Machine Vision and Applications (2000) 11: 300–305

Machine Vision and Applications

LADAR target detection using morphological shared-weight neural networks

Mohamed A. Khabou[1], **Paul D. Gader**[2], **James M. Keller**[2]

[1] Department of Physics, Computer Science and Engineering, Christopher Newport University, Newport News, VA 23606, USA
[2] Department of Computer Engineering and Computer Science, University of Missouri-Columbia, Columbia, Mo, USA

Abstract. Morphological shared-weight neural networks (MSNN) combine the feature extraction capability of mathematical morphology with the function-mapping capability of neural networks in a single trainable architecture. The MSNN method has been previously demonstrated using a variety of imaging sensors, including TV, forward-looking infrared (FLIR) and synthetic aperture radar (SAR). In this paper, we provide experimental results with laser radar (LADAR). We present three sets of experiments. In the first set of experiments, we use the MSNN to detect different types of targets simultaneously. In the second set, we use the MSNN to detect only a particular type of target. In the third set, we test a novel scenario, referred to as the Sims scenario: we train the MSNN to recognize a particular type of target using very few examples. A detection rate of 86% with a reasonable number of false alarms was achieved in the first set of experiments and a detection rate of close to 100% with very few false alarms was achieved in the second and third sets of experiments. In all the experiments, a novel pre-processing method is used to create a pseudo-intensity images from the original LADAR range images.

Key words: Mathematical morphology – Neural networks – Automatic target recognition – LADAR

1 Introduction

Designing a system to automatically detect a certain type of object (target) in a scene is a challenging problem that has yet to be fully automated. Automatic target recognition (ATR) systems must (1) have the ability to suppress clutter and noise, (2) have a high detection rate and low number of false alarms, and (3) tolerate the target's image variability (e.g., rotation, translation, illumination, and partial occlusion) [Bhanu et al. 1983; Bhanu and Jones 1992; Roth 1990].

Many authors have considered ATR using different techniques and imagery/data sets [Bhanu et al. 1997]. Some ATR systems rely on forming prototypical examples (templates) of a target's variability and then performing template matching for target identification. Other ATR systems extract target features that are invariant to position/rotation and use them to classify possible targets. Many feature-based approaches rely on human vision and intuition for the identification of meaningful features. Sometimes this leads to unintentional biases, because some image modalities are fundamentally different than visual imagery (e.g., in synthetic aperture radar (SAR) imagery, resolution does not depend on the distance from the target) [Koch et al. 1995; Sajda et al. 1995].

Some ATR systems detect possible targets by comparing the statistical characteristics of the pixels in the target window to a pre-defined threshold or to the characteristics of pixels in an annulus window placed around the target. Such systems include the constant-false-alarm rate (CFAR) filter, the DEVLIN filter, and the LODARK filter [Moore 1998; Li and Zelnio 1996].

Another approach widely used in ATR is the use of correlation filters. For a certain input image, the output plane is computed by cross-correlating the input image with the designed filters. The ultimate goal of designing the filters is to produce a high output at the positions of the targets and very small output elsewhere. Examples of such filters include the synthetic discriminant function (SDF), and the minimum average correlation energy filter (MACE) [Casasent and Chang 1986; Casasent and Ravichandran 1992; Mahalanobis et al. 1987; Sims et al. 1994; Won 1995; Hobson et al. 1994].

Morphological shared-weight neural networks (MSNN) have been shown to outperform both MACE filters and standard shared-weight neural networks (SSNN) of identical structure in ATR applications (more detections and fewer false alarms) [Won 1995; Won et al. 1997]. The MSNN multi-sensor capabilities in ATR have been previously demonstrated using different imagery data sets in the visible and beyond-visible spectrums including forward-looking infrared (FLIR) [Won 1995; Won et al. 1997] and SAR [Theera-Umpon et al. 1998; Khabou et al. 1999].

