proposed in the literature which can be roughly classified into two groups, i.e., learning-based (or feature-based) and model-based.[7] Learning-based approaches extract discriminative features from the training data or learn a subspace representation of the data for the purpose of classification. On the other hand, model-based approaches, such as Hausdorff metric, geometric hashing, and contour matching, use target templates which are built from models of the targets.[9–14] Moreover, in recent past, a hybrid version of learning-based and model-based approaches has emerged in the literature which utilizes the ideas behind these two approaches.[15–17] Even though it is shown in the literature that the model-based approaches offer better performances in comparison with the learning-based approaches, they usually suffer from the lack of computational tractability.[8]

The goal of the learning-based methods is to find a discriminative feature space in which the feature vectors of different target classes, i.e., imposter samples, are in separable regions, while the feature vectors of samples from the same target class, i.e., genuine samples, are close in space and form a cluster. Therefore, the feature extraction phase plays a crucial role in the performance of this group of ATR algorithms. Extensive research have been carried out in the literature to extract the most discriminative and reliable features of the target classes by computing certain types of features, such as Principal Ccomponent Analysis (PCA),[18] wavelet packets,[19] decision boundaries,[20] Histogram of Oriented Gradients (HOG),[21] speeded up robust features.[22] In addition, some other ATR algorithms are based on a specific classification scheme, such as Neural Networks (NN),[23] Learning Vector

E-mail: nasser.nasrabadi@mail.wvu.edu, hakazemi@mix.wvu.edu, seiranmanesh@mix.wvu.edu

Automatic Target Recognition XXVIII, edited by Firooz A. Sadjadi, Abhijit Mahalanobis, Proc. of SPIE
Vol. 10648, 106480I · © 2018 SPIE · CCC code: 0277-786X/18/$18 · doi: 10.1117/12.2304643

Quantization (LVQ),[24] or sparse representations.[25] The fusion of four different classification techniques, namely CNN, LVQ, modular neural network, and SVM, for target recognition task has also been studied in.[26]

Different contour-based algorithms have been proposed in the literature to address the ATR problem.[13,27,28] Duan et al.[13] formulated the ATR in aerial images as an optimization problem and solved it using an improved chemical reaction optimization algorithm. Their method was developed based on a contour grouping strategy called contour cut. In[27] a graph region merging scheme has been utilized for target segmentation in FLIR imagery. They evaluated their method on several real IR ship target images.

PCA,[29–32] and wavelet[33–35] have been also among the most popular feature extraction methods in ATR applications. Most recently, Sparse Representation-based Classification method (SRC) has been widely used for FLIR ATR[25,36,37] and is considered as one of the most promising ATR algorithms. However, Khan et al.[38] proposed a new ATR method based on dense HOG features and Relevance Grouping of Vocabulary (RGV) which outperformed most of the SRC-based methods.

The ATR task has also been approached as a texture analysis problem. A new soft concave-convex partition (SCCP) strategy is proposed in[39] to improve local binary ternary by dividing local features into distinct groups. In another research article,[40] a novel concave-convex local binary ternary feature extractor is proposed which outperformed the state of the art the SRC-based methods.

Two comprehensive surveys on different ATR algorithms, one in SAR imagery,[8] and the other in visible and infrared imagery[41] have been conducted in the literature recently. Their evaluation results suggested that none of the state-of-the-art detectors perform well in all datasets, urging for the development of a more accurate and reliable ATR algorithm.

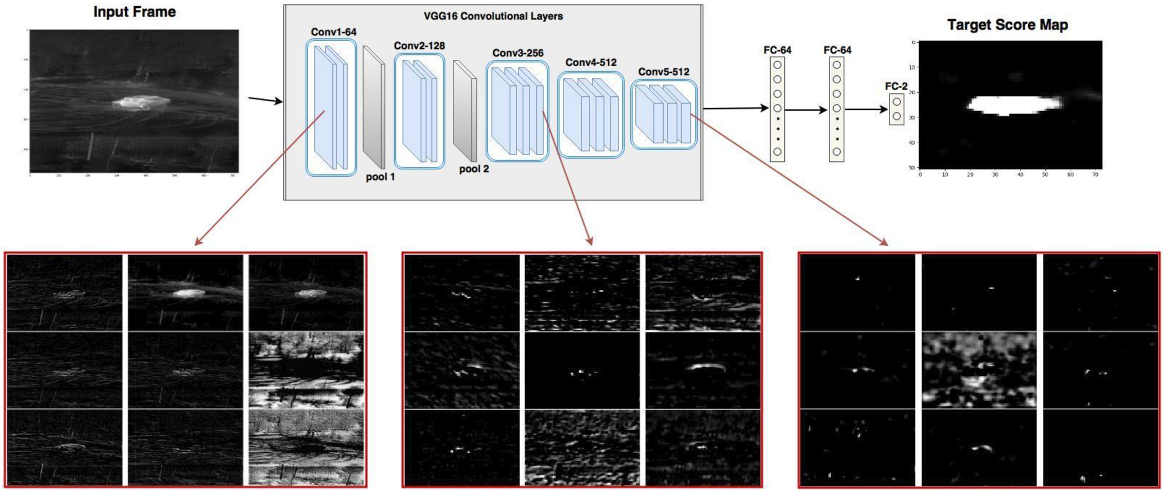
Over the last few years, DCNN has emerged as a powerful machine learning tool which outperformed the traditional approaches and made significant improvements in computer vision problems including object recognition, object detection, semantic segmentation and image synthesis.[42–48] Recent results indicate that CNNs can be utilized as a capable feature extractor and surpass handcrafted features.[49] However, there are very few studies that have exploited the power of CNNs in the context of ATR.[7,50,51]

Motivated by the success of DCNNs, in this paper we proposed a DCNN scheme to perform all the three distinct stages of ATR, namely detection, low-level classification, and high-level classification, simultaneously. For the sake of computational tractability, instead of using a sliding window, the DCNN is implemented as a FCN.[47] Consequently, the network can be applied to any arbitrary size imaged data and generates a score map which provides us the assigned class to each region of the image. Our proposed framework comprises two cascaded DCNNs for low level and high level classifications. The first network distinguishes between background clutters and targets and localizes the potential targets in an FLIR imagery scene. The detected target regions by the first network are fed to the second network to classify them into their target types. The second network is also trained to reject the false alarms of the first network, i.e., the clutters that are detected as potential targets by the first network. This paper is organized as follows: Section 2 describes the data and the preprocessing process; the proposed structure is described in Section 3; Experiment results are discussed in Section 4; and finally, the paper is concluded in Section 5.

