Detection of vehicles in infrared imagery using shared weight neural network feature detectors

Dick de Ridder[a], Klamer Schutte[b] and Piet Schwering[b]

[a] Pattern Recognition Group, Department of Applied Physics, Faculty of Applied Sciences,
Delft University of Technology, Lorentzweg 1, 2628 CJ Delft, The Netherlands

[b] Electro-Optical Systems, TNO Physics and Electronics Laboratory,
P.O. Box 96864, 2509 JG The Hague, The Netherlands

ABSTRACT

In this paper, we discuss the possibility of using artificial neural networks (ANNs) as feature detectors in automatic target recognition (ATR). The goal is to discern a vehicle in an infrared image. We train ANNs to recognize the most easily recognizable parts of the vehicles, the wheels. The specific ANNs we use, shared weight ANNs, are especially adept at such an image recognition task due to their specialized architecture. The feature detection stage results in an image containing in each pixel the output of the ANN, indicating its confidence in the classification. We can then use a simple sequence of image processing algorithms on this image to find peaks and, by counting the number of these peaks, vehicles. This system is tested on sensitivity to scale differences and background clutter and is shown to perform quite well.

Keywords: automatic target recognition, shared weight neural networks, radial basis functions

1. INTRODUCTION

The goal of the work described in this paper is to recognize vehicles in infrared (IR) imagery, i.e. automatic target recognition (ATR). There is a large body of literature describing attempts to use artificial neural networks (ANNs) to such a task; see for example the overviews of Roth[1] and Rogers et. al.,[2] the special issue of Neural Networks including Koch[3] and others, or Ranganath et. al..[4] However, in most of these approaches non standard learning methods or image preprocessing techniques have been used, or the ANNs used have been hand tailored to the problem.

We chose to use an off-the-shelf approach called shared weight ANNs. These ANNs are especially suited for image recognition tasks[5–8,3,9,10] due to their architecture, which allows the ANN to perform convolution-like operations (filtering and template matching) on its input. Furthermore these ANNs, although quite large in their number of connections, have only a limited number of free parameters, which can be an advantage in both training and final implementation. The ANN architectures are discussed in section 3.

There is a major difference between standard classification problems and this particular problem, in that rejection of unknown samples is of great importance. Obviously this has some effects on the ANN architecture and data structures used. This problem is also discussed in section 3.

In Gader,[10] an approach rather similar to ours is reported, also utilizing a shared weight ANN for the recognition of vehicles in IR imagery. The difference between their work and ours lies in the fact that we train the ANNs to recognize only a specific vehicle part, i.e. the wheels, using the ANNs only as feature detectors. In a later stage we can then easily determine the presence of a vehicle using some simple image processing steps by using the fact that wheels should occur in a row on the vehicles we are interested in. Assuming that false alarms from the ANNs are independent, this latter stage can catch a large number of the errors made by the ANNs since they have a very small chance of laying on a line. The postprocessing stage is presented in section 4.2.

Since the ANN feature detectors are built to receive image patches as inputs, no special preprocessing is necessary. The only problem is the scale of these patches, since training time and the number of units in the ANN increase approximately linearly with the number of pixels in the input. We chose to train the ANNs on fixed scale image patches and handle the scaling problem in the recognition stage. In fact, if we assume that the distance to the potential object is known (for example, using radar, a laser range finder, passive ranging, digital terrain models,

Further author information - DdR: e-mail: dick@ph.tn.tudelft.nl, KS: e-mail: Schutte@fel.tno.nl, PS: e-mail: Schwering@fel.tno.nl

Part of the SPIE Conference on Signal Processing, Sensor Fusion, and Target Recognition VII • Orlando, Florida • April 1998
SPIE Vol. 3374 • 0277-786X/98/$10.00