In this paper, we provide results with laser radar (LADAR). We present three sets of experiments. In the first set of experiments, we use the MSNN to detect different types of targets simultaneously. In the second set, we use

Correspondence to: P.D. Gader (gader@cecs.missouri.edu)

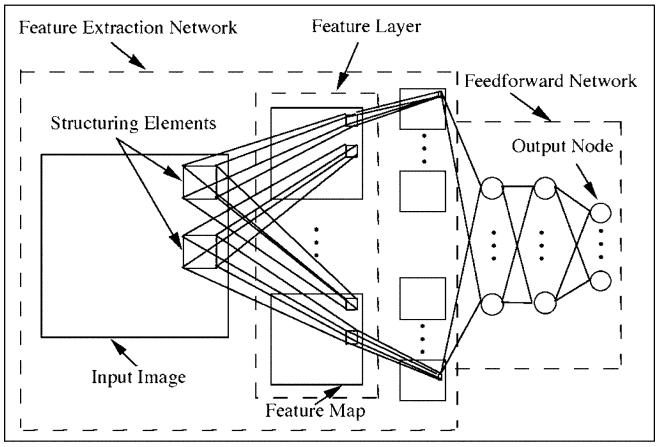


Fig. 1. Architecture of shared-weight neural networks

the MSNN to detect only a particular type of target. In the third set, we test a novel scenario that was suggested to us by Richard F. Sims, which we shall refer to as the Sims scenario. The scenario is the following: a reconnaissance-type flight acquires a very small number of images of a particular scene. These images are downloaded to a training program that is trained to recognize one or more particular objects in the scene. The training program must operate very quickly. The resulting ATR algorithm is downloaded into another system which returns to the physical scene. Of course, the objects in the scene may have moved and the angle of entry into the scene may be different. The ATR must detect examples of the object in the new images.

Our LADAR data set is structured to support this scenario. We trained the MSNN to recognize a particular type of object in a particular scene using very few examples (*one* or *two*). It then needed to detect the objects in other images from *different* viewing angles. Although the objects are very small and look very similar to the clutter, the results are good enough to indicate that this is a realistic approach to ATR.

This paper is organized as follows. First, we briefly describe the MSNN architecture, then we describe the LADAR data set and the novel pre-processing scheme we used in our experiments. Finally, we present the experimental results.

