Optimization of Adaboost Algorithm for Sonar Target Detection in a Multi-Stage ATR system

Tsung Han (Hank) Lin
University of California, San Diego, La Jolla, California 92122

Mentor: Dr. Thomas Lu
NASA Jet Propulsion Laboratory, Pasadena, California, 91109

ABSTRACT

JPL has developed a multi-stage Automated Target Recognition (ATR) system to locate objects in images. First, input images are preprocessed and sent to a Grayscale Optical Correlator (GOC) filter to identify possible regions-of-interest (ROIs). Second, feature extraction operations are performed using Texton filters and Principal Component Analysis (PCA). Finally, the features are fed to a classifier, to identify ROIs that contain the targets. Previous work used the Feed-forward Back-propagation Neural Network for classification. In this project we investigate a version of Adaboost as a classifier for comparison. The version we used is known as GentleBoost. We used the boosted decision tree as the weak classifier. We have tested our ATR system against real-world sonar images using the Adaboost approach. Results indicate an improvement in performance over a single Neural Network design.

Keywords: automatic target recognition, correlation, Adaboost, Gentleboost, texture, Texton filters, false alarm rate.

1. Introduction

In computer vision, important research has been done using artificial systems to extract information from images. This will allow the artificial systems to "see" and "understand" their environments. Jet Propulsion Laboratory (JPL) has developed an Automated Target Recognition system (ATR) for real-time image understanding under NASA and DoD funding. JPL's multi-stage ATR system is illustrated in Fig. 1. In the first stage, one of the key components, the Grayscale Optical Correlation (GOC) filter can instantly scan a 512 x 512 pixel image to identify small patches of ROIs in the image[1-6]. Then, in the second stage, a classifier processes the ROIs to identify the true target. This report focuses on the second stage to find a classifier that has a good balance between speed and accuracy. Previous attempts in reducing false-positive ROIs include applying Feed-forward Back-propagation Neural Network[2], Radial Basis Function Neural Network[3], and using various feature extraction techniques such as wavelet and Haar-like features. This report will present texture features and Adaboost learning methods to reduce the false-alarm rate.

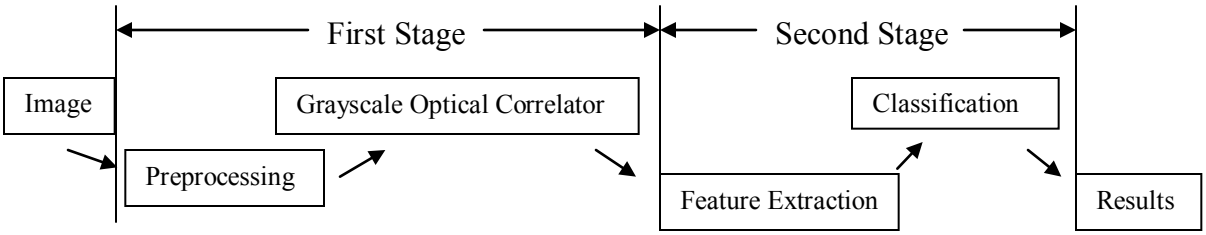


Figure 1. Multi-stage ATR system: Images are first preprocessed and fed into a Grayscale Optical Correlator, which contains the OT-MACH filter to quickly scan for possible ROIs. Next, PCA and Texton filters are used to extract features from each ROI. Finally, a Neural Network or an Adaboost classifierare used to identify true targets from the ROIs.

2. Texture image generation to detect edge feature

We generate our texture image using the Texton filters based on the techniques developed by Malik and Belongie[7]. We use an 18x18-pixel diagonal Texton filter to capture the edge features in the short-range sonar images, and use a 32x32-pixel vertical Texton filter for long-range sonar to capture the vertical shadow feature, as shown in Fig. 2. OT-MACH is capable of capturing the white speck in the long-range sonar, but is not capable of capturing the shadow of an object very well. Therefore, we focus the texture filter on capturing the shadows. Only one Texton filter is used in the ATR system due to processing time constraints.

For each image, the ATR system performs a number of operations before classification. We first preprocess the images by morphing segmented specks, and also apply medium filter to remove 'pepper and salt' noise. The preprocessed images are then scanned by OT-MACH filter, and we obtain many ROIs.