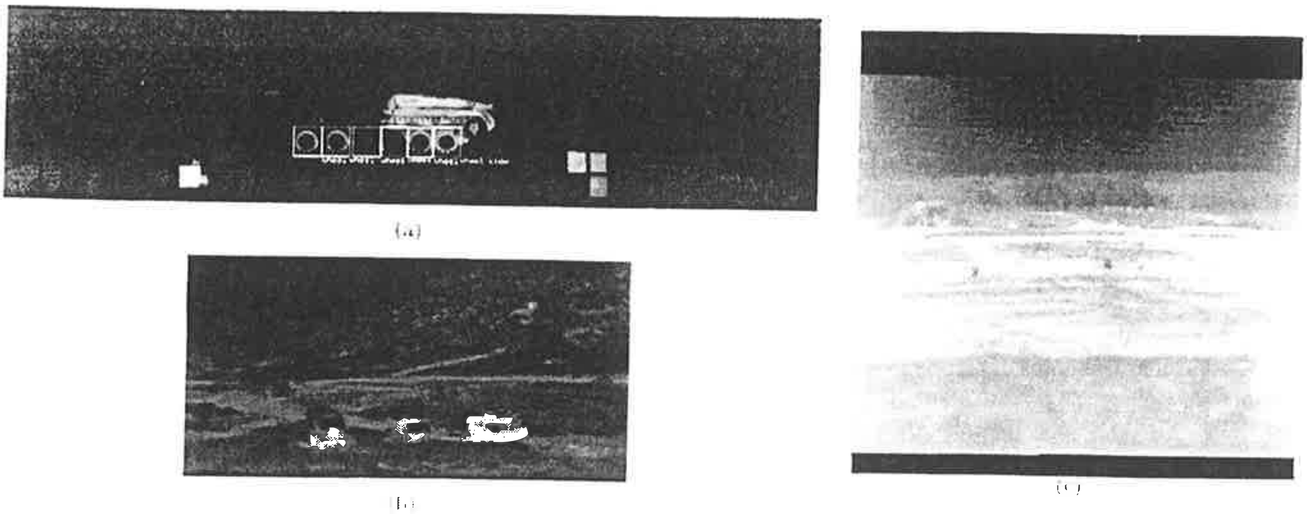


Figure 1. Example images: (a) A 3 to 5 μm IR image. Wheel sample bounding boxes are indicated. Note: all 160 images contain a number of "hot plates", which are included to calibrate the cameras. These plates were masked out when testing the system. (b) A clutter image taken with the TNO-FEL Scorpio camera. (c) A clutter image taken with the UA-92 camera.

etcetera), with a known instantaneous field of view (IFOV), we can derive the approximate scale parameter. This scale parameter is used to compute the magnification to match the input of the ANN feature detector. The procedure to search for the row of wheels is described in section 4.2.

Results obtained with the developed algorithms are presented in section 5. The techniques have been tested on scale sensitivity and sensitivity to background clutter. We will show that the system behaves adequately. Finally, we will some ideas on further research.

2. THE DATA

2.1. Content

Our data set consisted of a database of IR images. It contained 160 12-bit 1000 × 240 images, each of which depicts one vehicle under different rotations. Eighty of these images were taken in the 3 to 5 μm range, the other eighty in the 8 to 12 μm range, all using the TNO-FEL DUDA-1 camera. Figure 1 (a) shows an example of an image.

Three extra images were used to test the sensitivity of the system to background clutter. Two of these (512 × 512 pixels) were taken with the 8 to 12 μm UA-92 camera; the other (512 × 256) was taken with the TNO-FEL Scorpio 8 to 12 μm camera. Examples of these clutter images are shown figures 1 (b) and 1 (c). Although the number of clutter images is rather low, we believe that the images used are sufficient to get an idea of the clutter sensitivity.

In each image, samples of wheels were indicated by a bounding box. This was done by hand. Note that, since the images contained various vehicles with several orientations, there was quite some variation between samples and a large number of partially obscured samples.

2.2. Preprocessing

All images were preprocessed by converting to 2D 32 bit float images, subtracting the image mean and dividing the result by 2.0 × the image standard deviation. Because the gray value distributions usually were asymmetric, the conversion resulted in images containing values roughly in the range $[-1.0, 7.0]$.

Table 1. The data sets used. Note that the train and validation set sizes are given as an average, since these sets were randomly drawn from the entire data set five times. Furthermore, note that the test set consists of entire images, containing 290 wheel samples in total.

Set	Train (on average)				Validation (on average)				Test
Sample type	Wheels	Mirrored wheels	Background	Total	Wheels	Mirrored wheels	Background	Total	Wheels
Size	747	747	2153	3647	149	149	431	729	290

2.2.1. Test set extraction

The data was split into three parts: a training set, used to train the ANN, a test set used to judge its performance and a validation set used to stop training. This set plays the role of test set in the training phase, to avoid overtraining: if the error on the validation set starts to rise, the level of maximum generalization has been reached.

