Comparison of Neural Network Classifiers for Automatic Target Recognition

Mark Carlotto[1], Mark Nebrich, and David Ramirez
General Dynamics Mission Systems

Abstract

We consider a challenge problem involving the automatic detection of large commercial vehicles such as trucks, buses, and tractor-trailers in Quickbird EO pan imagery. Three target classifiers are evaluated: a "bagged" perceptron algorithm (BPA) that uses an ensemble method known as bootstrap aggregation to increase classification performance, a convolutional neural network (CNN) implemented using the MobileNet architecture in TensorFlow, and a memory-based classifier (MBC), which also uses bagging to increase performance. As expected, the CNN significantly outperformed the BPA. Surprisingly, the performance of the MBC was only slightly below that of the CNN. We discuss these results and their implications for this and other similar applications.

Introduction

Challenge problems with crowd-sourced solutions are becoming increasingly popular in the machine learning community. We describe a specific problem of interest for automatic target recognition – the detection of vehicles in complex cluttered environments, specifically large commercial vehicles such as trucks, buses, and tractor-trailers in overhead imagery such as Quickbird EO pan imagery. Our problem is simpler in scope than NGA's recent xView detection challenge[2] but arguably more challenging in terms of the complexity of the background clutter.

Two of the three classifiers evaluated in this study currently operate within GD's Image Data Conditioner (IDC), which is a hybrid ATR architecture that combines model-based detection, segmentation, and classification algorithms with machine learning. ATR algorithms use 3-D geometrical models that represent objects of interest in terms of their size and shape to find possible instances of those objects in the image. IDC then uses neural networks to filter detections based on their appearance. Convolutional neural networks (CNN) contain layers that learn shift, rotation, and scale invariance. The model-based component of IDC generates "normalized" chips that are centered on possible objects of interest, rotated so that object is oriented horizontally, and scaled to be a fixed size in pixels[3] as shown in Figure 1. Normalization significantly reduces the complexity of neural

---

[1] Mark.Carlotto@gd-ms.com
[2] "The Pentagon Wants Your Help Analyzing Satellite Images," *Wired*, Feb 21, 2018.
See: https://www.wired.com/story/the-pentagon-wants-your-help-analyzing-satellite-images/
[3] As a result of scaling, the pixel resolution (meters/pixel) varies from chip to chip.

network training and allows IDC to support non-CNN architectures but can be affected by segmentation errors as seen in the figure.

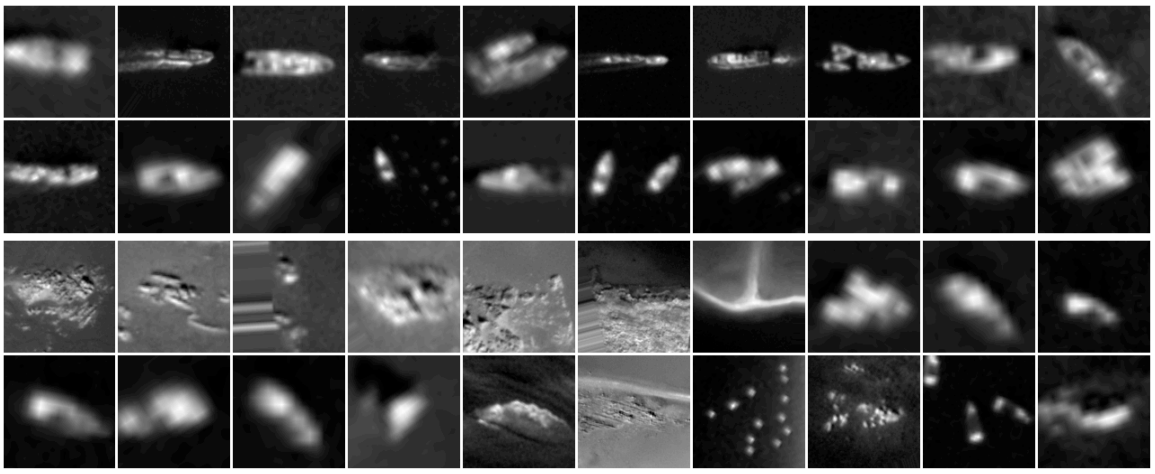

Figure 1 Example of training data used for ship classification. Normalized chips are automatically generated by IDC.