2. DATA DESCRIPTION AND PREPROCESSING

In this paper, we use the Comanche (BoeingSikorsky, USA) FLIR dataset for training and evaluation of the proposed method. This dataset consists of different targets at different orientations and ranges. The data were taken under various conditions and divided into two sub-datasets. The first dataset, namely SIG, comprises the images which are taken with targets in open while the images of the less favorable conditions, such as targets in different backgrounds and under various weather conditions, make up the ”ROI” dataset.

The ROI dataset has also two sub-datasets, which we will refer to as ROI_2 and ROI_1. The range of targets in this dataset varies from 688 to 3403 meters. The images of ROI_2 were taken in the summer in the Arizona desert and therefore the background contains high-temperature spots. Consequently, ROI_2 is more difficult database in comparison with ROI_1 whose images were taken in the spring in central California and the background is cool compared with most of the targets in the database. In addition, images in ROI_2 dataset have been taken



(a) Conv1-1 Feature Maps (b) Conv3-3 Feature Maps (c) Conv5-3 Feature Maps

Figure 1: DCNN-detector architecture and its corresponding feature maps for an input frame. Nine sample feature maps are plotted randomly from (a) first convolutional layer (Conv1_1), (a) seventh convolutional layer (Conv3_3), and (c) The thirteenth convolutional layer (Conv5_3). The higher convolutional layers represent more abstract features than those at the lower layers.

from longer distances. More precisely, the range of targets in this dataset varies from 1180 to 5172 meters. The SIG dataset has ten different target classes while the ROI has only five of the ten target classes. In this paper, we denote these targets as TG1 to TG10. For each target, there are images for every 5° in azimuth from 0° to 355° with the total of 72 different orientations.

We used the SIG dataset to train our networks and ROI dataset to evaluate the system. The images of both datasets are 10-bit gray-scale of size 480 × 720 pixels (frames). All the targets of the SIG dataset are cropped and form 13860 target images (chips) of size 40 × 75. We also created 5227 clutter chips to train the target detection network. As we have two max-pooling layers with stride 2, we resized the chips to 40 × 72 so we have feature maps of size 10×18 pixels in the last convolutional layer. The SIG dataset is also randomly divided into 90% training and 10% validation sets. We used the validation set to stop the training process.

The only preprocessing applied on the images is removing the mean of the training dataset from each image before passing it to the networks. In addition, as the input to the original VGG16 network,[52] which is the basis of our DCNNs, is assumed to be in color and therefore the input has 3 channels (RGB), we copied each input on all the three channels before feeding it to the networks.

In the training process, to increase the number of training samples for the sake of over-fitting prevention, and also to make the network less sensitive to scale, translation, and rotation, we augmented the training dataset by random scaling, shifting, rotation, and cropping.

3. PROPOSED SYSTEM

3.1 Convolutional Neural Networks

Convolutional neural network is a specific type of neural networks which are specially developed for 2D inputs such as images. It incorporates a stack of convolutional layers and spatial pooling layers. Convolutional layers apply linear convolutional filters followed by nonlinear activation functions, such as rectifier, sigmoid, or tanh to generate feature maps from the input data. A CNN can learn low level features in its early layers and more semantic features as it goes deeper. That's the justification behind the development of DCNNs to extract more semantic and salient features.

DCNNs have shown a remarkable performance on many different image processing and computer vision tasks such as image classification, segmentation and face recognition. One of the key ideas behind the development

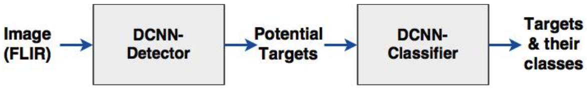


Figure 2: End-to-End ATR System Architecture

of CNNs is that they are able to automatically learn a complex model which extracts the most useful and salient features for the defined task. In addition, the learned features are usually robust to translation, scaling, skewing, and other forms of distortion. On the contrary, most of the traditional feature extraction methods utilize handcrafted features which are based on the understanding of the human from the data. However, DCNNs rely on large training datasets to prevent memorization of the training dataset which is referred to as overfitting.

In general, in some tasks such as classification, a CNN is followed by one or more fully connected layers. The convolutional layers are usually considered as the feature extractor network. The learned feature maps by the last convolutional layer of CNN are vectorized and fed into several stacked fully connected layers to perform the classification task. Even though a linear convolution is sufficient in some cases, features that represent a good abstraction of the input data are generally highly nonlinear functions of them. Therefore, to add nonlinearity to the network, a nonlinear activation function is applied on the output of each layer. By virtue of translation invariance and reducing the spatial size, a number of max or mean pooling layers is usually added to a CNN. Each convolution layer's filter (kernel) has learnable weight, W, and bias, b. These parameters are trained using back-propagation algorithm along with an optimization algorithm such as gradient descent.[53] Figure 1 shows an FLIR image and its corresponding learned feature maps at different convolutional layers. Nine sample feature maps are plotted randomly from the first, seventh, and thirteenth convolutional layers of the network. As it is shown in this figure, the higher convolutional layers extract more abstract features than those at the lower layers.

In summary, we can formulate a convolutional layer as

$$o^k = g(\sum_{i=1}^{n_c}(W_i^k * x_i) + b^k), \tag{1}$$

where o^k denotes the output of k^{th} filter, $g(.)$ is the nonlinear activation function, W_i^k is the corresponding weight of the k^{th} filter for i^{th} input channel, b^k is the bias, $*$ stands for convolution operation, x_l denotes the i^{th} channel of the input, and n_c is the total number of input channels.