First, some 25% of the images (40) was chosen at random and put aside to serve as a test set. Since it would have been too problematic to split the set on the sample level, because we judge the system's performance on entire images, the fraction of test samples is not always exactly 25%.

From the remaining 120 images, samples of wheels and background were extracted. Each sample consists of an input vector and an accompanying desired output vector or label. Both must to be coded to suit the ANN. These codings are discussed below.

Of each wheel in the image, two samples were created: the original and a horizontally mirrored version to incorporate the prior knowledge that recognition of wheels should be invariant under this transformation. As a side effect, this (artificially) enlarges the training set, which is a bonus for training the ANNs.

Furthermore, from each image a number of negative (i.e., non wheel) samples were extracted: background samples, samples taken between the wheel centers and samples taken above or below the wheel centers. The latter two were added to force the ANN to give precise localized responses on the wheel centers only. The size of background samples was chosen to fall in the same range as the size of the foreground samples.

2.2.2. Coding

The input vectors of the samples were created by extracting regions indicated by the bounding boxes. Square regions were taken to preserve the aspect ratio of the samples. These regions were resized to 16 × 16 pixels using linear interpolation.

Since we scale all samples to the same size here, there will be no inherent scale invariance in the ANN. In our opinion it is extremely difficult, if not impossible, to train an ANN to recognize an entire scale range of samples. Furthermore, for large sample vector sizes (e.g., a bounding box of 50 × 50 pixels) training becomes intractable since the input layer becomes too large. We believe scaling should instead play a role in deciding at what resolution test images should be presented to the ANN.

The desired output pattern, or target, differed for the various ANNs we used (see section 3.2.2). For ANNs with one output, the target was 0.25 for background samples and 0.75 for wheel samples. For ANNs with two output units, the target was a vector of length 2 with values 0.75, 0.25 and 0.25, 0.75 for background and wheel samples, respectively.

2.2.3. Validation set extraction

Finally, each of these four sets was randomly split five times into a training set (62.5% of the original number of samples, of which 25% was already set aside to serve as test set) and a validation set (12.5%). Table 1 gives an overview of the data sets used.

3. THE NETWORKS

In this section, we will give a short overview of shared weight ANNs and the modifications made to the standard architecture in these experiments. We assume a basic knowledge of neural networks; see for example Hertz et al..[11]

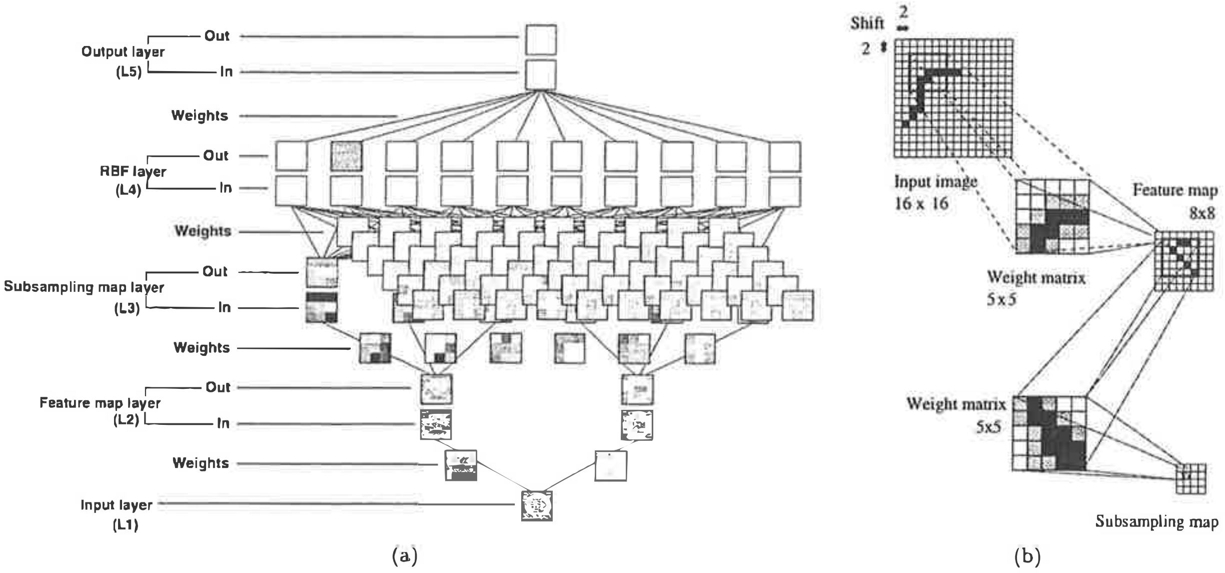


Figure 2. (a) An example shared weight ANN. This figure shows the basic architecture with an added layer of radial basis function units. (b) A feature map and sub sampling map.