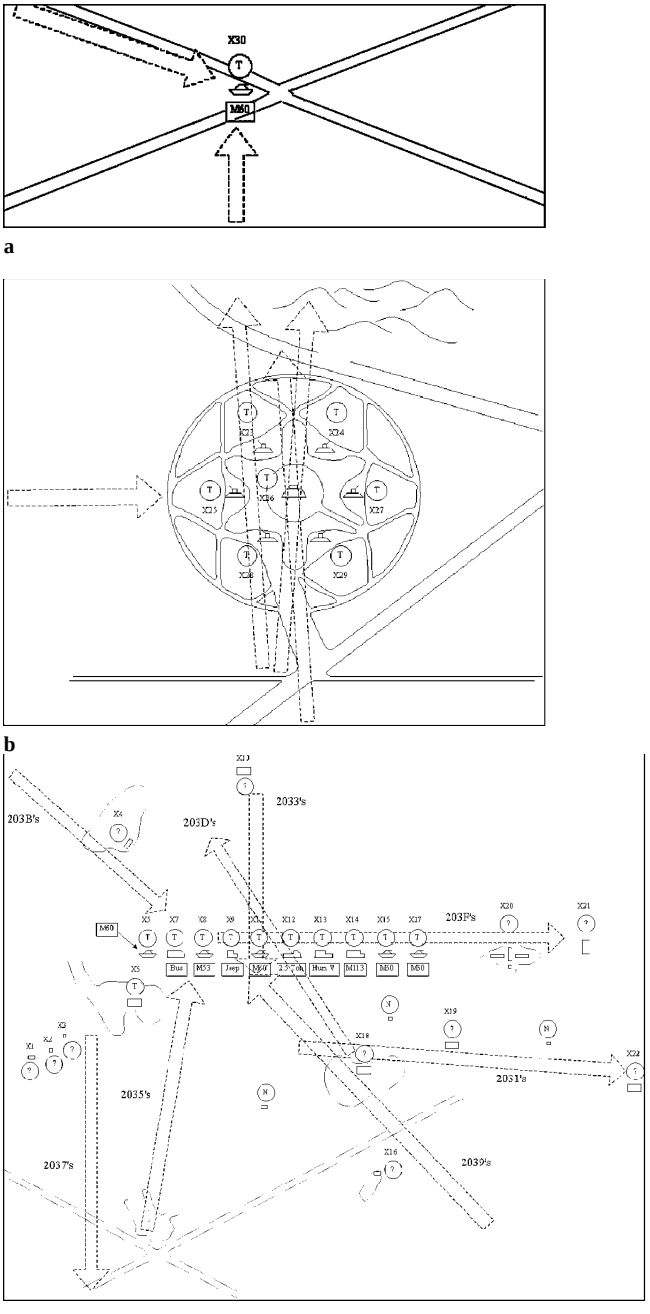


Fig. 2. Ground-truth layout of the **a** "crossroads", **b** "SAM site" and **c** "convoy" scenes showing the different flight paths (*dashed arrows*)

2 MSNN architecture

Before describing the MSNN architecture, we provide brief definitions of some gray-scale morphological operations. A full discussion can be found in [Dougherty 1992; Serra 1982, 1988; Haralick et al. 1997]. The basic morphological operations of erosion and dilation of an image f by a structuring element (SE) g are

$$erosion : (f \ominus g)(x) = \min\{f(z) - g_x(z) : z \in \mathrm{D}[g_x]\}, \quad (1)$$

$$dilation : (f \oplus g)(x) = \max\{f(z) - g_x^*(z) : z \in \mathrm{D}[g_x^*]\}, \quad (2)$$

where $g_x(z) = g(z - x)$, $g^*(z) = -g(-z)$ and $\mathrm{D}[g]$ is the domain of g. The gray-scale hit-miss transform is defined as

$$f \otimes (h, m) = (f \ominus h) - (f \oplus m^*). \quad (3)$$

It measures how a shape h fits under f using erosion and how a shape m fits above f using dilation. High values indicate good fits.

MSNN has the same architecture as SSNN, except that MSNN extract features using a hit-miss transform instead of linear convolution. MSNN is composed of two cascaded sub-networks, called stages: a feature extraction stage followed by a feed-forward classification stage. The feature extraction stage is composed of one or more feature extraction layers. Each layer is composed of one or more feature maps. Associated with each feature map, is a pair of structuring elements – one for erosion and one for dilation. The

values of a feature map are the result of performing a hit-miss operation with the pair of structuring elements on a map in the previous layer (see Fig. 1). The values of the feature maps on the last layer are fed to the feed-forward classification stage of the MSNN [Gader et al. 1994, 1995].

3 LADAR data set description

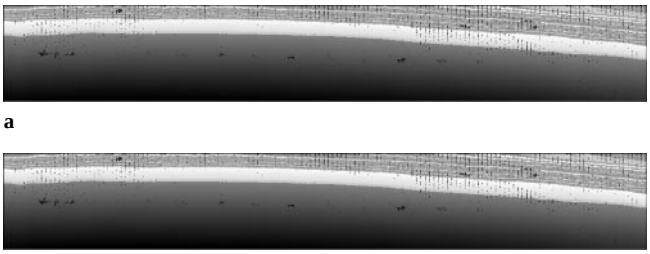
The LADAR data set was collected by an aircraft equipped with a LADAR sensor during 14 separate flights over several desert locations. Each flight produced 7 image data files, except for flight F203B, in which one data file was damaged, making a total of 97 images. The data set contains several images of three different scenes: "crossroads", "convoy" and "SAM site" (surface-to-air missiles) from different viewing angles. The "crossroads" scene contains a single M60 tank. The "convoy" scene contains a convoy of 10 identified targets, 1 unidentified target and many questionable target-like objects scattered throughout the scene. The "SAM site" scene contains 6 missile launchers positioned radially around the control center. The ground-truth layouts of the scenes are shown in Fig. 2, where the known targets are labeled with "T", the unknown objects are labeled with "?", and the flight directions are shown with dashed arrows. Since there are multiple passes over each scene, this data set allows us to simulate the Sims scenario.

