Deep Convolutional Neural Networks with Integrated Quadratic Correlation Filters for Automatic Target Recognition

Brian Millikan, Hassan Foroosh
Department of Electrical Engineering and Computer Science
University of Central Florida
{brian.millikan@knights, foroosh@cs}.ucf.edu

Qiyu Sun
Department of Mathematics
University of Central Florida
qiyu.sun@ucf.edu

Abstract

Automatic target recognition involves detecting and recognizing potential targets automatically, which is widely used in civilian and military applications today. Quadratic correlation filters were introduced as two-class recognition classifiers for quickly detecting targets in cluttered scene environments. In this paper, we introduce two methods that integrate the discrimination capability of quadratic correlation filters with the multi-class recognition ability of multi-layer neural networks. For mid-wave infrared imagery, the proposed methods are demonstrated to be multi-class target recognition classifiers with very high accuracy.

1. Introduction

Automatic target recognition (ATR) involves detecting and recognizing potential targets automatically, which is widely used in civilian and military applications today. In the 1980s, ATR was first introduced in the 1980s with aircraft-mounted targeting pods for military aircraft [21,8]. Numerous ATR algorithms have been proposed during the last few decades [22,21,5,2,1], and it is still a crucial element to many military applications today [22,8]. An ATR system can be divided into four parts [3,9]: detection, clutter rejection, feature extraction and classification. In this paper, we consider the latter two items together by introducing a quadratic correlation filter (QCF) coupled with a convolutional neural network (CNN).

QCFs were introduced by Mahalanobis et al. in [13] to perform target detection. The original QCF is based on the Fukunaga-Koontz Transform (FKT), which keeps the eigenvectors corresponding to the largest and smallest eigenvalues. A single-layer perceptron QCF was introduced in [14] and shown to be effective for two-class target detection. In many military mid-wave infrared (MWIR) applications, multi-class target recognition is required. This suggests the modification of the two-class target detection methods to support multi-class target recognition.

2. Quadratic Correlation Filters

The quadratic correlation filter (QCF) was developed in [13] to efficiently detect targets in a cluttered scene. In their paper, a couple of methods are proposed to find a QCF H to perform target detection. The filter H = FF^T − GG^T can be a composite matrix consisting of a filter F to detect targets and a filter G to detect the background [13,14]. It can be applied efficiently to evaluate a scene at every location s in the image. If a region of the image x at location s is rearranged in vectorized form as x_s which is of dimension $\mathbb{R}^{rc}$ for an $r \times c$ region, then the application of the QCF $H \in \mathbb{R}^{rc \times rc}$ is $\varphi(x_s) = x_s^T H x_s$, which can be...
we use a differentiable squashing function, \( \sigma \). We expect that \( x \) occurred. Given a test image patch \( x \), we can classify it as a target or the background by examining \( \sigma(x) \). Since multi-layer perceptron neural networks are fully-connected networks, the input size must match the training input patch size. This means that this network must be applied multiple times in a sliding window type approach to perform classification for an entire image. To remedy this, we propose an all-convolutional CNN with a QCF layer for multi-class target recognition and take advantage of the invariance properties of the CNN. In addition, an all-convolutional CNN offers the ability to perform pixel-wise image classification using only image patches for training. This makes this type of CNN ideal for ATR applications.

3. Proposed Methods

In Section we recalled a single-layer perceptron QCF for two-class target detection. However, target recognition requires a multi-class discriminator. Therefore, in this section we propose a multi-layer perceptron neural network with a quadratic filter input layer for multi-class target recognition. A design of QCFs proposed in minimizes the sum of the squared error between the output statistic \( \varphi \) and the true label \( d \) for all training samples. It is based on the back propagation training algorithm for kernel neurons to generate the filter [25]. Denote the statistic of an image patch \( x_n \) by

\[
\varphi(x_n) = x_n^T H x_n, \quad 1 \leq n \leq N_e,
\]

where \( N_e = 2N_S \) is the total number of training images. We expect that \( x_n \) is a target when \( \varphi(x_n) \) is positive and large, it is the background when \( \varphi(x_n) \) is negative and small.