For each ROI, we make two copies, one copy will be processed with Principal Component Analysis (PCA), and the other copy will be processed with a Texton filter and then PCA. The features from both copies are then combined to form the feature vector of the ROI. In sum, each ROI contains features from both the original and the texture images. All together, the processing takes about 3- 5 seconds per image.

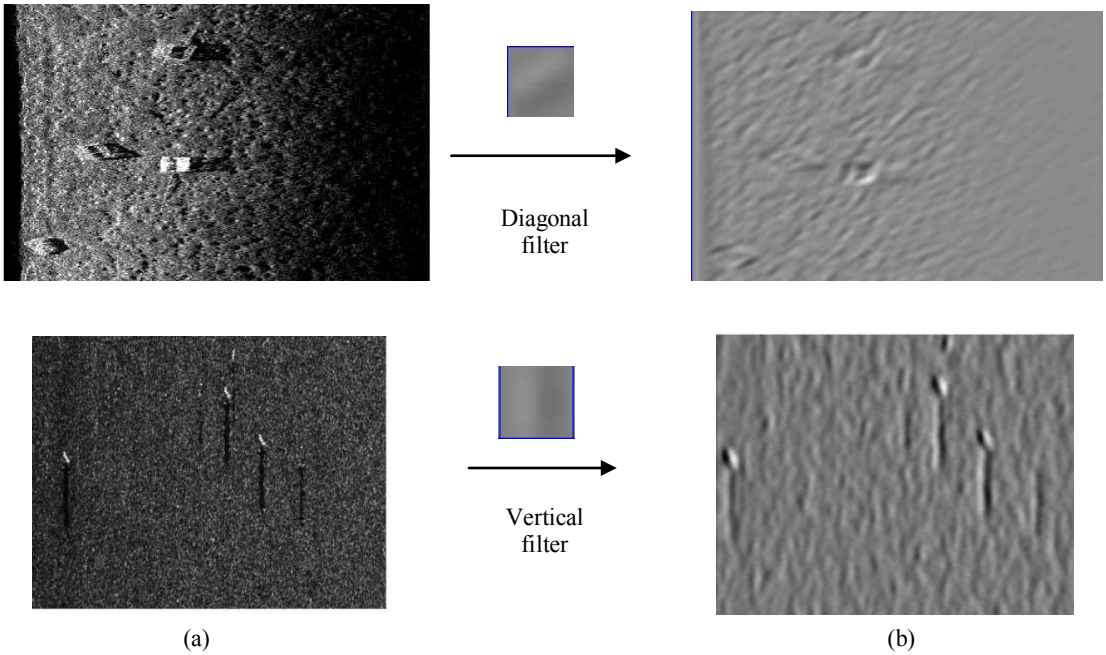


Figure 2. Texture images generation. (a) short-range sonar with diagonal Texton filter and (b) long-range sonar with vertical Texton filter.

3. Adaboost Learning for Target Classification

In this research project, we intend to improve ATR classification by implementing an Adaboost classifier. We have tested two variations of this classifier; RealBoost[8] and GentleBoost[9]. Our research demonstrates that either of these variations has enhanced the performance of an ATR system. GentleBoost was adopted for use in our ATR system because it is more resilient to noisy sonar data compared to RealBoost. GentleBoost algorithm is described in Fig. 3.

Gentle AdaBoost

1. Start with weights $w_i = 1/N, i = 1, 2, \ldots, N, F(x) = 0$.
2. Repeat for $m = 1, 2, \ldots, M$:
   (a) Fit the regression function $f_m(x)$ by weighted least-squares of y_i to x_i with weights w_i.
   (b) Update $F(x) \leftarrow F(x) + f_m(x)$.
   (c) Update $w_i \leftarrow w_i \exp(-y_i f_m(x_i))$ and renormalize.
3. Output the classifier $\text{sign}[F(x)] = \text{sign}[\sum_{m=1}^{M} f_m(x)]$.