3.1. Shared weight ANNs

Shared weight ANNs[6–8] were proposed in the late 1980s by Le Cun et al. for the problem of handwritten digit recognition. A number of variations on the basic architecture have been proposed. Here we discuss the ANN used in our experiments, which was originally proposed by Viennet.[5]

3.1.1. Introduction

A shared weight ANN is basically a back propagation neural network. It uses the concept of shared weight, that is, a set of units in one layer using the same incoming weight. Furthermore, it uses these shared weights to implement receptive fields – groups of units detecting features locally. The use of shared weight ensures that all these units detect the same feature, though at different positions in the input image. The detected features are - at a higher level - combined to ensure shift-invariant feature detection. Although the architecture was designed specifically for one goal, i.e., recognition of handwritten digits, some of the underlying principles are more widely applicable.

3.1.2. Architecture

The basic ANN (based on a proposal by Viennet[5]) comprises four layers, including input and output layers. Figure 2 (a) shows an extension of this basic architecture (with an added radial basis functions (RBF) layer (4), see section 3.2). Each of the layers is discussed in detail below.

Input layer (L1)

The input layer consists of a square region of units. Sample images are presented to the ANN here.

Feature maps (L2)

The second layer contains the so-called *feature maps* (see figure 2 (b)). Each unit in such a feature map has the same set of incoming weights, but is connected to a square at a unique position in the input image. This set of weights can be looked on as a convolution filter, or template; that is, if a unit in a feature map fires, this corresponds to a match with the template. The location of the match in the input image corresponds to the location of the unit in the feature map. This is what is meant by *receptive fields*: units react to the same feature at different positions in the input image. The receptive field for two neighboring units is shifted a number of pixels in the input image. Thus the image is *under sampled*. The idea behind this is that, while high resolution is important for detecting a feature, it is not necessary to know its position in the image with the same precision.

In the ANN used here, there are two feature maps of different sizes. One contains 7 × 7 units, the other 5 × 5 units. Both are connected to the input layer through 4 × 4 sets of incoming weights, albeit with different pixel shifts (two and three, respectively). Therefore, the smaller map detects features at a coarser level than the larger map.

The feature maps form one of the most interesting concepts of the Le Cun architecture. The technique of template matching and filtering is well known from classical image processing, but here the ANN trains the filters itself; no prior knowledge of the actual data (here, wheel and vehicle images) is involved.

Sub sampling layer (L3)

The third layer is a sub sampling layer (figure 2 (b)).

The principle is the same as for the feature maps: each unit in a sub sampling map is connected to a 3 × 3 square and all units in one sub sampling map share the same set of 9 weights. Here, too, the feature map is under sampled, again losing some of the information about the location of detected features.

In this ANN, two groups of three maps are connected to a single feature map each.

Other layers (L4)

Usually, a possible hidden layer and output layer are normal ANN layers, fully connected to the previous layer. In the basic architecture, there is no hidden layer; figure 2 (a) shows a variation with an RBF layer (4). The output layer consists of 2 units. The variations on this architecture constructed are discussed in the next section.

Parameters

The total number of units in the basic architecture thus is 386; the total number of parameters is 3556 and the total number of *unique* parameters is only 194, due to weight sharing. It should be evident that this reduction in the number of weights is of great use in reducing training time and stabilizing the learning process.

3.2. Radial basis function ANNs

3.2.1. Rejection

In standard pattern recognition problems, one is usually given a number of training examples $\mathbf{x}_i$ with a corresponding label y_i (the "ground truth") and one is interested in finding or learning a function $F(\mathbf{x}_u) = y_u$, which returns the right class label y_u for an unknown object $\mathbf{x}_u$ as well as possible. Usually, a *between-class rejection criterion* is added, which enables the classifier F to return a "don't know" for samples for which it is too difficult to tell to which class they belong.