For the first and second sets of experiments, the data is divided into training and testing sub-sets in the following manner. The odd-numbered images from each flight were included in the training set – except the images from flight F2053, which were all included in the testing set – and the rest of the images were included in the testing set. In this manner, we categorized the data into 52 training images containing a total of 89 targets and 45 testing images containing a total of 86 targets.

The LADAR data set came in three different configurations: range, intensity and height. In our experiments, we used a transformation of the range data, which we refer to as the pseudo-intensity data. The pseudo-intensity image enhancement accentuates the targets and features of the terrain. To create the pseudo-intensity images, the input LADAR range image is first inverted to make the targets brighter than the background, and then median filtered using a 3×3 neighborhood. The 3×3 Sobel edge masks are used to estimate the partial derivatives, g_x and g_y, at each point in the range image. The value of the pseudo-intensity image at a point is defined by:

$$p = \frac{255}{\sqrt{g_x^2 + g_y^2 + 1}} \,. \tag{4}$$

The rationale for referring to this image as pseudo-intensity is that the intensity of the return should depend on the relationship between the normal vector at a point and the sensor. If the normal vector points in the direction of the sensor, then the partial derivatives should be zero, and the pseudo-intensity attains a maximum value. This is intuitive since, if the LADAR impinge on a planar surface with normal vector pointing towards the sensor, we would then expect a maximum intensity return. As the normal vector rotates away from the direction of the sensor, the intensity of the return is reduced nonlinearly. Notice in Fig. 3 that the objects appear more defined in the pseudo-intensity images than they are in the range images.



a

b

Fig. 3a,b. A "convoy" scene image **a** before and **b** after pseudo-intensity enhancement

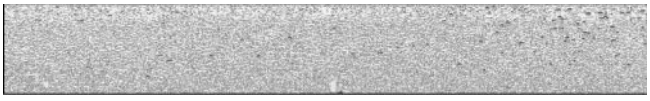


Fig. 4. Example of a "hard" scene that produced many false alarms. None of the "blobs" in this scene is a target

4 MSNN ATR results using a LADAR data set

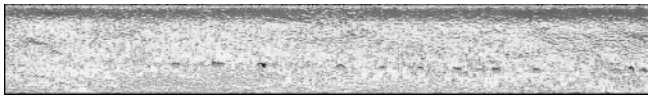
4.1 Training procedure

We used the "dynamic random selection" method described in [Won et al. 1997] to train the MSNNs. This training method has been shown to be effective in background suppression (i.e., reducing the number of false alarms) because it uses more background than target samples and it replaces samples once it learns them. The training begins with selecting random target and background sub-images from the training scenes. The pattern sum squared (PSS) error is measured for each sub-image at every epoch. If it is low for a certain sub-image, a new sub-image is randomly selected to replace it. The training is continued until the root mean squared (RMS) error attains a pre-defined level or the maximum number of epochs is attained.

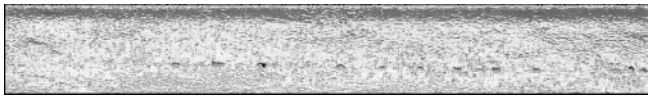
4.2 Target aim point selection algorithm

During training in target detection, MSNN takes a sub-image as input and produces two output values: target or non-target. For testing, the network scans an entire input scene and generates an output image, the detection plane. The values of the detection plane are proportional to the outputs from the target class node. ATR systems require a point, called the target aim point (TAP), at which to aim. A network performance can be measured by the location of the TAPs. The TAP selection algorithm we used consists of thresholding