Figure 3. GentleBoost Algorithm

We are given a training set $S = < (x_1,y_1),\ldots(x_N,y_N)>$ where x_i are the feature vectors of ROIs, and binary labels $y \in \{-1,+1\}$, where +1 is for the target and -1 for the non-target. We start by making an equal distribution to the sample, each i_{th} sample assigned with weight 1/N. The GentleBoost algorithm updates the distribution each round. For each round, we build a decision tree to classify the samples, and update the weight of each sample according to step 2 (see Fig 3). Here $f_m = P_w(y=1|x) - P_w(y = -1|x)$, and in the end of each round the samples that are misclassified have higher weights than those of correctly classified samples. In the next round, the decision tree focuses on classifying correctly the sample with higher weights. After M rounds, each decision tree has a weight f_m, with more accurate trees having higher weights. Finally, the decision trees take a vote for an overall classifier, as shown in step 3 of Fig 3. As a result, the GentleBoost algorithm builds a weak classifier each round, and this weak classifier focuses on classifying correctly the samples that are misclassified in the last round. The weak classifiers with lower classification error will have higher weights and more influence on the overall votes for the final classifier.

The final output in step 3 is further modified. We want to count the ROIs with high confidence values as true targets. Instead of counting all ROIs with greater than 0 outputs as true targets, we send the outputs to a tan-sigmoid function, mapping the output between 0 and 1. Confidence values above a certain threshold (which we set at .95) are counted as true targets. We have found that counting ROIs with high confidence values produce better results.

$2 \log (P_w(y = 1|x)) / (P_w (y = -1|x)$. The log-ratios of RealBoost can result in weights of samples that may increase very rapidly without bound. Imagine an outlier data that is misclassified, and after each round a new weak classifier is trained to focus on the outlier data, and after many rounds we have many poor weak classifiers. GentleBoost is designed to put a bound on the weight of the sample, namely in the range of [-1,1], which grows more slowly than e^x of RealBoost. The bound will help the GentleBoost to not 'chase' after noisy data, and thus it is more resistant to noise than RealBoost.

We use a Feedforward Backpropagation Neural Network as the basis for comparison against GentleBoost. The Neural Network contains 2 hidden layers- first layer with 20 neurons, second layer with 10 neurons- and both layers use the tangent-sigmoid transfer function. Training is done for 30 runs, and the best Neural Network is selected for comparison.

4. GentleBoost Classification Results

4.1 Short-range sonar training and testing sets

Here we present the testing of multi-stage ATR system on short-range and long-range sonar images. Both GentleBoost and Neural Network are compared for their accuracy. For short-range sonar, we present two test sets. The first set is called 'Sonar,' with 223 images, and is used primarily for training. The second set, 'Sonar2,' contains 450 images, and is used primarily for testing.

Sample	% of Training	% of True ROIs	% of True ROIs used for training	% of False ROIs used for training

Sonar (32% vs 6%, 223 images, 6400 ROIs)	**8%**	**5%**	**32%**	**6%**
Sonar (72% vs 8%, 223 images, 6400 ROIs)	**12%**	**5%**	**72%**	**8%**
Sonar2 (450 images, 13100 ROIs)	No training (testing only)	No training (testing only)	No training (testing only)	No training (testing only)