In this paper, two distinct DCNNs are utilized for target detection and target classification tasks. The architecture of our proposed system is shown in Figure 2. The first DCNN, called DCNN-detector, takes a FLIR imagery data as an input and detect the potential targets in the image. Subsequently, the detected potential targets are fed to the second DCNN, called DCNN-classifier, which classifies the targets by their classes and rejects the false alarms (clutters) of the first network at the same time.

The incarnation of both DCNNs are based on a VGG network[52] which is pre-trained on the ImageNet dataset. In the original paper,[52] they have proposed multiple structures with different number of layers. There is a trade-off among speed, memory, and accuracy for a given application which determine the best choice of network in terms of depth and complexity. We assessed 4 different VGG architectures with distinct depths, namely VGG11, VGG13, VGG16, and VGG19. Figure 3 compares their performance using their target detection ROC curves for ROI_1 dataset. As the figure shows, the gain in performance from VGG19 to VGG16 is negligible while there is a significant difference in the complexity of them in terms of the number of trainable parameters (Table 1). Utilizing a highly complex network in absence of enough training data can also degrades the performance by overfitting on the training data. As a consequence, we selected the VGG16 as the core structure of our system. For the rest of the paper, all the results are reported based on the experiments using VGG16 architecture.

The original VGG16 has five max-pooling and three fully connected layers. However, we removed the last three max-pooling layers of the networks to increase the size of the output for the sake of more accurate target localization. The three fully connected layers are also replaced with 1 × 1 convolutional layers[47] with different sizes. This replacement gives the network the flexibility of accepting arbitrary sized images as its input and provide us with dense target detection and classification score maps. We also added batch-normalization[54] to

Table 1: Complexity of different VGG networks

Network	VGG11	VGG13	VGG16	VGG19
Number of Parameters (in millions)	15.1	15.3	20.6	25.9

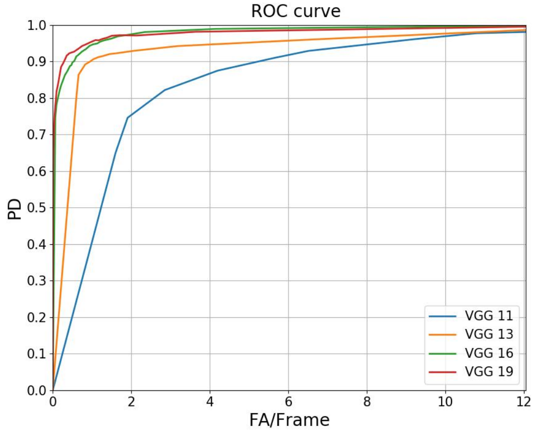


Figure 3: ROC curve for VGG network with different number of layers.

all the VGG16 convolutional and fully connected layers to reduces the internal covarience shift, speed up the training, make the training less sensitive to the initialization, and reduce the chance of overfitting on the training dataset. Finaly, we initialized the convolutional layers of both DCNNs with the pre-trained VGG16 with the ImageNet dataset and then fine tuned the layers with our training sets.

3.2 Target Detection DCNN

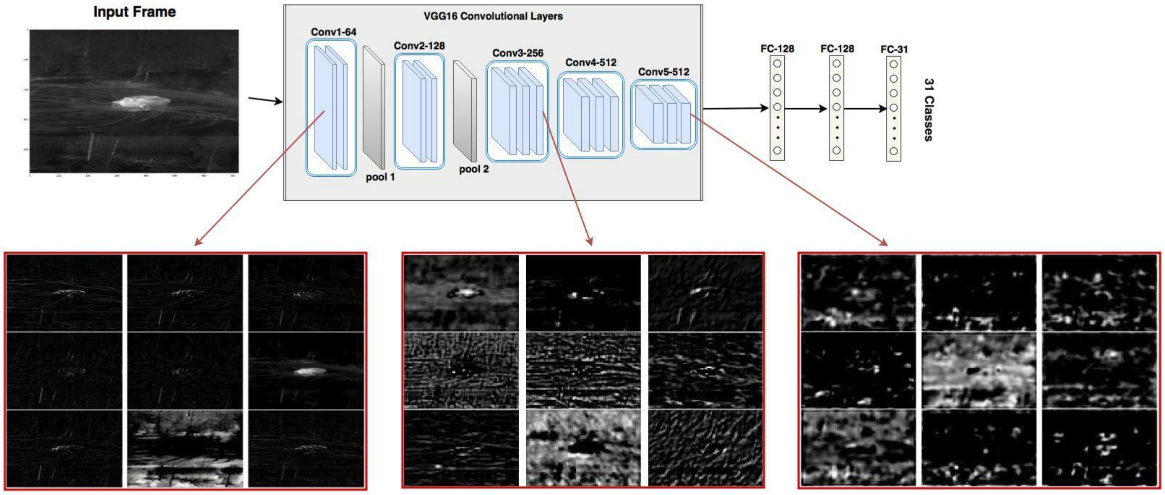
The architecture of the DCNN-detector is illustrated in Figure 1. The main task of this network is localization of potential targets in a FLIR image. During the learning process, the input to the network is a chip image and the output is its corresponding label which could be any of target or clutter classes. A cross-entropy loss function is minimized as the classification loss using the Adam optimization algorithm[55] and is given as follows:

$$loss_1 = -\sum_{n}\sum_{i=1}^{2} y_i^{(n)} \log \hat{y}_i^{(n)}, \tag{2}$$

where $y_i^{(n)}$ is the true probability of the n^{th} chip in the batch of the training data to belong to the i^{th} class which can be 0 or 1. Similarly, $\hat{y}_i^{(n)}$ is the predicted probability of the n^{th} chip in the batch of the training data to belong to the i^{th} class which can be any value between 0 and 1.