Fig. 5. Training image for scenario 4

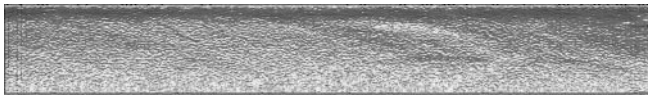


a

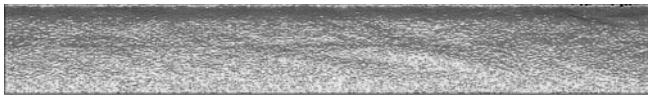


b

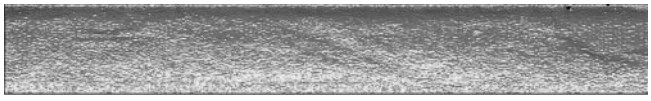
Fig. 6a,b. Training images for scenario 6



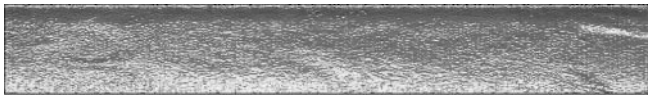
a (0, 0, 0)



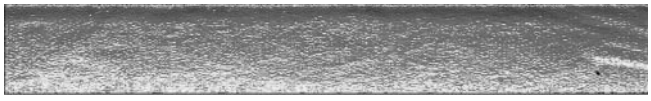
b (0, 0, 0)



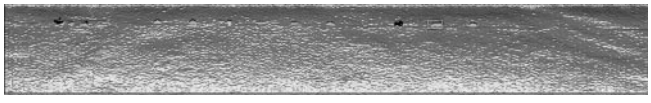
c (0, 0, 0)



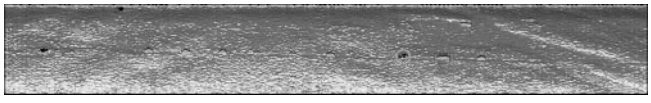
d (0, 0, 0)



e (0, 0, 0)



f (1, 1, 0)



g (1, 1, 1)

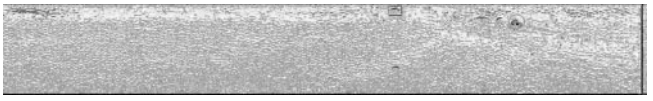
Fig. 7a–g. Testing images for scenario 4. A total of two out of two targets (*squares*) were detected with 1 false alarm (*circle*)

the output plane and applying binary opening with a 2×2 structuring element to remove speckle noise. The result is then labeled using a component-labeling algorithm. The areas and centroids of the connected components are recorded. Only components with areas in an acceptable range are considered. The centroids of these components are the TAPs.

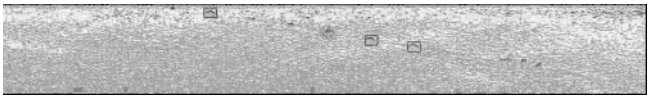
4.3 Experimental results

For the first and second sets of experiments, we trained two MSNNs using the LADAR training data set. Both MSNNs had a 20×20 input and one feature extraction layer with two

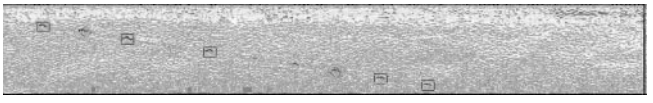
a (0, 0, 0)



b (1, 1, 1)

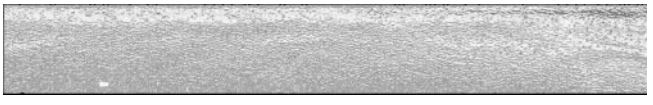


c (3, 3, 1)



d (5, 5, 2)

e (2, 2, 1)



f (0, 0, 0)

g (0, 0, 1)

Fig. 8a–g. Testing images for scenario 6. A total of 11 out of 11 targets were detected with 6 false alarms

Table 1. MSNN detection results on LADAR testing data

	No. of detections	False alarms
MSNN trained on different types of targets	75/86	133
MSNN trained only on tanks	30/30	45

feature maps. The downsampling rate was 2 (i.e., 10×10 feature maps) and both structuring elements were 3×3. The feed-forward stage of the MSNN was composed of a three-unit hidden layer and a two-unit output layer (target and non-target). This structure worked the best among many structures we experimented with.

The first MSNN was trained on all 89 different targets present in the training data set, while the second MSNN was trained only on the tanks (36 of them in the training set). We wanted to see if we need only one MSNN to detect all types of targets or we need one MSNN for each target type.