Denote the output of the single-layer perception \( d_n \in \{-1, 1\} \) by 1 when the training image is a target and by \(-1\) when it is the background. To meet the expectation, we use a differentiable squashing function, \( \sigma(\varphi(x_n)) = \tanh(\varphi(x_n)) \), to force the output of \( \varphi(x_n) \) to be between \(-1\) and \(+1\), and minimize the objective function

\[
J(H) = \frac{1}{2} \sum_{n=1}^{N} |d_n - \sigma(\varphi(x_n))|^2 = \frac{1}{2} \sum_{n=1}^{N} |d_n - \sigma(x_n^T H x_n)|^2.
\]

We can iteratively solve for \( H \) using the gradient descent algorithm

\[
H_{t+1} = H_t - \eta \nabla J(H_t), \quad t \geq 0,
\]

and \( H_0 = 0 \), where the gradient \( \nabla J(H) \) is given by

\[
\nabla J(H) = -\sum_{n=1}^{N} [d_n - \sigma(\varphi(x_n))] (1 - \sigma^2(\varphi(x_n))) x_n x_n^T.
\]

We denote the solution to \( J \) as \( H^* \) which occurs after the convergence criterion has been met or \( T \) iterations have occurred. Given a test image patch \( x \), we can classify it as a target or the background by examining \( \sigma(\varphi(x)) \) which will be between 0 and 1 for a target and between \(-1\) and 0 for background, see Figure 1.

Figure 1: Target detection using a quadratic correlation filter (QCF) learned from a single-layer perceptron neural network (SLPNN): (a) Original image with an armoured personnel carrier (BMP2), (b) correlation output. We observe that the BMP2 can be detected, but not recognized or identified, using this method.
Figure 2: Quadratic multi-layer perceptron neural network (QMLPNN) for target recognition. The first layer is composed of a quadratic filter which requires a 3D weight tensor. The rest of the network forms a standard MLPNN.

3.1. Quadratic Multi-layer Perceptron Neural Network for Target Recognition

In this subsection, we introduce a quadratic multi-layer perceptron neural network (QMLPNN) for multi-class target recognition, see Figure 2. Given an input patch $x_n$, $1 \leq n \leq N$ of size $m$, the output of the first hidden layer is a quadratic function, which is learned during the network training, of the form

$$z_n^{(2)} = \left( \sigma(x_n^T H_k^{(1)} x_n + b_k^{(1)}) \right)_{1 \leq k \leq K_1},$$

where $K_1$ is the number of hidden nodes in the first layer, $H^{(1)} = \{H_k^{(1)}\}_{1 \leq k \leq K_1}$ is a three-dimensional weight tensor of dimension $m \times m \times K_1$, $b^{(1)} = \{b_k^{(1)}\}_{1 \leq k \leq K_1}$ forms a bias vector and $\sigma$ is a non-linear activation function. The other hidden layers are composed of conventional activation functions, $z_n^{(\ell+1)} = \left( \sigma(h_k^{(\ell)}, z_n^{(\ell)}) + b_k^{(\ell)} \right)_{1 \leq k \leq K_\ell}$, $2 \leq \ell \leq L - 1$, where $K_\ell$ is the number of hidden nodes, $H^{(\ell)} = \{h_k^{(\ell)}\}_{1 \leq k \leq K_\ell}$ forms the weight matrix of dimension $K_\ell \times K_{\ell-1}$, $b^{(\ell)} = \{b_k^{(\ell)}\}_{1 \leq k \leq K_\ell}$ forms the bias vector for the $\ell$-th layer, and $L$ is the total number of layers [4][15].