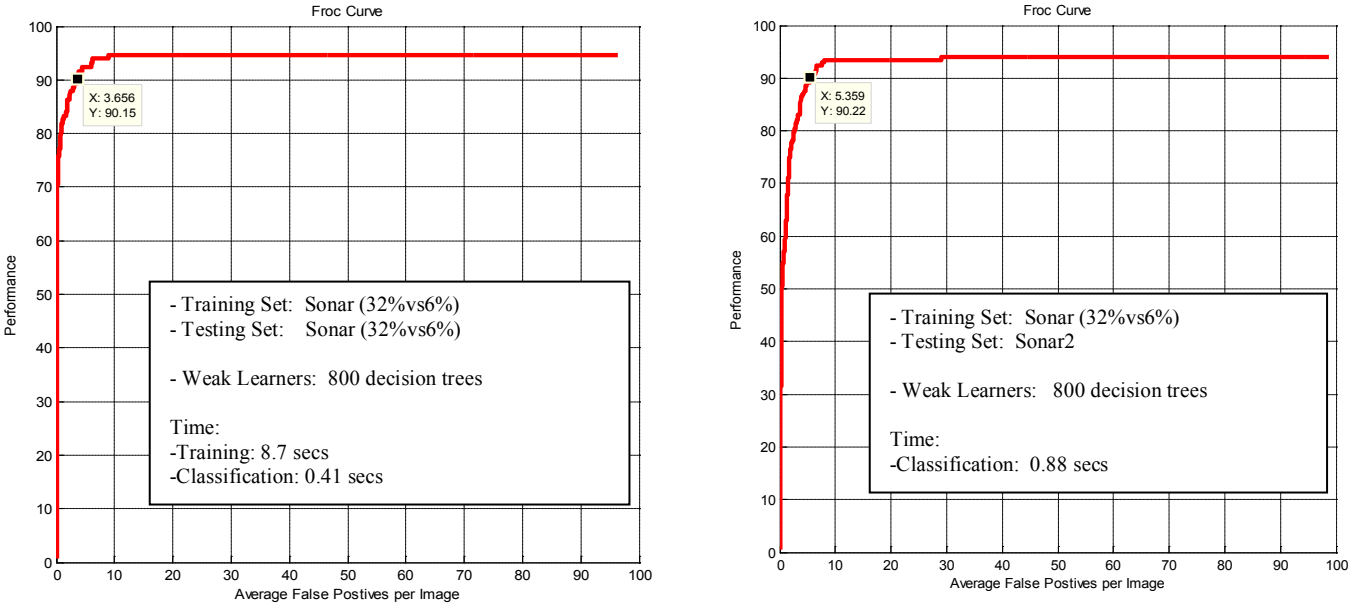
Table 1. Training set for short-range sonar. The size of a training set is kept to a minimum, while continuing to be representative. Notice true ROIs used for training are 32% and 72%. False hits used for training are very low, less than 8%.

We use a FROC curve as a way to benchmark the results. On the y-axis, 'Performance,' indicates the percentage of targets that are caught. Targets are considered caught if one or more ROIs cover the target. On x-axis, the number of average false-positive ROIs detected per image is shown. Any ROI not covering a target is counted as false-positive. We want the curve to be as much toward upper-left corner as possible, with the perfect curve being 100% target detection rate with 0 false-positive.

A good classifier would maintain generalization. In our testing we start with minimal training set and weak learners, and gradually increase the training set and weak learners. In each step, we check if the detection rate increases and if GentleBoost maintains generalization.