Bagged Perceptron Algorithm

The perceptron was the first neural network algorithm integrated into IDC for ship classification[4] (Figure 1). If the classes of interest are linearly separable, the perceptron will converge to a solution vector $\mathbf{w}$ such that

$$f(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{w} \cdot \mathbf{x} + \mathrm{b} > 0 \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where $\mathbf{x}$ is a feature vector computed from the chip. Being a simple, and relatively weak classifier, we implemented a method known as bootstrap aggregation or "bagging" to improve performance. Our "bagged" perceptron algorithm (BPA) computes K perceptrons by sampling the training set K times with replacement. Perceptron outputs are then aggregated:

$$F(\mathbf{x}) = \sum_{k} f_k(\mathbf{x}) \tag{2}$$

The aggregated output is divided by K to produce a score between zero and one that represents the target probability.

BPA used a set of biologically-inspired features modeled after those computed in the feed-forward path of the ventral stream in the primate visual cortex[5]. The ventral stream (also

[4] Mark J. Carlotto and Mark A. Nebrich, "Integrating Visual Learning Within a Model-based ATR System," *Proc. SPIE 10200, Signal Processing, Sensor/Information Fusion, and Target Recognition XXIV,* (April 2017).

known as the "what pathway") is involved with object and visual identification and recognition. This is in contrast to the dorsal stream (or, "where pathway") that is involved with processing the object's spatial location relative to the viewer[6]. The first two layers in the ventral stream consist of simple and complex neurons that act as a bank of spatially tuned Gabor filters (S1 layer) combined using max pooling (C1 layer). In our implementation, for a 128x128 pixel chip (receptive field), there are 2,088 features.

In order to understand the structure of the underlying data in various applications we use a nonlinear mapping algorithm[7] to visualize high dimensional feature spaces as 2D maps. Figure 2 (left) is a 2D color-coded visualization of the 2,088-dimenional S1-C1 feature vectors that were used to train an early version of the BPA ship classifier. The separation of the training data was good, which lead to a probability of correct classification Pcc = 0.94. As the size of the training set increased, the separation decreased (right), which resulted in a lower Pcc = 0.76.

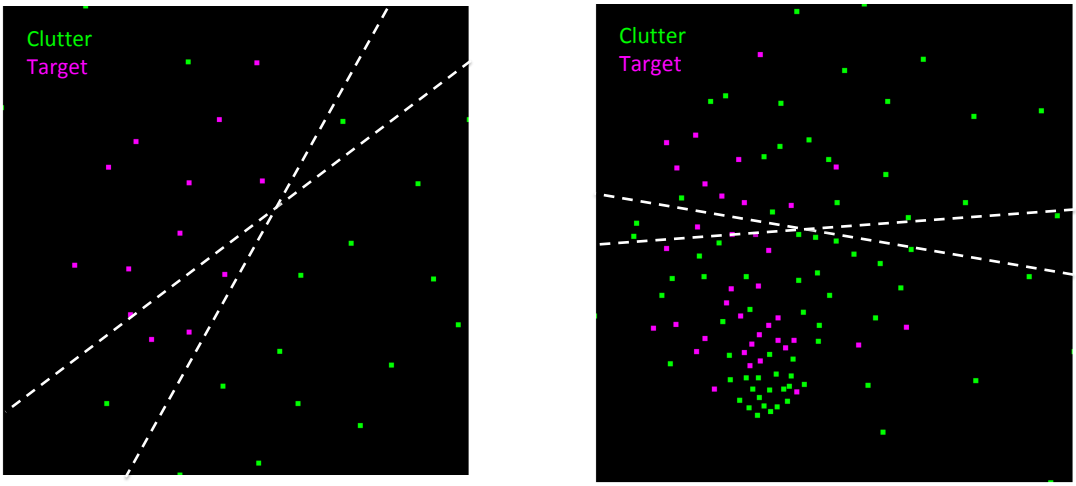

Figure 2 2D visualization of chip data used to train BPA ship classifier. Dotted lines depict decision boundaries computed by the algorithm for 29 and 111 target and clutter exemplars (left and right, respectively).