However, during the testing phase, a complete test frame is input to the network, and the output of the network is two score maps (heat maps), one for target and the other one for clutter classes. Pixels in each heat map represent the probability of different regions in the input image belonging to a target or a clutter class. In other words, the corresponding receptive field of a target heat map's pixel on the input image belongs to a target if the pixel gets a high value. Note that the clutter and target heat maps are complementary to each other. Figure 1 shows an input frame and only its corresponding target heat map.

Clutter Chip Selection: In addition to the target chips, we need a set of clutter chips to train the DCNN-detector. To maximize the quality of the data, we trained our network using hard negative mining approach[56] which is a popular technique in detection problems. In other words, we start by a set of pre-selected clutters and trained our network. The trained network was tested on the SIG frames (training dataset) and the false alarms were added to the training dataset as new clutter chips. We performed this procedure twice to have enough



(a) Conv1-1 Feature Maps (b) Conv3-3 Feature Maps (c) Conv5-3 Feature Maps

Figure 4: DCNN-classifier architecture and its corresponding feature maps for an input frame. Nine sample feature maps are plotted randomly from (a) first convolutional layer (Conv1_1), (a) seventh convolutional layer (Conv3_3), and (c) The thirteenth convolutional layer (Conv5_3). The higher convolutional layers represent more abstract features than those at the lower layers.

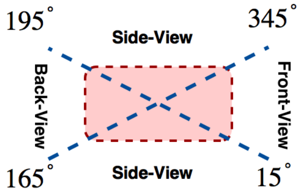


Figure 5: Different Classes based on the target orientation.

clutter chips for the training phase and the performance of the network on the training dataset were maximized.

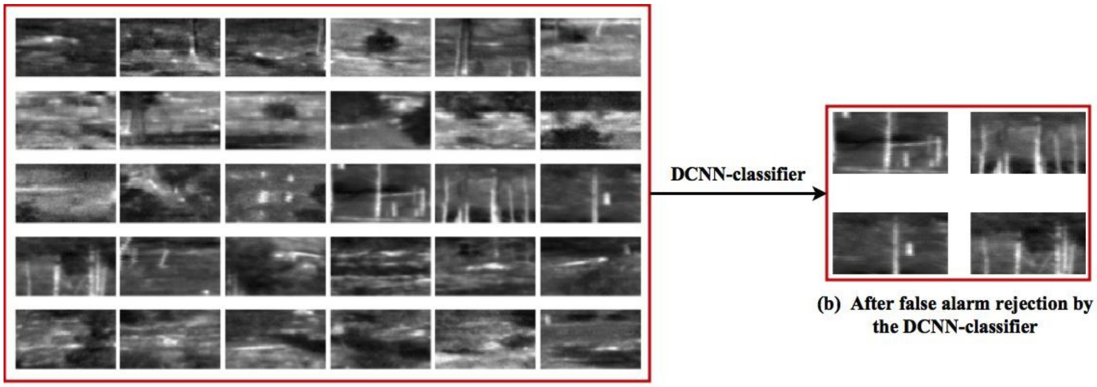
3.3 Target Classification DCNN

The architecture of the DCNN-classifier is demonstrated in Figure 4. This network classifies the potential targets provided by the first network into different target classes. However, because of the huge difference between targets of the same class but with different orientations, each target class is broken down into three classes based on the orientation. Targets are sub-classified into front-view, with an orientation in range of $[345°, 15°]$, rear-view, with an orientation in range of $[165°, 195°]$, and side-view, with an orientation in range of $[15°, 165°]$ or $[195°, 345°]$. Figure 5 shows these three sub-classes of each target type. Therefore, we ended up having 30 different classes for the total of ten target classes. However, to give the DCNN-classifier an opportunity to reject also the false alarms, which are clutter chips detected as targets from the DCNN-detector, we add the 31^{th} class which is the clutter class.

The loss function of this network is the same as the first network but we have 31 prediction output in the cross-entropy loss:

$$loss_2 = -\sum_{n}\sum_{i=1}^{31} y_i^{(n)} \log \hat{y}_i^{(n)}. \tag{3}$$

Even though the target classification network is trained and tested on both target and clutter chips, however due to using fully-convolutional layers as substitute for fully-connected layers, DCNN-classifier also can operate



(a) DCNN-detector false alarms

(b) After false alarm rejection by the DCNN-classifier

Figure 6: (a) A batch of false alarms from DCNN-detector. (b) the remaining false alarms after DCNN-classifier.