In the case of recognizing (small) objects in images however, it is of equal importance to be able to use rejection of a second kind: *outlier detection.* This rejection is necessary when we need to classify a sample as either belonging to a certain class or belonging to no class at all. Clearly, when a classifier must discern vehicles or vehicle parts against a noisy background, this kind of rejection becomes important.

ANNs (and other pattern recognition techniques) by default do not support rejection of the second kind. Instead, usually some kind of distance measure has to be used to reject samples. For example, rejection of the first kind can be implemented by looking at the outputs of an ANN. If the difference between the highest output and the second highest output is small, this indicates that the ANN is "not sure" about the classification. Indeed, since ANN outputs can be regarded as a posteriori probabilities (under certain prerequisites), this indicates that the ANN is uncertain.

Rejection of the second kind is much harder to implement, since the output of an ANN unit is usually transformed by some sigmoid function (which makes the ANN a nonlinear classifier).

Many advanced approaches to incorporating rejection of the second kind have been proposed, among which the use of special learning rules (see e.g. Moya and Hush[12]) which forces ANNs to learn closed decision boundaries and advanced forms of vector quantization such as adaptive resonance theory (ART, see e.g. Moya et al[13] and Carpenter et al.[14]).

In this paper, we test two rather simple solutions. The first is to add an output unit to the network which should represent all non wheel (background) samples. The second, more sensible approach is to use a form of vector quantization. We expect the latter method to perform better. The standard method of incorporating vector quantization is to use a radial basis function (RBF) layer, containing units with a Gaussian transfer function, localizing the ANN's response, of which width and position are learned.[11] Its advantage is that each representative vector has its own neighborhood size, which is learned. The main disadvantage is that the number of representative vectors must be chosen in advance.

Table 2. ANN architectures used in the experiments.

ANN	RBF layer	Hidden layer	Number of outputs	Number of units	Number of links	Number of weights
NET_A1	No	No	2	386	3556	194
NET_A2	No	Yes	2	406	5580	1206
NET_B1	Yes	No	1	395	4440	636
NET_B2	Yes	Yes	1	415	5920	1376

3.2.2. Networks used in the experiments

To see if the use of an RBF layer would lead to better results, and to get an indication of the ANN complexity needed for this task, we tested four variations of the architecture described in section 3.1. These variations, also listed in table 2, are:

- NET_A1: the normal network
- NET_A2: a network with a 20 unit normal hidden layer before the output layer
- NET_B1: a network with a 10 unit RBF layer before the output layer
- NET_B2: a network with both a 20 unit hidden layer and a 10 unit RBF layer before the output layer (in that order).

4. EXPERIMENTS

4.1. ANN feature detection

We trained five instances of each of the ANNs listen in table 2 on differently randomized and split data sets. In all experiments, the algorithm used was the conjugate gradient descent method.[15,16] This algorithm has no parameters. A validation set was used to prevent overtraining.

The ANNs were trained on isolated samples. However, in reality one wants to "scan" an image for the presence of certain classes. Hence it would be appropriate to use the ANNs as a nonlinear filtering technique. The input of the ANN feature detector will be the content of a window sliding over the input image; the result of the filtering will be an image. In this image, each pixel contains the probability of the presence of the class represented by that unit at that location for one ANN evaluation. Figures 4 (b) and 4 (e) show examples of such filtered images.

Testing the ANNs proved to be highly time-consuming, due to the fact that we tested each ANN on 40 test images at a large range of scales. To restrict the number of testing runs, we only used ANNs whose performance was judged viable by eye for further processing. These ANNs are indicated by a "•" in table 5.1. Note that none of the ANNs NET_A1 were tested; these were all deemed to perform too poorly to be of any further interest.

4.2. Wheel row detection

The output of the ANN feature detector is, for each pixel, a number representing the probability of the presence of a wheel at that location. Within a 5×5 environment we search for a local maximum. All local maxima are thresholded using a certain threshold t_{lm}. This threshold can be found according to some criterion; for example, we can set the threshold in such a way that no false alarms are generated.