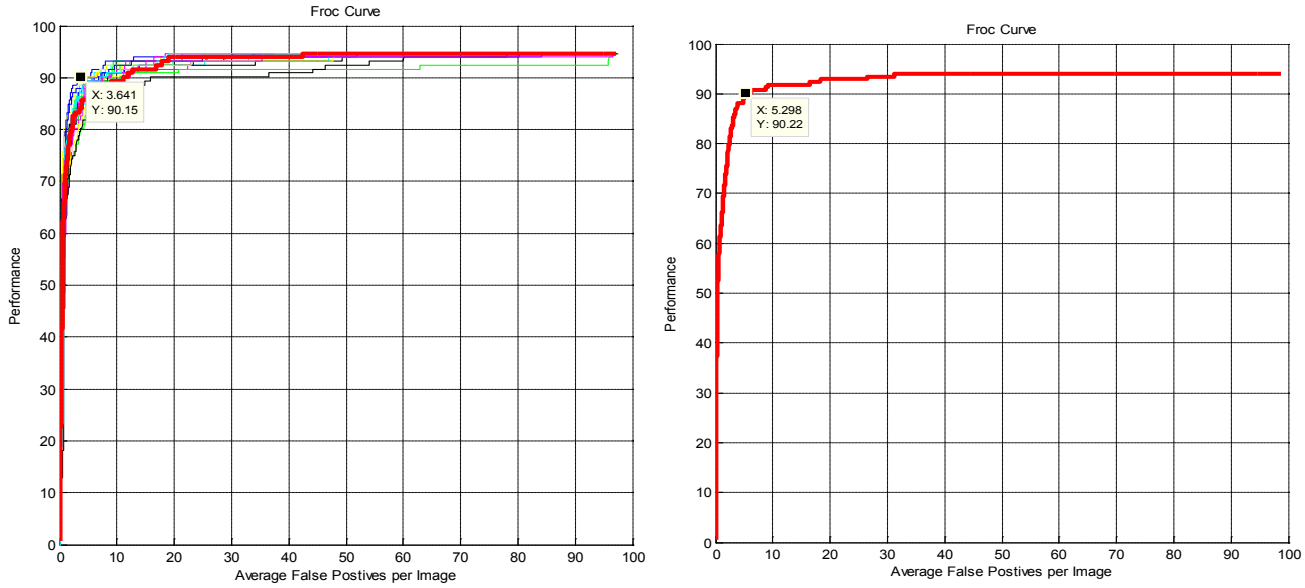
4.2 GentleBoost with 800 decision trees

In the first test, we use first set, 'Sonar,' for training, with 32% true training examples and 6% false training examples, and then classify on the same set. GentleBoost uses 800 decision trees, and even with these many learners, GentleBoost took only 8.7 seconds to train and 0.41 seconds to classify all 6400 ROIs. We obtain 90% catch rate for 3.6 false-positives per image. Next, we test on a second test set called 'Sonar2'. This second test checks if generalization is maintained. We obtain 90% performance for 5.3 false-positives per image (see Fig 4). Results for both sets are similar to those of the Neural Network that was already in the ATR system, (see Fig 5).



(a) (b)

Figure 4. GentleBoost initial result using 800 decision trees. (a) For the first set 'Sonar', the catch rate is 90% for 3.6 average false-positives per image. (b) For second set 'Sonar2,' catch rate is 90% for 5.3 average false-positive per image. The result is similar to that of Neural Network.



(a) (b)

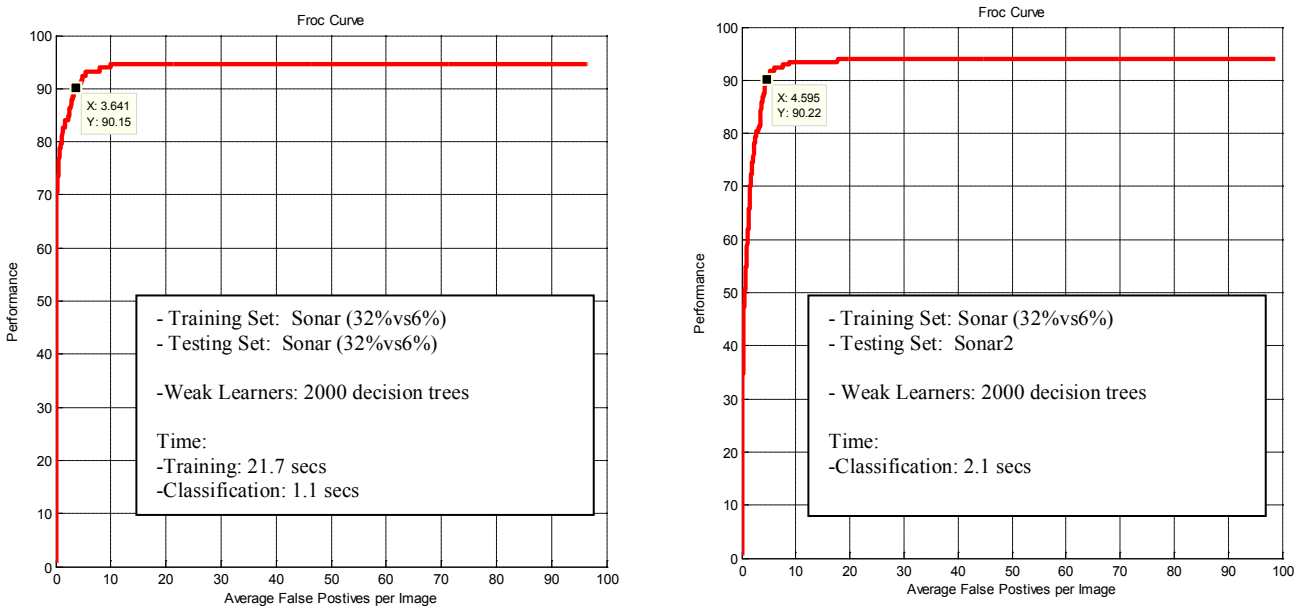
Figure 5. Neural Network result for short-range sonar, used as a benchmark (a) With first set 'Sonar,' with 32% true and 6% false training examples, we obtain 3 false-positives per image for 90% catch rate after 30 trainings. Using the best Neural Network out of 30 trainings, we classify the second set 'Sonar2' (b) and obtain 5 false-positives per image for 90% catch rate.