We denote the set of all target class types by $\mathcal{C}$ and label each input pattern by $d_n \in \mathcal{C}$, $1 \leq n \leq N$. We note that the last layer has $K_L$ hidden nodes which is equal to the number of classes in $\mathcal{C}$. Given this, we write $H^{(L)} = \{h_k^{(L)}\}_{c \in \mathcal{C}}$ for convenience. The objective function $E_n(H, b)$ for the input patch $x_n$ is defined as

$$E_n(H, b) = - \sum_{c \in \mathcal{C}} [d_n = c] \log \left( \sum_{j \in \mathcal{C}} \exp \left( (h_j^{(L)}, z_n^{(L)}) + b_j^{(L)} \right) \right)$$

(7)

where $[i = j]$ is the Iverson bracket notation for the Kronecker delta function, $H = \{H^{(1)}, H^{(2)}, ..., H^{(L)}\}$ and $b = \{b^{(1)}, b^{(2)}, ..., b^{(L)}\}$.

Training is accomplished using gradient descent with gradients determined by the backpropagation algorithm. All layers will result in a weight matrix with the exception of the last layer, which is a three-dimensional tensor. The error term is

$$\delta^{(L)} = - \left( [d_n = c] - P \left[ d_n = c | z_n^{(L)}, H^{(L)}, b^{(L)} \right] \right)_{c \in \mathcal{C}},$$

(8)

for the last layer, and

$$\delta^{(\ell)} = (H^{(\ell)})^T \delta^{(\ell+1)} \circ (\sigma'(h_k^{(\ell)}, z_n^{(\ell)}))_{1 \leq k \leq K_\ell}$$

(9)

for internal layers $2 \leq \ell \leq L - 1$, where we use the notation $\circ$ to represent the element-wise Hadamard product [15]. The internal layers in the network have standard fully-connected nodes. As in [4][15], the gradient associated with error term $\delta^{(\ell+1)}$ is

$$\nabla_{H^{(\ell)}} E_n(H, b) = \delta^{(\ell+1)}(z_n^{(\ell)})^T.$$  

(10)

However, the framework in the first layer [5] is specifically designed for multi-target recognition networks. The gradient associated with the objective function $E_n(H, b)$, is defined by

$$\nabla_{H_k^{(1)}} E_n(H, b) = \delta_k^{(2)} x_n x_n^T, 1 \leq k \leq K_1.$$  

(11)

3.2. Quadratic Correlation Filter Convolutional Neural Network for Target Recognition

In this section, we introduce a quadratic correlation filter convolutional neural network (QCFCNN) for automatic target recognition. The weight matrix $H^{(1)} = \{H_k^{(1)}\}_{1 \leq k \leq K_1}$, for the first layer, illustrated in Figure 3 contains multiple QCFs $H^{(1)}$ of size $M_1 \times M_1$ that are applied to the input image. To apply the weight matrix in the first layer, we define

$$\Theta_k^{(1)} = H_k^{(1)} + (H_k^{(1)})^T,$$  

(12)
and perform its eigendecomposition
\[
\Theta_k^{(1)} V_k = V_k \Lambda_k
\]
where the diagonal matrix \(\Lambda_k\) has real eigenvalues \(\lambda_{k,M_1} \geq \lambda_{k,M_1-1} \geq \cdots \geq \lambda_{k,1}\) as its diagonal entries and orthonormal matrix \(V_k = [v_{k,1}, \ldots, v_{k,M_1}]\) contains the corresponding eigenvectors. The first layer is
\[
z_n^{(2)} = \left(\sigma \left( \sum_{i=1}^{M_1} \lambda_{k,i} |x_n \otimes v_{k,i}|^2 + b_k^{(1)} \right) \right)_{1 \leq k \leq K_1}
\]
for every location in the image \(x\) of height \(M\) and width \(K\), where the symbol \(\otimes\) indicates the 2D cross-correlation operation. The internal layers are standard convolutional layers,
\[
z_n^{(\ell+1)} = \left(\sigma \left( \sum_{i=1}^{K_{\ell-1}} h_{k,i}^{(\ell)} \otimes z_{n,i}^{(\ell)} + b_k^{(\ell)} \right) \right)_{1 \leq k \leq K_\ell}, 2 \leq \ell \leq L-1,
\]
where \(L\) is the total number of layers in the CNN.