For wheel row detection, the typical pattern of wheel detections must be found, i.e. the normal configuration of wheels on the vehicle. For the vehicles considered in this research, the typical pattern is a set of wheels located on a line. This leads to a search space in the shape of an elongated rectangle. For the images within the data set considered, the observer was at the same height as the object itself. This means that the rectangle is always oriented horizontally, independent of the orientation of the vehicle. The search is executed as a score boarding process: each detected wheel leads to a score within its rectangular 81×9 area. A wheel row is detected if for a certain pixel a score greater than or equal to 4 is obtained.

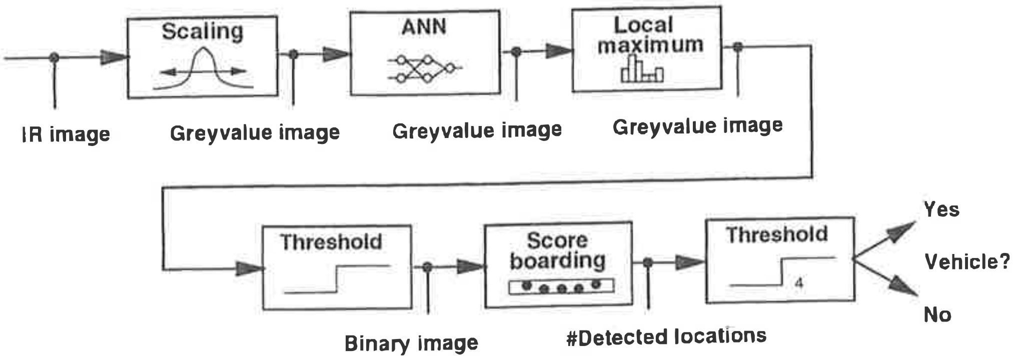


Figure 3. The detection algorithm.