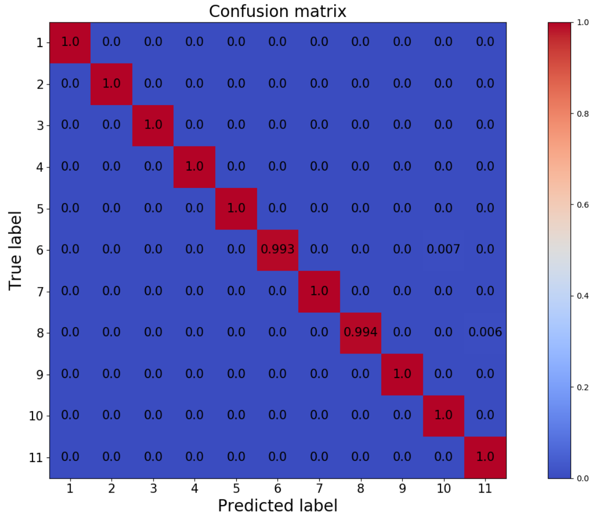


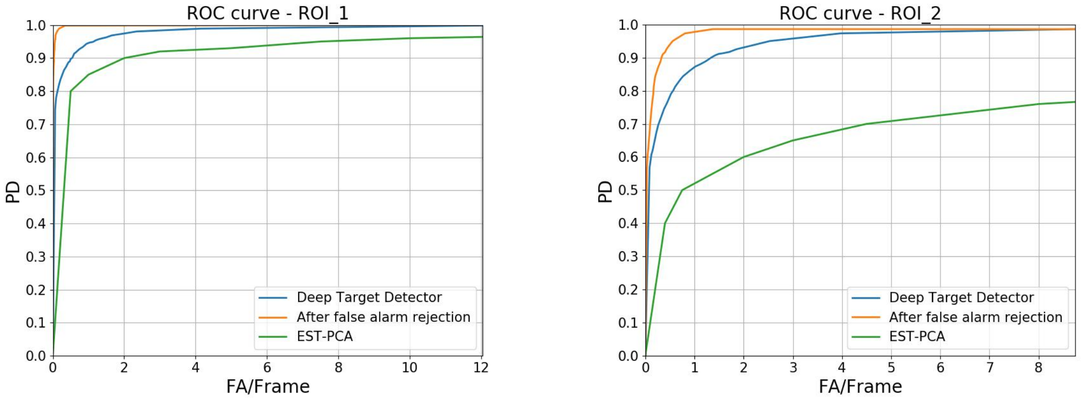
Figure 7: Confusion Matrix - Validation dataset

on the whole frame and predict the classes associated to different regions of the input. In this case, DCNN-classifier is acting as a detector and a target classifier. However, the performance of the network, to perform the both tasks of target detection and classification by its own, is quite poor. Therefore, the first network, DCNN-detector, localizes the potential targets. Subsequently, the predicted target chips are resized to 40 × 72 and fed to the target classification network.

Clutter Chip Selection: For training of the DCNN-classifier, we added a clutter class alongside with the target classes to reject the false alarms of the DCNN-detector. Therefore, we added the remaining false alarms of the DCNN-detector, after training on the augmented dataset (explained in Section 3.2), to the training set of the DCNN-classifier in order to learn the false alarms. Figure 6 shows a batch of false alarms by the DCNN-detector and their remaining false alarms after false alarm rejection by the target classifier DCNN.

4. EXPERIMENTS

In this section, we evaluate our proposed network by comparing it with the state-of-the-art on the ROI dataset.[29] Following the hard negative mining approach, the target detection DCNN is trained three times on the SIG dataset and the new false alarms are combined with the training dataset. The training process termination is based on the performance of the proposed network on the validation dataset. After the third training process, the remaining false alarms are added to the training dataset of the classification DCNN. Thereafter, the target classification network is trained to classify the potential targets into clutters and 30 different target types (three



(a) ROI_1

(b) ROI_2

Figure 8: The ROC curves for target detection, (a) ROI_1 and (b) ROI_2. The blue line shows the output of the DCNN-detector, the orange line shows the ROC curve after clutter rejection by the DCNN-classifier, and the green line shows the EST-PCA detector.[29]

target orientation class for each of the ten target classes). Figure 7 shows the confusion matrix corresponding to the validation dataset. The confusion matrix is plotted based on the accuracy of the predicted target classes despite the correctness of the detected orientation class. In the confusion matrix, row 1 corresponds to the clutter class, while the other 10 rows stand for 10 different target classes.

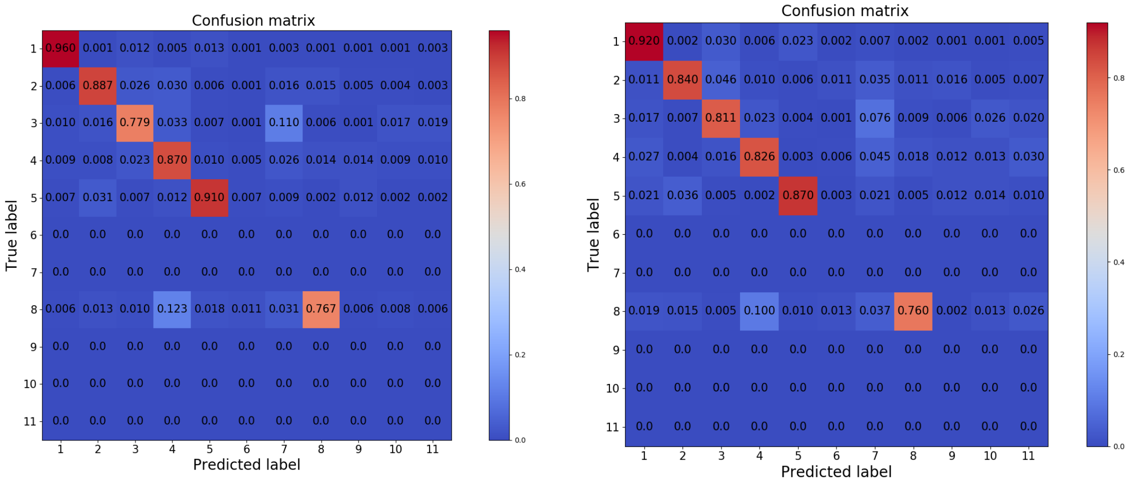
We performed two test experiments on ROI_1 and ROI_2 datasets. In the first step, each FLIR frame is fed to the first network to localize the potential targets within the frame. It uses a multi-resolution image pyramid in order to detect targets at different ranges. The detected potential targets are cropped from the frame, resized to 40×72, and fed to the DCNN-classifier. The ROC results of the DCNN-detector for the proposed implementation as well as the previous state of the art[29] are depicted in Figure 8a and 8b for the ROI_1 and ROI_2 databases, respectively. The detection rate of our DCNN-detector is about 88% for 0.35 false alarm per frame in ROI_1 and 78% for 0.5 false alarm per frame in ROI_2. In these plots, the targets which have an overlap more than %50 with any of the detected potential targets are considered as detected targets.

However, the potential targets are sent to the DCNN-classifier for the false alarm rejection as well as target classification. The corresponding results after false alarm rejection by the second DCNN are also plotted in Figure 8a and 8b (orange lines). The DCNN-classifier significantly improved the results by rejecting most of the false alarms. The detection rate after DCNN-classifier is about 99.8% for 0.35 false alarm per frame in ROI_1 and 97.9% for 0.5 false alarm per frame in ROI_2. For false alarm per frames greater than 0.35 in ROI_1 and 0.5 in ROI_2 we do not have significant improvement in detection rate. Consequently, we can choose the corresponding thresholds for the proposed detector.