The testing results are shown in Table 1. It is worth mentioning that, out of the 75 detections produced by the MSNN trained on all types of targets, 30 were tanks and that the number of false alarms is almost half of that produced by the different detectors reported in [Moore 1998]. Many of the false alarms were due to target-like "blobs" present in

Table 2. Sims scenario training conditions

Scenario	Target type	Flight directions	No. of scenes/targets
1	Bus	2035 & 2039	2/2
2	Bus	203D & 2039	2/2
3	Bus	2033 & 203B	2/2
4	Bus	203B	1/1
5	Bus	2033	1/1
6	Tank	2035 & 2039	2/8
7	Tank	203D & 2039	2/6
8	Tank	2033	1/5
9	Tank	2039	1/3

Table 3. Sims scenario testing results

Scenario	No. of scenes	Flight direction	No. of detections	False alarms
1	7	203D	2/2	0
2	7	2035	2/2	0
3	7	2035	2/2	1
4	7	2033	2/2	1
5	7	2035	1/2	1
6	7	203D	11/11	6
7	7	2035	7/10	4
8	6	203B	3/5	0
9	7	203D	11/11	6

some of the testing scenes (Fig. 4). These "blobs" are very hard to distinguish from actual targets, even to a trained eye, since they are far away in range.

Both MSNNs detected all 30 tanks in the test set, however, the MSNN that was trained only on tanks produced far fewer false alarms than the MSNN that was trained on all types of targets. This may suggest that designing different MSNNs that detect different types of targets and fusing their results may produce more detections and fewer false alarms than using only one MSNN trained on all types of targets.

We now report on our experimental results with the Sims scenario. *One* or *two* instances of an object (tank or bus) from either one or two flight directions are used to quickly train MSNN. Then the trained net is scanned across all frames of one of the other sequences (*different* flight direction) to find the object of interest. The different scenarios we tried and their corresponding results are described in Tables 2 and 3. Figure 5 shows the *one* frame from flight direction 203B used in training for scenario 4 and Fig. 6 shows the *two* frames from flight directions 2035 and 2039 used in training for scenario 6. Figure 7 shows the detection results on the testing scenes from flight direction 2033 used in testing scenario 4 and Fig. 8 shows the detection results on the testing scenes from flight direction 203D used in testing scenario 6. In Fig. 7 and 8, the three numbers below each image indicate, respectively, the number of targets in that image, the number of targets detected, and the number of false alarms.

Depending on the number of images and the number of targets per image used in the training process, the MSNNs only took between 30 and 90 s to fully train on a Sun Ultra 1 workstation. The scanning time is a small fraction of a second on the same workstation and can be improved considerably, either by better coding or by use of FFT techniques.

5 Conclusion

In this paper, we have demonstrated the ATR capability of MSNN using a LADAR imagery data set. The first set of experimental results indicate that designing MSNN to detect a certain type of target is better than designing MSNN that detects all target types simultaneously. Furthermore, the false-alarm rate is reduced by 50% over previous results reported in [Moore 1998]. Those previous results were obtained using a combination of CFAR, DEVLIN, and LODARK filters and regular feed-forward neural networks. It is important to mention that we used exactly the same training and testing data. The MSNN performed well in the Sims scenario in which it was only trained on *one* or *two* scenes in about 30–90 s on a standard Sun Ultra 1 workstation and then tested on scenes from *different* flight directions. Our results show the MSNN robustness, its ability to recognize targets even when trained on very few samples, and its ability to reduce the number of false alarms.

Acknowledgements. This work was partially supported by the Office of Naval Research (ONR) grant N00014-96-0439. We thank Dr. Clifford Lau of ONR for his support. The data sets and many useful discussions were provided by Mr. Jesse Hodge, Mr. Dave DeKruger, and Dr. Paul Kersten of the Naval Air Warfare Center. We would like to thank Mr. A. Köksal Hocaoglu of the Computer Engineering and Computer Science Dept. at University of Missouri-Columbia for his pseudo-intensity enhancement algorithm and Mr. Jerry Moore for generating the ground-truth layouts of the LADAR scenes.