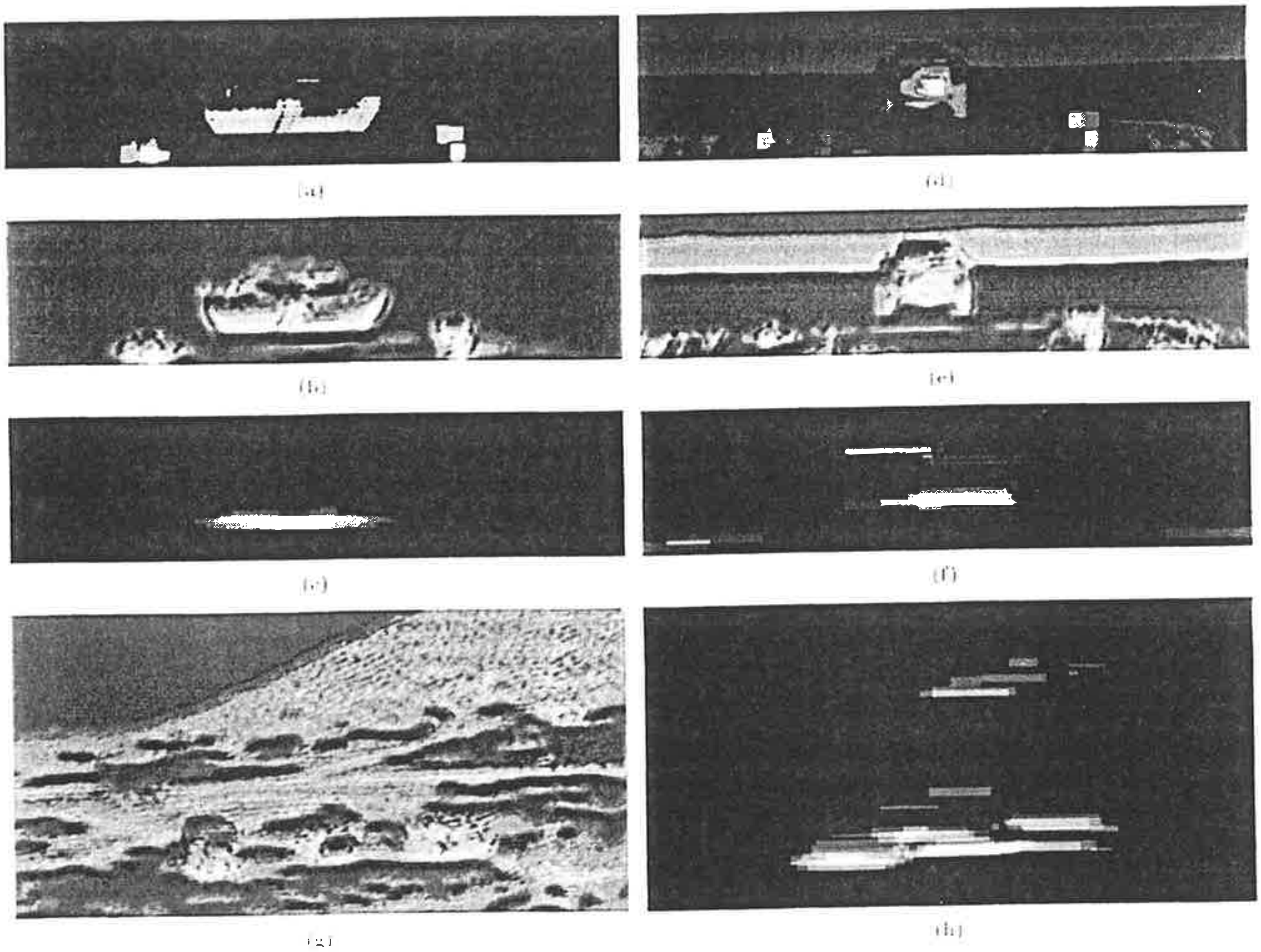


Figure 4. The recognition process for three images: an image containing wheels (a)-(c), an image containing no wheels (d)-(f) and the clutter image of figure 1 (b) in (g)-(h). The top image is the original: the middle is the version filtered by an ANN (in this case, `NET_A2_2`) and the bottom image shows the result of the score boarding process. *Note*: for the evaluation of the sensitivity to background clutter, the vehicles in the clutter image have been masked out.

Table 3. Training results in % error. ANNs marked with "•" were used in the testing phase; the other ANNs were judged to perform too poorly to warrant further processing. Note that the 0.0% error for the first two NET_B2's does not indicate that these networks performed well; instead, these networks converged to local minima.

ANN	Error on	Trial					Minimum on validation	Average	Standard deviation
		1	2	3	4	5			
NET_A1	Train	19.0	17.5	23.9	22.2	18.1	17.5	20.1	2.8
	Validation	20.0	18.5	26.3	22.6	19.9	18.5	21.5	3.1
NET_A2	Train	• 16.6	• 20.6	• 15.8	20.3	• 24.5	15.8	19.6	3.5
	Validation	21.3	20.4	16.3	22.8	26.1	16.3	17.2	8.7
NET_B1	Train	• 6.7	12.8	• 12.5	18.3	• 8.3	6.7	11.7	4.5
	Validation	8.0	13.6	12.6	18.2	7.4	8.0	12.0	4.4
NET_B2	Train	0.0	0.0	• 15.8	• 16.3	• 13.1	0.0	9.0	8.3
	Validation	0.0	0.0	14.0	16.9	14.9	0.0	9.2	8.4

A different approach should be taken if the observer is not located in the plane orthogonal to the rotation axis of the vehicle. In that case, the row of wheels has a different orientation in the image, dependent on its rotation angle. This means that during the detection phase, the score boarding rectangle has to be applied for all rotation angles. However, a finite set of representative rotations of this score boarding rectangle will do. As a side effect, the false alarm rate will increase (as possibly more patterns are erroneously identified as objects). Thus, any knowledge about the possible rotation angles should be incorporated in the algorithm to reduce the false alarm rate.

For many ground to ground scenarios it can be argued that the current solution, assuming that the wheel line is (nearly) horizontal, is adequate. As such we feel that this choice does not limit the usability of the system.

Figure 3 shows a schematic presentation of the recognition algorithm. Figure 4 shows some examples of processed images.

5. RESULTS

5.1. ANN training

In this section the results gathered during the training phase are given. The error percentages were calculated as follows:

- For non-RBF ANNs (having background as an extra class), the index of the unit with the maximum output should be the index of the output unit associated with the true class. For example, when testing NET_A2 with a background sample, the second output should be the largest.
- For RBF-ANNs, there is a need to distinguish between two cases:
  - *The sample is a true sample.* In this case the demand is the same as before, with the addition that the maximum output should be larger than 0.5.
  - *The sample is a background sample.* In this case, all outputs should be smaller than 0.5.