Memory-Based Classifier

The linear classifier appeared to be adequate for ship classification when the number of features exceeded the number of exemplars[8]. As the training set grew, the performance of the BPA began to decrease. This became even more evident as we began to address the

---

[5] T. Serre, M. Kouh, C. Cadieu, U. Knoblich, G. Kreiman, and T. Poggio, "A Theory of Object Recognition: Computations and Circuits in the Feedforward Path of the Ventral Stream in Primate Visual Cortex," *AI Memo 2005-036*, Massachusetts Institute of Technology, Cambridge, MA, December 2005.
[6] Two-streams hypothesis, *Wikipedia*, see https://en.wikipedia.org/wiki/Two-streams_hypothesis.
[7] Mark J. Carlotto, "Nonlinear mapping algorithm and applications for multidimensional data analysis," *Journal of Visual Communication and Image Representation*, Vol. 4, No. 3, Sept. 1993.
[8] T.M. Cover, "Geometrical and Statistical properties of systems of linear inequalities with applications in pattern recognition". *IEEE Transactions on Electronic Computers*, 1965.

challenge problem (Figure 3). The difficulty of separating vehicles from manmade clutter is evident in the distribution of target and clutter feature vectors from just a portion of the training set as shown in Figure 4.

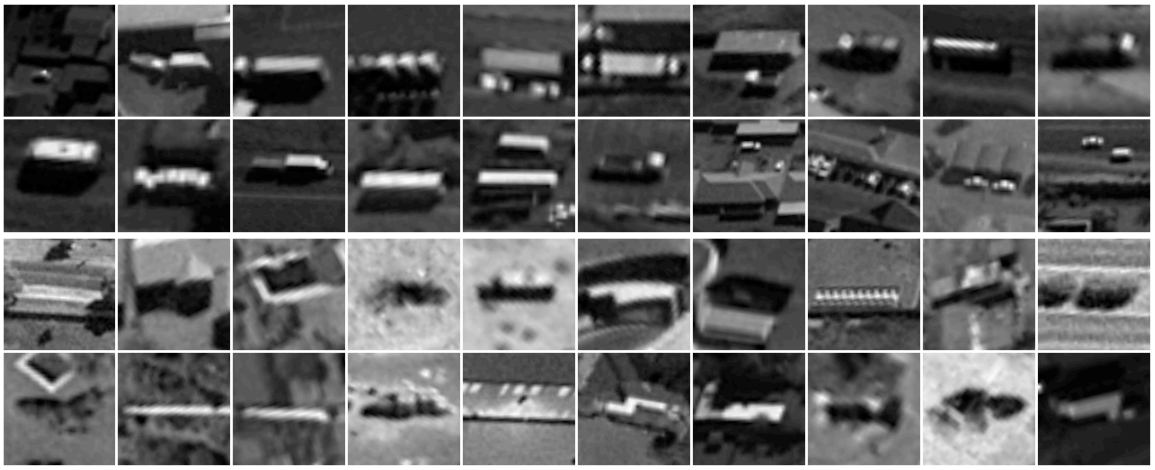

Figure 3 Preliminary training set for vehicle classification challenge problem. 20 of 237 target chips (top) and 20 of 300 clutter chips (bottom).