4.3 GentleBoost with 2000 decision trees does not overfit

Next, we seek to improve GentleBoost. Increasing either the number of weak learners, or the number of training sets may help improve GentleBoost, but it may also become harder to maintain generalization. We first increase the number of decision trees from 800 trees to 2000 trees, train with first set with 32% true and 6% false targets, and run on both test sets. For the first set 'Sonar,' we obtain the same result as for 800 trees, 90% catch rate for about 3 average false-positives per image. However, for the second set, 'Sonar2,' we have an improvement from 5 false-positives down to 4 false-positives per image for 90% catch rate, as shown in Fig. 6. Even though we more than double the number of trees, GentleBoost is still able to maintain generalization and improve its performance. The result supports that boosting is resistant to overfitting [9].

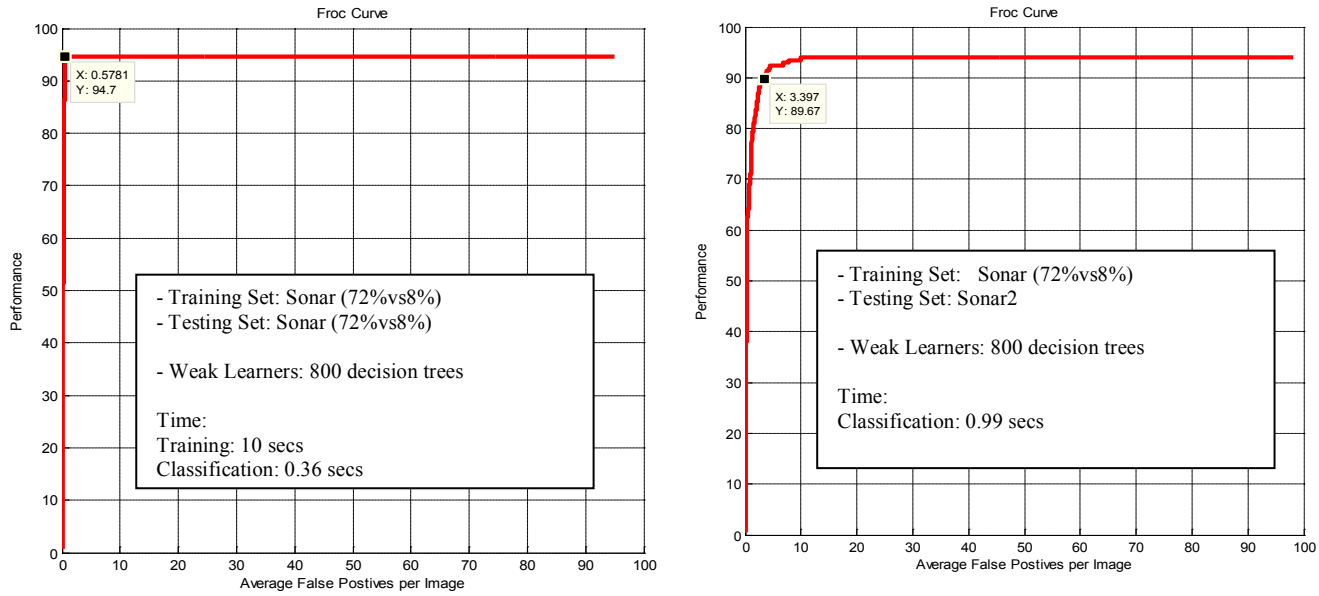
4.4 GentleBoost with more training set does not overfit

Next, we increase the training set. In the first set, the training examples are increased from 32% true vs 6% false targets to 72% true vs 8% false targets. The increase in training examples is set to a point for GentleBoost to obtain perfect classification, fig 10. Note the optimal catch rate is only up to 94% and not 100%. The 6% targets are missed by the first stage OT-MACH filter, and thus no ROIs are covering those targets for classification. When we run the new GentleBoost with increased training sets on the second test set 'Sonar2', we see an even further improvement; from 5% false-positives down to 3 false-positives for 90% catch rate, shown in Fig. 7. This result indicates that boosting maintains generalization of the data and thus is resistant to overfitting.