The DCNN-classifier also classifies the targets into 30 different target classes in addition to the clutter rejection. Figure 9a and 9b show the confusion matrices of the target classification network for the ROI_1 and ROI_2 databases, respectively. The confusion matrices are barely scattered in off-diagonal elements, meaning the targets are less likely to be misclassified. It should be pointed out that in the test experiments with the ROI dataset, despite the missing five target types (15 classes) but we performed the classification with 31 classes.

To assess the performance of the first network on target localization task, we also calculated the Intersection of Union (IU) metric for the predicted bounding boxes after adopting non-maximum suppression (NMS) on the proposal regions. This metric is defined as the intersection of the predicted and ground truth bounding boxes over their union. Figures 10a and 10b show the histogram of IUs for ROI_1 and ROI_2 datasets, respectively. The mean IU (mIU) for ROI_1 and ROI_2 are 43.5% and 39.28%, respectively.

In couple of recent state-of-the-art studies in the literature, researchers have randomly partitioned the SIG dataset into two non-overlapped sub-groups, SIG-TRAIN (about 80%) and SIG-TEST (about 20%). These subgroups are used for training and testing respectively. For the sake of comparison we performed a new experiment



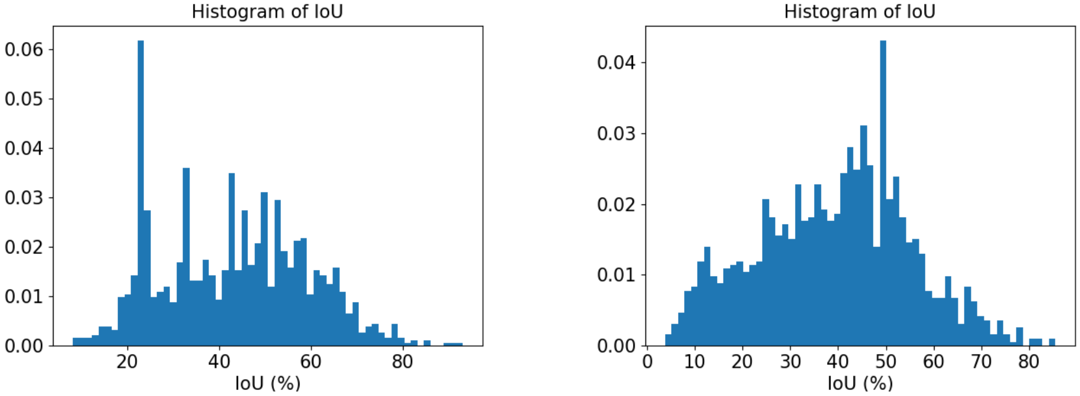
True label \ Predicted label	1	2	3	4	5	6	7	8	9	10	11
1	0.960	0.001	0.012	0.005	0.013	0.001	0.003	0.001	0.001	0.001	0.003
2	0.006	0.887	0.026	0.030	0.006	0.001	0.016	0.015	0.005	0.004	0.003
3	0.010	0.016	0.779	0.033	0.007	0.001	0.110	0.006	0.001	0.017	0.019
4	0.009	0.008	0.023	0.870	0.010	0.005	0.026	0.014	0.014	0.009	0.010
5	0.007	0.031	0.007	0.012	0.910	0.007	0.009	0.002	0.012	0.002	0.002
6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.006	0.013	0.010	0.123	0.018	0.011	0.031	0.767	0.006	0.008	0.006
9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

(a) Confusion Matrix - ROI_1

True label \ Predicted label	1	2	3	4	5	6	7	8	9	10	11
1	0.920	0.002	0.030	0.006	0.023	0.002	0.007	0.002	0.001	0.001	0.005
2	0.011	0.840	0.046	0.010	0.006	0.011	0.035	0.011	0.016	0.005	0.007
3	0.017	0.007	0.811	0.023	0.004	0.001	0.076	0.009	0.006	0.026	0.020
4	0.027	0.004	0.016	0.826	0.003	0.006	0.045	0.018	0.012	0.013	0.030
5	0.021	0.036	0.005	0.002	0.870	0.003	0.021	0.005	0.012	0.014	0.010
6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.019	0.015	0.005	0.100	0.010	0.013	0.037	0.760	0.002	0.013	0.026
9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

(b) Confusion Matrix - ROI_2

Figure 9: Confusion Matrices - (a) ROI_1 and (b) ROI_2 datasets



(a) ROI_1

(b) ROI_2

Figure 10: Histogram of Intersection of Union (IoU), (a) ROI_1 and (b) ROI_2.

Table 2: Comparison of the proposed method with the recent approaches in terms of recognition accuracy (%)
* A single test result on 3456 target chips from both ROI datasets

	Proposed DCNN-classifier		LVQ[24]	NN[23]	Averaged Bayes[26]	RGV[35]	SRC[25]	SCCP[39]
TEST-SIG	99.85		93.4	95.49	-	99.10	97.69	98.72
ROI_1	84.71	83.23*	69.68*	75.58*	82.1*	-	-	-
ROI_2	82.73					-	-	-

by partitioning the SIG dataset into SIG-TRAIN and SIG-TEST with the same 80%-20% portions. The results of our proposed DCNN (classifier), LVQ,[24] NN,[23] Averaged Bayes,[26] RGV,[35] SRC,[25] and SCCP[39] methods on the SIG-Test are reported in Table 2.