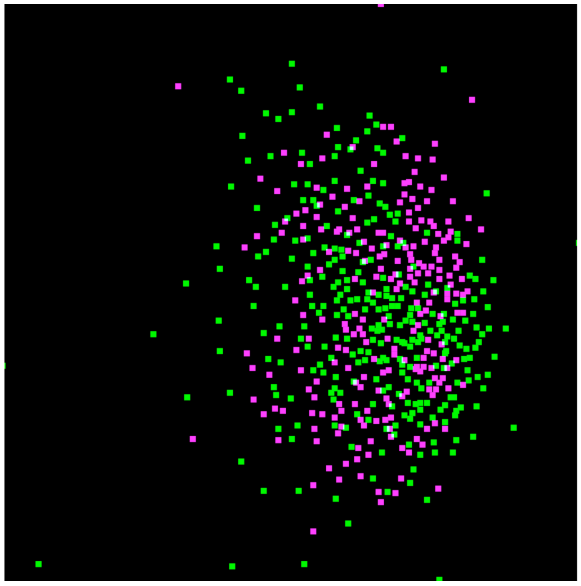

Figure 4 Distribution of feature vectors of 237 target chips and 300 clutter chips.

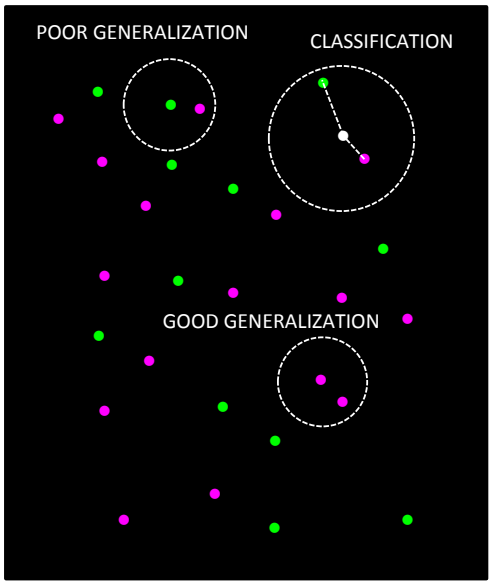

Figure 5 Memory-based classification assigns the class of the nearest exemplar.

The complexity of the underlying feature space motivated a new approach to classification based on the concept of memory-based reasoning[9]. The memory-based classifier (MBC) is,

[9] Craig W. Stanfill, "Memory-Based Reasoning Applied to English Pronunciation," *AAAI-87*.

in essence, a non-parametric minimum distance classifier that compares the features $\mathbf{x}$ of an unknown chip to those of all exemplar chips and assigns the class y of the nearest chip

$$f(\mathbf{x}) = y_m | \, m = \text{argmin} \|\mathbf{x} - \mathbf{x}_n\| \; \forall \; n \in T \tag{3}$$

where $y_m = \{0,1\}$ and T is the training set consisting of M target and clutter exemplars $\{\mathbf{x}_m, y_m\}$. Unlike the perceptron algorithm, the generalization performance of the MBC depends on the underlying data distribution (Figure 5). Good generalization occurs in places where there are clusters of points with the same class. MBC generalizes poorly in places where target and clutter classes are interspersed.

A leave-one-out test provides an estimate of MBC performance by comparing classes of neighboring exemplars. Define T_m to be the training set excluding the m-th exemplar. Let

$$m^* = \text{argmin} \|\mathbf{x}_{\boldsymbol{m}} - \mathbf{x}_n\| \; \forall \; n \in T_m \tag{4}$$

be the index of the vector that is nearest to the m-th exemplar. If

$$\delta(m) = \begin{cases} 1 & y_m = y_{m^*} \\ 0 & y_m \neq y_{m^*} \end{cases} \tag{5}$$

then

$$\widetilde{P_{cc}} = \frac{1}{M} \sum_m \delta(m) \tag{6}$$

is an estimate of the classification performance. The idea of leaving one out leads to a bootstrapped version of the MBC, which like the BPA, combines the outputs from multiple classifiers

$$f_k(\mathbf{x}) = y_m | \, m = \text{argmin} \|\mathbf{x} - \mathbf{x}_n\| \; \forall \; n \in T_k \tag{7}$$

constructed from subsets of the training set T_k using Eq. 2, where the T_k are generated by randomly selecting a specified fraction of the training set.