(a) (b)

Figure 6. Increasing from 800 trees to 2000 trees help improve performance for GentleBoost. Performance has improved when testing on second set 'Sonar2' (b), false-positives drop from 5 to 4 for 90% catch rate. Even with more rounds of training, GentleBoost maintains generalization, and this supports that boosting tends not to overfit.



(a) (b)

Figure 7. Increasing training examples for GentleBoost improve the performance. (a) Training examples from first set is increased to obtain perfect performance for GentleBoost. When test on second set (b), the performance has improved from 5 false-positives down to 3 false-positives. GentleBoost does not overfit.

4.5 Long-range sonar

Finally, we compare Neural Network against GentleBoost in long-range sonar images. The images are extremely noisy. The targets blend in well with the background and even for humans it takes some scrutiny to find the targets. Targets appear to have no particular shape, but are white specks with faint, thin, and long shadows below them. Many times non-targets appear similar to targets, and the difference only comes down to the length of the shadow. We use a data set called 'port_easy' for training. The examples in these images are easier to spot, with targets that slightly more distinguishable from the background. Next, we use port_starboard data for testing, which contains easy and hard target samples. The Neural Network used here contains 2 hidden layers, with 30 neurons for the first layer and 20 neurons for the second layer. GentleBoost contains 1600 decision trees. The result for Neural Network is 8 false-positives per image, while for GentleBoost it is only 2 false-positives. Both for 70% catch rate (see Fig. 8). Performance of GentleBoost exceeds that of the Neural Network greatly. This outcome demonstrates the superior classification potential of GentleBoost over Neural Network for a very difficult test set.

Sample	% of Training	% of True hits	% of True hits used for training	% of False hits used for training
Port_easy (79 images, 23108 hits)	**6.6%**	**7.7%**	**32%**	**4.5%**
Port_starboard (262 images, 76257 hits)	No training (testing only)	No training (testing only)	No training (testing only)	No training (testing only)

Table 2. Training set for high-range sonar. Port_easy is used for training GentleBoost and port_starboard is used for testing. Training sets are kept to minimum but representative, with only 32% of true targets and a low 4.5% of false targets used for training.