5. CONCLUSION

In this work, a framework for ATR using the concept of Deep Convolutional Neural Networks (DCNNs) has been developed. The framework compromises two DCNNs: the first DCNN detects and localizes the potential targets in an FLIR imagery scene, and the second network recognizes the class associated with each potential target. The second DCNN is also able to reject the false alarms of the target detector network. Due to the inherent nature of DCNNs the proposed approach is robust to scale, translation, rotation, and illumination. To improve the robustness we augmented the training dataset by random scaling, shifting, rotation, and cropping. Both networks were fine-tuned with augmented data after we initialized their convolutional layers with the weights of a pre-trained VGG16 on ImageNet dataset. To show the effectiveness of the proposed framework, we conducted two experiments on the Comanche (Boeing Sikorsky, USA) FLIR ROI datasets. The experiments have shown substantial improvements in the target detection and recognition in comparison with the previous sate-of-the-art for these datasets. The proposed method can be easily adapted to other ATR applications as well.

REFERENCES

[1] Pierucci, L. and Bocchi, L., "Improvements of radar clutter classification in air traffic control environment," in [*Signal Processing and Information Technology, 2007 IEEE International Symposium on*], 721–724, IEEE (2007).

[2] Fernández-Caballero, A., López, M. T., and Serrano-Cuerda, J., "Thermal-infrared pedestrian ROI extraction through thermal and motion information fusion," *Sensors* **14**(4), 6666–6676 (2014).

[3] Li, X., Guo, R., and Chen, C., "Robust pedestrian tracking and recognition from FLIR video: A unified approach via sparse coding," *Sensors* **14**(6), 11245–11259 (2014).

[4] Christiansen, P., Steen, K. A., Jørgensen, R. N., and Karstoft, H., "Automated detection and recognition of wildlife using thermal cameras," *Sensors* **14**(8), 13778–13793 (2014).

[5] Gade, R. and Moeslund, T. B., "Thermal tracking of sports players," *Sensors* **14**(8), 13679–13691 (2014).

[6] Schachter, B., [*Automatic Target Recognition, Second Edition*], SPIE (2017).

[7] Li, B., Chellappa, R., Zheng, Q., Der, S., Nasrabadi, N., Chan, L., and Wang, L., "Experimental evaluation of FLIR ATR approaches: A comparative study," *Computer Vision and image understanding* **84**(1), 5–24 (2001).

[8] El-Darymli, K., Gill, E. W., Mcguire, P., Power, D., and Moloney, C., "Automatic target recognition in synthetic aperture radar imagery: A state-of-the-art review," *IEEE Access* **4**, 6014–6058 (2016).

[9] Verly, J. G. and Delanoy, R. L., "Model-based automatic target recognition ATR system for forwardlooking groundbased and airborne imaging laser radars (LADAR)," *Proceedings of the IEEE* **84**(2), 126–163 (1996).

[10] Hummel, R., "Model-based ATR using synthetic aperture radar," in [*Record of the IEEE 2000 International Radar Conference [Cat. No. 00CH37037]*], 856–861 (2000).

[11] Bharadwaj, P. and Carin, L., "Infrared-image classification using hidden Markov trees," *IEEE Transactions on pattern analysis and machine intelligence* **24**(10), 1394–1398 (2002).

[12] Lamdan, Y. and Wolfson, H. J., "Geometric hashing: A general and efficient model-based recognition scheme," *In Proc. of ICCV* (1988).

[13] Duan, H. and Gan, L., "Elitist chemical reaction optimization for contour-based target recognition in aerial images," *IEEE Transactions on Geoscience and Remote Sensing* **53**(5), 2845–2859 (2015).

[14] Wu, J., Mao, S., Wang, X., and Zhang, T., "Ship target detection and tracking in cluttered infrared imagery," *Optical Engineering* **50**(5), 057207–057207 (2011).

[15] Liebelt, J., Schmid, C., and Schertler, K., "Viewpoint-independent object class detection using 3D feature maps," in [*Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*], 1–8, IEEE (2008).

[16] Khan, S. M., Cheng, H., Matthies, D., and Sawhney, H., "3D model based vehicle classification in aerial imagery," in [*Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*], 1681–1687, IEEE (2010).

[17] Toshev, A., Makadia, A., and Daniilidis, K., "Shape-based object recognition in videos using 3D synthetic object models," in [*Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*], 288–295, IEEE (2009).

[18] Wang, Y., Luo, L., Freedman, M. T., and Kung, S.-Y., "Probabilistic principal component subspaces: a hierarchical finite mixture model for data visualization," *IEEE Transactions on Neural Networks* **11**(3), 625–636 (2000).

[19] Tian, B., Shaikh, M. A., Azimi-Sadjadi, M. R., Haar, T. H. V., and Reinke, D. L., "A study of cloud classification with neural networks using spectral and textural features," *IEEE Transactions on Neural Networks* **10**(1), 138–151 (1999).

[20] Lee, C. and Landgrebe, D. A., "Decision boundary feature extraction for neural networks," *IEEE Transactions on Neural Networks* **8**(1), 75–83 (1997).

[21] Kobayashi, T., "BFO meets HOG: feature extraction based on histograms of oriented PDF gradients for image classification," in [*Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*], 747–754 (2013).

[22] Juan, L. and Gwun, O., "A comparison of SIFT, PCA-SIFT and SURF," *International Journal of Image Processing (IJIP)* **3**(4), 143–152 (2009).

[23] Wang, L.-C., Der, S. Z., and Nasrabadi, N. M., "Automatic target recognition using a feature-decomposition and data-decomposition modular neural network," *IEEE Transactions on Image Processing* **7**(8), 1113–1121 (1998).

[24] Chan, L. A. and Nasrabadi, N. M., "Automatic target recognition using vector quantization and neural networks," *Optical Engineering* **38**(12), 2147–2161 (1999).

[25] Patel, V. M., Nasrabadi, N. M., and Chellappa, R., "Sparsity-motivated automatic target recognition," *Applied optics* **50**(10), 1425–1433 (2011).

[26] Rizvi, S. A. and Nasrabadi, N. M., "Fusion of FLIR automatic target recognition algorithms," *Information Fusion* **4**(4), 247–258 (2003).

[27] Tao, W., Jin, H., and Liu, J., "Unified mean shift segmentation and graph region merging algorithm for infrared ship target segmentation," *Optical Engineering* **46**(12), 127002–127002 (2007).