Training is accomplished by standard backpropagation for convolutional neural networks. The standard error function for CNN network internal layers is
\[
\delta_k^{(\ell)} = \text{upsample} \left( \sum_{i=1}^{K_{\ell-1}} h_{k,i}^{(\ell)} \otimes f \delta_k^{(\ell+1)} \right)
\]
\[
\circ \sigma' \left( \sum_{i=1}^{K_{\ell-1}} h_{k,i}^{(\ell)} \otimes z_{n,i}^{(\ell)} + b_k^{(\ell)} \right)
\]
which can be used to define the gradient, where \(2 \leq \ell \leq L-1, 1 \leq k \leq K_\ell\) and the notation \(\otimes f\) denotes a full 2D cross-correlation operation. However, for the gradient definition, we must first define the vectorization of a \(d \times d\) subimage \(X_{l,m}^{(n)}\),
\[
\vec{X}_{l,m}^{(n)} = [x_{l,m}^{(n)}, \ldots, x_{l,m+d-1}^{(n)}, x_{l+1,m}^{(n)}, \ldots, x_{l+1,m+d-1}^{(n)}, \ldots, x_{l+d-1,m}^{(n)}, \ldots, x_{l+d-1,m+d-1}^{(n)}]^T.
\]
Finally, we can evaluate the gradient for the first layer by
\[
\nabla_{H_k^{(1)}} E_n(H,b) = \frac{\partial E_n}{\partial H_k^{(1)}} = \frac{\partial E_n}{\partial \Theta_k^{(1)}} \frac{\partial \Theta_k^{(1)}}{\partial H_k^{(1)}}
\]
\[
= \sum_{l,m} \delta_k^{(2)} \vec{X}_{l,m}^{(n)} \vec{X}_{l,m}^{(n)^T},
\]
where \(1 \leq k \leq K_1\).

4. Numerical Experiments

We conducted our simulations using the SENSIA ATR Algorithm Image Database [23]. This database is a mid-wave infrared (MWIR) dataset from the U.S. Army Night Vision and Electronic Sensors Directorate (NVESD). It contains 207GB of MWIR data which includes 10 vehicle target types and 2 scenarios of humans. As shown in Figure 4 we considered all 10 vehicle target types, which include a Pickup Truck (PICKUP), Sport Utility Vehicle (SUV), Armored Personnel Carriers (BTR70 and BMP2), an Infantry Scout Vehicle (BRDM2), a Main Battle Tank (T72), an Anti-Aircraft Weapon (ZSU23-4), a Self-Propelled Howitzer (2S3), an Armoured Reconnaissance Vehicle (MTLB), and a Towed Howitzer (D20). A 40 x 80 bounding box is formed around each target using the ground truth data to generate a target image patch, \(x\), for training.

In our simulations, we select images that contain targets less than 3000 meters in range from the camera at any time of day. We train a single neural network (NN) using a different class of targets within each range of aspect angles. For instance, we may divide up a 360° aspect angle range into 4 separate classes for each target category. For a 10 target set, this will yield a total of 40 classes. All of these factors make this a difficult dataset to classify. In our simulations, a misclassification is only counted if a target is classified outside of its target category (i.e. aspect angle is ignored). In order to compare the performance of each classification method, we train and test using image patches with the target centered in the patch. For the QCFCNN, we also train with example background image patches to segment target areas from background using the embedded QCF. Since this network is built using all convolutional layers, we can train
Figure 4: Target images in the visible spectrum: (a) civilian pickup truck (PICKUP), (b) civilian sport utility vehicle (SUV), (c) armoured personnel carrier (BMP2), (d) armoured personnel carrier (BTR70), (e) main battle tank (T72), (f) infantry scout vehicle (BRDM2), (g) towed howitzer (D20), (h) armoured reconnaissance vehicle (MTLB), (i) anti-aircraft weapon (ZSU23-4), and (j) self-propelled howitzer (2S3). We observe that many of the vehicles are very similar in appearance which makes the target recognition task more difficult.