Using a single MBC and all of its training data is analogous to an "overfitted" CNN, neither of which will perform well outside of the training data set. Bootstrapping (Eq. 7) selects a fraction of the training set that forces the classifier to find nearest neighbors that are farther away. Together with aggregation (Eq. 2) bagging generalizes the performance of the MBC. Decreasing the sampling fraction reduces the density of points and effectively increases the size of the neighborhood and the amount of generalization.

Challenge Problem

The challenge problem data set is divided into separate training and test subsets. The training data contain 7168 chips of vehicles and background clutter from scenes over two geographical areas: South Africa (814 target and 4583 clutter chips) and Afghanistan (199 target and 1572 clutter chips). The test data contains 1037 chips from another scene over a third geographical area: Russia (483 target and 554 clutter chips). Clutter includes natural landforms (e.g., trees, drainage patterns, bodies of water, etc.) and manmade objects such as roads, buildings, and other structures. Target chips include isolated vehicles, multiple vehicles next to one another, and vehicles embedded in complex backgrounds; e.g., parked next to a building.

A benchmark classifier was implemented in TensorFlow[10] using the MobileNet[11] CNN architecture. The MobileNet architecture was adjusted with the model shrinking hyperparameter α to scale the number of convolutional kernels at each layer. This adjustment to model width was done to optimize performance for this particular challenge problem. The training dataset was augmented using the Keras[12] ImageDataGenerator. Different combinations of training, validation, and test data sets were evaluated.; e.g., using the South Africa (SA) data to train the classifier, the Afghanistan (AF) data to validate the classifier, and the Russia (RUS) data to independently test the classifier. Table 1 summarizes classification accuracy at a decision threshold value of 0.5 for all data set combinations.

Table 1 CNN classification accuracy on different combinations of training, evaluation, and test datasets. Data sets are over South Africa (SA), Afghanistan (AF), and Russia (RUS).

Case	Training Data		Evaluation Data		Test Data	
1	SA	0.82	AF	0.82	RUS	0.75
2	SA	0.87	RUS	0.81	AF	0.75
3	AF	0.76	SA	0.75	RUS	0.66
4	AF	0.79	RUS	0.72	SA	0.66
5	RUS	0.74	SA	0.70	AF	0.64
6	RUS	0.73	AF	0.75	SA	0.73

We discovered that the validation accuracy peaked very quickly in the training process (Figure 6). This is typically a sign of overfitting due to an oversized architecture versus an undersized dataset. It was theorized that stretching out this training period would slow overfitting, and improve final performance. The width scale factor of the MobileNet architecture was scaled to a minimum size, but this did not improve final trained performance of the system.

---

[10] "TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems," 2015.

[11] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," *ARXIV 2017*.

[12] François Chollet, et. al., "Keras: The Python Deep Learning Library, 2015, " See: https://Keras.io.

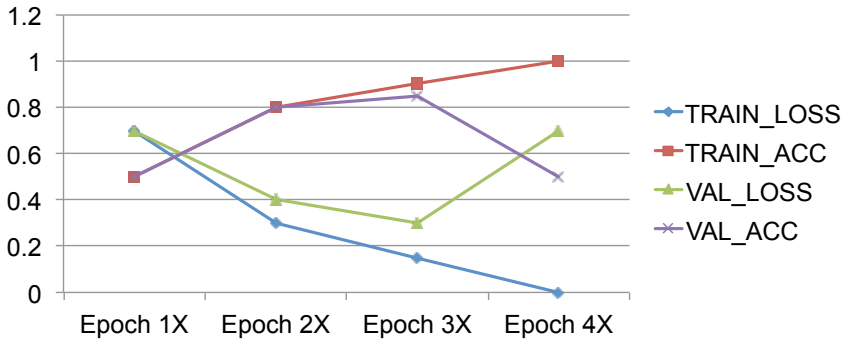

Figure 6 Typical training statistics for neural network training. Stage 1 shows rapid performance gains. Stage 2 shows slowed performance gains. Stage 3 shows overfitting of training set, and decreased performance on validation set.