[28] Yu, L., Fan, G., Gong, J., and Havlicek, J. P., "Joint infrared target recognition and segmentation using a shape manifold-aware level set," *Sensors* **15**(5), 10118–10145 (2015).

[29] Young, S. S., Kwon, H., Der, S. Z., and Nasrabadi, N. M., "Adaptive target detection in forward-looking infrared imagery using the eigenspace separation transform and principal component analysis," *Optical Engineering* **43**(8), 1767–1776 (2004).

[30] Chan, A. L., Der, S. Z., and Nasrabadi, N. M., "A joint compression-discrimination neural transformation applied to target detection," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* **35**(4), 670–681 (2005).

[31] Hou, Q.-y., Zhang, W., Wu, C.-f., Li, Q.-m., Lu, L.-h., and Cao, Y.-m., "Adaptive small target detection based on evaluating complex degree of infrared image," in [*International Symposium on Photoelectronic Detection and Imaging 2009*], 738317–738317, International Society for Optics and Photonics (2009).

[32] Cao, Y., Liu, R. M., and Yang, J., "Infrared small target detection using PPCA," *International Journal of Infrared and Millimeter Waves* **29**(4), 385–395 (2008).

[33] Vasuki, P. and Roomi, S. M. M., "Automatic target recognition for SAR images by discrete wavelet features," *European Journal of Scientific Research* **80**(1), 133–139 (2012).

[34] Kumar, B., Prabhakar, B., Suryanarayana, K., Thilagavathi, V., and Rajagopal, R., "Target identification using harmonic wavelet based ISAR imaging," *EURASIP Journal on Applied Signal Processing* **2006**, 132–132 (2006).

[35] Khan, J. and Alam, M., "Target detection in cluttered FLIR imagery using probabilistic neural networks," in [*Defense and Security*], 55–66, International Society for Optics and Photonics (2005).

[36] Zhao, J., Chen, J., Chen, Y., Feng, H., Xu, Z., and Li, Q., "Sparse-representation-based automatic target detection in infrared imagery," *Infrared Physics & Technology* **56**, 85–92 (2013).

[37] Chen, Y., Nasrabadi, N. M., and Tran, T. D., "Sparse representation for target detection in hyperspectral imagery," *IEEE Journal of Selected Topics in Signal Processing* **5**(3), 629–640 (2011).

[38] Khan, M. N. A., Fan, G., Heisterkamp, D. R., and Yu, L., "Automatic target recognition in infrared imagery using dense hog features and relevance grouping of vocabulary," in [*Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*], 293–298 (2014).

[39] Wu, X., Sun, J., Fan, G., and Wang, Z., "Improved local ternary patterns for automatic target recognition in infrared imagery," *Sensors* **15**(3), 6399–6418 (2015).

[40] Sun, J., Fan, G., Yu, L., and Wu, X., "Concave-convex local binary features for automatic target recognition in infrared imagery," *EURASIP Journal on Image and Video Processing* **2014**(1), 1–13 (2014).

[41] Razakarivony, S. and Jurie, F., "Vehicle detection in aerial imagery: A small target detection benchmark," *Journal of Visual Communication and Image Representation* **34**, 187–203 (2016).

[42] Krizhevsky, A., Sutskever, I., and Hinton, G. E., "Imagenet classification with deep convolutional neural networks," in [*Advances in neural information processing systems*], 1097–1105 (2012).

[43] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A., "Going deeper with convolutions," in [*Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*], 1–9 (2015).

[44] Kazemi, H., Iranmanesh, M., Dabouei, A., and Nasrabadi, N. M., "Facial attributes guided deep sketch-to-photo synthesis," in [*Applications of Computer Vision (WACV), 2018 IEEE Workshop on*], IEEE (2018).

[45] He, K., Zhang, X., Ren, S., and Sun, J., "Deep residual learning for image recognition," in [*Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*], 770–778 (2016).

[46] Iranmanesh, M., Dabouei, A., Kazemi, H., and Nasrabadi, N. M., "Deep cross polarimetric thermal-to-visible face recognition," in [*Biometrics (ICB), 2018 International Conference on*], IEEE (2018).

[47] Long, J., Shelhamer, E., and Darrell, T., "Fully convolutional networks for semantic segmentation," in [*Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*], 3431–3440 (2015).

[48] Dabouei, A., Kazemi, H., Iranmanesh, M., and Nasrabadi, N. M., "Fingerprint distortion rectification using deep convolutional neural networks," in [*Biometrics (ICB), 2018 International Conference on*], IEEE (2018).

[49] Sharif Razavian, A., Azizpour, H., Sullivan, J., and Carlsson, S., "CNN features off-the-shelf: an astounding baseline for recognition," in [*Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*], 806–813 (2014).

[50] Mirelli, V. and Rizvi, S. A., "Automatic target recognition using a multilayer convolution neural network," *in Proc. SPIE* **2755**, 106–125 (1996).

[51] Kim, S., Song, W.-J., and Kim, S.-H., "Infrared variation optimized deep convolutional neural network for robust automatic ground target recognition," in [*Computer Vision and Pattern Recognition Workshops (CVPRW), 2017 IEEE Conference on*], 195–202, IEEE (2017).

[52] Simonyan, K. and Zisserman, A., "Very deep convolutional networks for large-scale image recognition," *in Proceedings of International Conference on Learning Representations* (2015).

[53] Bottou, L., "Large-scale machine learning with stochastic gradient descent," in [*Proceedings of COMPSTAT'2010*], 177–186, Springer (2010).

[54] Ioffe, S. and Szegedy, C., "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)* , 448–456 (2015).

[55] Kingma, D. and Ba, J., "Automatic target recognition using a feature-decomposition and data-decomposition modular neural network," *International Conference on Learning Representations (ICLR)* (2015).

[56] Felzenszwalb, P., McAllester, D., and Ramanan, D., "A discriminatively trained, multiscale, deformable part model," in [*Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*], 1–8, IEEE (2008).