Table 1: Comparison of target recognition classifiers. The QMLPNN with quadratic layer has 7.3% increase in accuracy over the basic MLPNN. Similarly, the QCFCNN with quadratic correlation filter layer has an increase in accuracy of 2.32%. The QMLPNN has a 39.46% increase in accuracy and the QCFCNN has a 40.03% increase in accuracy over the baseline linear SVM (without retraining).

<table>
<thead>
<tr>
<th>Method</th>
<th>Training</th>
<th>Iter/Tol</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>QMLPNN</td>
<td>L-BFGS</td>
<td>1000 iter</td>
<td>0.9813</td>
</tr>
<tr>
<td>MLPNN [24]</td>
<td>L-BFGS</td>
<td>1000 iter</td>
<td>0.9083</td>
</tr>
<tr>
<td>QCFCNN</td>
<td>AdaGrad</td>
<td>4680 iter</td>
<td>0.9870</td>
</tr>
<tr>
<td>CNN [4]</td>
<td>AdaGrad</td>
<td>4680 iter</td>
<td>0.9638</td>
</tr>
<tr>
<td>SVM [6]</td>
<td>C-SVC</td>
<td>0.001 tol</td>
<td>0.5867</td>
</tr>
</tbody>
</table>

We conducted experiments to compare our methods for target recognition with the standard convolutional neural network (CNN) [4], a conventional MLPNN [24] and a linear support vector machine (SVM) [6]. Table 1 shows the results from our simulations, where we use 1000 iterations for QMLPNN and MLPNN, 4680 iterations (or 20 epochs) for QCFCNN and CNN, and a tolerance of 0.001 in the libsvm MATLAB software package for the linear SVM [5]. From Table 1 we learn that substitution of a quadratic layer causes an increase in accuracy from 90.8% for the standard MLPNN to 98.1% for the QMLPNN, and a jump in accuracy from 96.38% for the standard CNN to 98.7% for the QCFCNN.
Figure 5: Some correctly classified examples: QCFCNN output class label map and labeled image for (a)-(b) main battle tank (T72); (c)-(d) pickup truck (PKP); (e)-(f) self-propelled howitzer (2S3); (g)-(h) sport utility vehicle (SUV); (i)-(l) armoured personnel carrier (BMP2); (m)-(p) infantry scout vehicle (BRDM2); (q)-(t) armoured personnel carrier (BTR70).
Figure 6: Some mislabeled Examples: QCFCNN output class label map and labeled image for (a)-(b) BRDM2 labeled as BTR70; (c)-(d) BMP2 labeled as BRDM2; (e)-(f) BTR70 labeled as BRDM2; (g)-(h) BTR70 labeled as BMP2; (i)-(j) SUV labeled as PKP; Most misclassifications are due to the similarity between target classes.

Figure 7: (a) Quadratic multi-layer perceptron neural network (QMLPNN) and (b) Quadratic correlation filter convolutional neural network (QCFCNN) confusion matrices for target recognition with the ATR Algorithm Image Database [23] dataset. As can be seen in both matrices, the target recognition accuracy is very high. However, in both matrices, there is some probability of misclassification of the D20 and MTB (MTLB) target types as seen in the lower right corner.
5. Conclusions

The single-layer perceptron QCF has satisfactory performance for two-target discrimination as demonstrated in [14]. The proposed QCFCNN and QMLPNN, that integrate QCF with multi-layer neural networks, have very high multi-target recognition rate for mid-wave infrared images. Multiple applications of the QMLPNN or a single application of the QCFCNN can be used for image classification. The quadratic filter layer in each helps with separating target areas from background and localization, whereas the top-most layers assist with classification. This makes these networks ideal for ATR applications.

In this work we have restricted the target range to 3000 meters. It would be interesting to see how the embedded QCF handles targets of different scales. Another area for research would be to evaluate these methods on other types of sensor data (e.g. long-wave infrared (LWIR), short-wave infrared (SWIR), synthetic aperture radar (SAR)).

References