BPA and MBC were tested using the published Gabor filter tunings (filter sizes) described by Serre[5] et al

GaborLR: {5,7,9,11,13,15,17,19,21,23,25,27,29,31,33,35}

and an alternate set of tunings

GaborHR: {3,5,7,9,11,13,15,17,19,21,23,25,27,29,31,33}.

which produce 6,148 features from a 128x128 pixel chip.

We also tested the two classifiers using power spectral density (PSD) features derived from the discrete Fourier transform of the chip with a cosine window. For a 128x128 chip, there are 64x64 = 4,096 unique power spectral values. Since the number of clutter chips was much greater than the number of target chips, we augmented the training set by generating permutations of the target chips with constrained random translations, rotations, and scales as was done with the CNN.

Figure 7 summarizes the receiver operating characteristic (ROC) performance of the challenge problem classifiers. All three classifiers were trained/validated on the SA and AF data sets and tested on the RUS data set. The CNN had the best performance (top curve), which was much lower than expected (for reasons that will be discussed later). The MBC classifier using PSD features was slightly below that of the CNN, followed by MBC using GaborHR and GaborLR features, respectively. The best MBC performance was achieved using a sampling fraction of 0.01 and a bagging parameter K=1000. The performance of BPA for all feature options was significantly below that of the MBC.

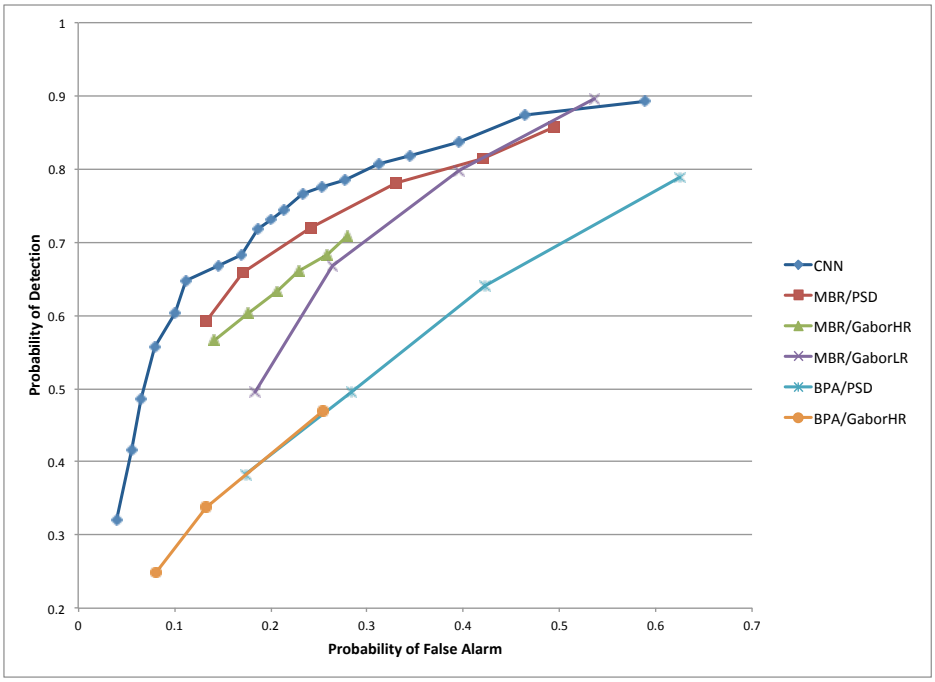

Figure 7 Challenge problem neural network classifier performance summary.

Discussion

It is conjectured that the behavior of the MBC mimics that of the CNN, at least for this problem. Although it not necessarily a practical consideration, one wonders if it would be possible to emulate (or at least approximate) the behavior of a CNN with a MBC containing all of the training data and augmentations (random permutations).

The relatively poor test performance of the CNN is consistent with the pattern of its training performance that peaked very early in the training process (Figure 6). It is likely that the performance of the CNN and other classifiers was limited in this problem by the resolution (level of detail) of the data and in the similarity (in many cases) between vehicles and background clutter, especially small buildings.

Acknowledgments

The authors wish to thank David DeMichele and Lam Nguyen for their assistance in the experiments