References

Bhanu B, Jones T (1992) Image Understanding Research for Automatic Target Recognition. Proc. DARPA Image Understanding Workshop, pp 255–263

Bhanu B, Politopoulos AS, Parvin BA (1983) Architecture and Algorithms for Digital Image Processing. Proc SPIE 435: 90–97

Bhanu B, Dudgeon DE, Zalino EG, Rosenfeld A, Casasent D, Reed IS (1997) Special Issue on Automatic Target Detection and Recognition. IEEE Trans Signal Process 6(1)

Casasent D, Chang WT (1986) Correlation Synthetic Discriminant Functions. Appl Opt 25: 2343–2350

Casasent D, Ravichandran G (1992) Advances Distortion-Invariant Minimum Average Correlation Enery (MACE) Filters. Appl Opt 31: 1109–1116

Dougherty ER (1992) An Introduction to Morphological Image Processing. SPIE Press , Bellingham, Wash.

Gader PD, Won Y, Khabou MA (1994) Image Algebra Networks for Pattern Classification. Proc SPIE (Conference on Image Algebra and Morphological Image Processing V) 2300: 157–168

Gader PD, Miramonti JR, Won Y, Coffield P (1995) Segmentation Free Shared-Weight Networks for Automatic Vehicle Detection. Neural Networks 8: 1457–1473

Haralick RM, Sternberg SR, Zhuang X (1987) Image Analysis Using Mathematical Morphology. IEEE Trans Pattern Anal Mach Intell 9: 532–550

Hobson G, Sims SRF, Gader PD, Keller JM (1994) MACE Prefilter Networks for Automatic Target Recognition. Proc SPIE (Automatic Object Recognition IV) 2234: 402–409

Khabou MA, Gader PD, Shi H (1999) Entropy Optimized Shared-Weight Neural Networks. Opt Eng 38: 263–273

Koch M, Moya M, Hostetler L, Folger R (1995) Cueing, Feature Discovery, and One-Class Learning for Synthetic Aperture Radar Automatic Target Recognition. Neural Networks 8: 1081–1102

Li J, Zelnio E (1996) Target Detection with Synthetic Aperture Radar. IEEE Trans Aerosp Electron Syst 32: 613–627

Mahalanobis A, Vijaya Kumar BVK, Casasent S (1987) Minimum Average Correlation Filters. Appl Opt 26: 3633–3640

Moore J. (1998) Fuzzy Logic Automatic Target Detection System for LADAR Range Images. Master Thesis. University of Missouri, Columbia, Miss.

Roth MW (1990) Survey of Neural Network Technology for Automatic Target Recognition. IEEE Trans Neural Networks 1: 28–43

Sajda P, Spence C, Hsu S, Pearson J (1995) Integrating Neural Networks with Image Pyramids to Learn Target Context. Neural Networks 8: 1143–1152

Serra J. (1982) Image Analysis and Mathematical Morphology, vol. 1, Academic Press, New York, N.Y.

Serra J. (1988) Image Analysis and Mathematical Morphology, vol. 2, Academic Press, New York, N.Y.

Sims SRF, Epperson JF, Vijaya Kumar BVK, Mahalanobis A (1994) Synthetic Discriminant Functions Using Relaxed Constraints. Proc SPIE (Automatic Object Recognition IV) 1959: 146–157

Theera-Umpon N, Khabou MA, Gader PD, Keller J, Shi H, Li H (1998) Detection and Classification of MSTAR Objects Via Morphological Shared-Weight Neural Networks. Proc SPIE (Conference on Algorithms for SAR imagery V) 3370: 530–540

Won Y. (1995) Nonlinear Correlation Filter and Morphology Neural Networks for Image Pattern and Automatic Target Recognition. Ph.D. Thesis. University of Missouri, Columbia, Miss.

Won Y, Gader PD, Coffield P (1997) Morphological Shared-Weight Networks with Applications to Automatic Target Recognition. IEEE Trans Neural Networks 8: 1195–1203