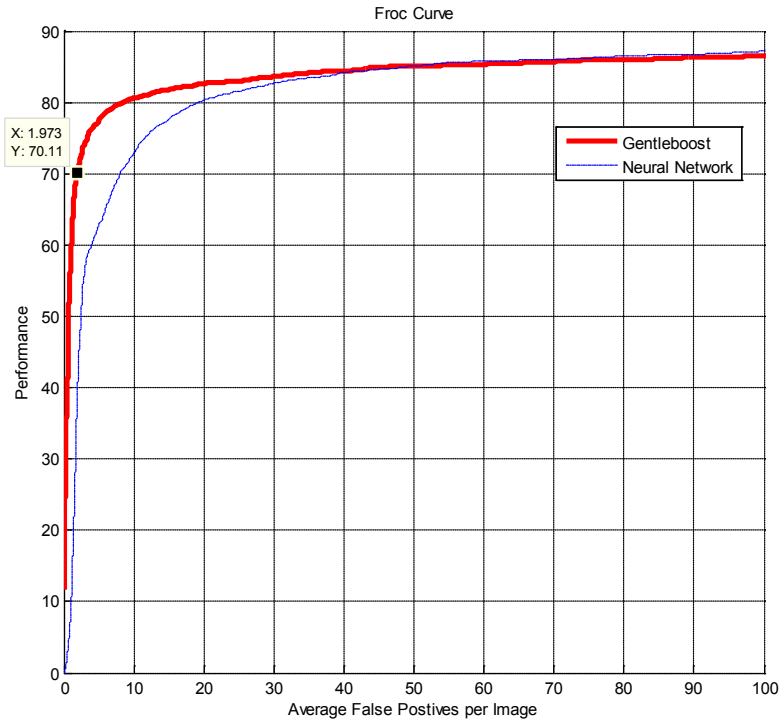


Figure 8. GentleBoost outperforms Neural Network in noisy long-range sonar. Port_starboard contains very noisy targets, which have no apparent shape, and have non-targets that look similar to the targets. Neural Network achieves 8 false-positives for 70% catch rate, while GentleBoost achieves 2 false-positives for 70% catch rate.

Conclusions

In this paper we have compared the Backpropagation Feedforward Neural Network with GentleBoost classifiers. Two real-world sonar image sets were used for testing. In both cases, GentleBoost outperformed the Neural Network. In the long-range sonar set, which contains very difficult images, the performance of GentleBoost greatly exceeds that of the Neural Network. This demonstrates its strong classification potential. Also, GentleBoost is found to be resistant to overfitting. In data set one (Sonar) with short-range sonar images, we greatly increased training examples and tree learners while training GentleBoost, yet GentleBoost continued to exhibit the ability to generalize during testing. In the short-range data set that contains 450 images, GentleBoost (with 2000 decision trees) took only 2 seconds to classify 13100 ROIs. This indicates that GentleBoost using linear regression, is promising for real-time classification. In the future, we will use GentleBoost to boost an ensemble of Neural Networks. Improvement in performance may be possible because the Neural Network is a non-linear classifier as opposed to decision tree, which is a linear classifier. Due to time constraints, we only used only one texture filter for short-range sonar. However, we can also present two texture filters instead of just one (say horizontal and vertical filter to help in discriminating power). Future work can also be done on optimizing the OT-MACH filter. To achieve real-time operability of the ATR system, more work can be done to reduce the processing time of GOC and texture filtering. Combined, these filtering protocols take 5 seconds to process one 5060x559 pixel image.

Acknowledgments

This research was carried out at the Jet Propulsion Laboratory, California Institute of Technology under a contract with the National Aeronautics and Space Administration (NASA), and was sponsored under the Undergraduate Student Research Program (USRP) through NASA.

REFERENCES

[1] Lu, T., Hughlett, C., Zhou, H., Chao, T. H., Hanan, J. C., “Neural network postprocessing of grayscale optical correlator,” Proc. SPIE 5908, 291-300 (2005).

[2] Ye, D., Edens, W., Lu, T., Chao, T., “Neural Network target identification system for false alarm reduction,” Proc. SPIE 7340, 77030B (2009).

[3] Lin,T.H., Lu,T., Braun H., Edens W., Zhang, Y, Chao, T., Assad, C, and Huntsberger, T, ″Optimization of a multi-stage ATR system for small target identification,″ Proc. SPIE 7696C, 76961Y (2010)

[4] Zhou H., Hughlett C. L., Hanan J. C., Lu T., Chao T-H.. “Development of streamlined OT-MACH-based ATR algorithm for grayscale optical correlator,” Proc. SPIE 5816 ,78-83 (2005).

[5] Chao, T., Zhou, H., and Reyes,G. , "Compact 512x512 Grayscale Optical Correlator", Proc. SPIE 4734, 9- 12 (2002).

[6] Johnson, O.C., Edens, W., Lu, T., Chao, T., “Optimization of OT-MACH filter generation for target recognition,” Proc. SPIE 7340, 734008 (2009).

[7] Malik, J., Belongie, S., Leung T. K., Shi, J., “Contour and texture analysis for image segmentation,” Int. Journal of Computer Vision 43(1), 7–27 (2001)

[8] Schapire,R.E., Singer,Y., “ Improved boosting algorithms using confidence-rated predictions,” Machine Learning 37(3), 297-336 (1999).

[9] Friedman, J., Hastie, T., and Tibshirani,R., “Additive logistic regression: A statistical view of boosting,” The Annals of Statistics 38(2), 337–374 (2000)