It is clear that during training the RBF ANNs we use 0.5 as a rejection threshold. This need not be the optimal choice. However, the optimal threshold can only be found after training. The error percentage can at least be used for comparison between training sessions, ANN architectures and data sets. Testing on a test set will be discussed in the next section.

The results are given in table 5.1. Note that the errors given in this table are *not* test errors and that anomalies may occur due to the rather arbitrary way of measurement. For example, the 0.0 error of NET_B2 does not indicate that the ANN will perform well on test images; instead, these ANNs converged to some local minima. Furthermore, it is evident that results are quite erratic. There are two possible reasons:

Table 4. The maximum threshold for each network for which no false alarm was found in the clutter images and the corresponding percentage of detected vehicles.

ANN	Threshold	Vehicles detected (percentage of total)	ANN	Threshold	Vehicles detected (percentage of total)
NET_A2_1	0.86	23	NET_B1_3	0.75	35
NET_A2_2	0.67	83	NET_B1_5	0.71	59
NET_A2_3	0.72	60	NET_B2_3	0.76	9
NET_A2_5	0.76	14	NET_B2_4	0.74	40
NET_B1_1	0.79	35	NET_B2_5	0.71	49

- the different splitting of the data into training and validation sets for the five instances, which would indicate that there is not enough training data for the ANN to generalize on;
- the difficulty of the problem easily leads the ANNs into local minima, depending on the weight initialization.

5.2. Detection performance and scale sensitivity

Figure 5 shows the detection performance, using the wheel row detection algorithm, of NET_A2_2. This was the best performing ANN, but other ANNs give similar results. Figure 5 (a) indicates for threshold t_{lm} = 0.67, for a number of scales, the average number of detected patterns in the 81 × 9-pixel search area; figure 5 (b) shows the performance around the optimal scale, that is the scale for which the wheels in the image have on average the same size as was used to train NET_A2_2, 16×16 pixels. It clearly shows a peak around the optimal scale. However, taking into account the imprecision in the range finding procedure and the errors made when the bounding boxes were indicated in the data set, it is unlikely that we will be able to calculate this optimal scale with enough precision.

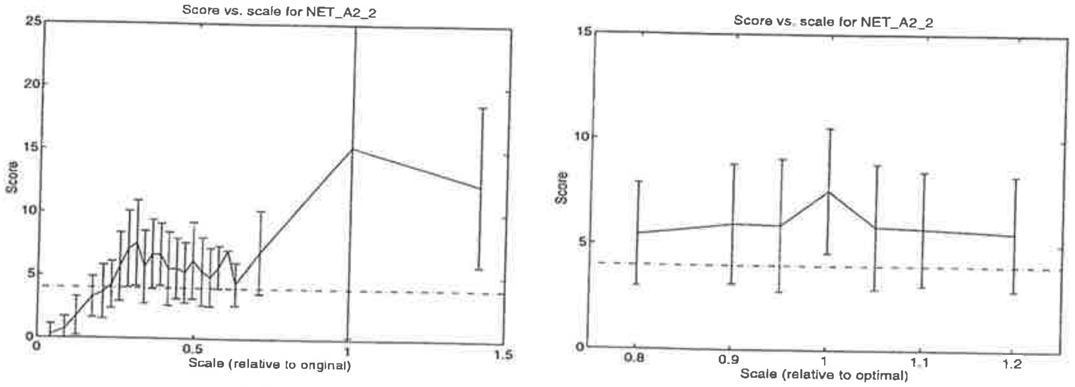
The 0.67 threshold was found to be the optimal threshold for NET_A2: it was the highest threshold that gave no false alarms on the three clutter images. Note that this threshold need not be optimal for all ANNs (see table 4 for an overview).

If four or more patterns are detected (the dotted line in figures 5 (a) and 5 (b)), the conclusion is that a vehicle is present in the images. The graphs have been somewhat smoothed (using bins of 0.025) to present the results clearly. The figures show that, to a certain extent, the number of detected patterns per unit area is constant and independent of the scale. Only for very small scales this does not hold: the number of detected patterns drops below 4 and consequently no vehicle is detected. From this we can deduce that tests on clutter images will not have to be performed on an entire range of scales, since a single scale gives a good indication of the rest of the range. Another conclusion is that the ANN feature detector does not only detect wheels; otherwise, we would expect at most 19 (and probably less) detected patterns, since that is the maximum number of wheels present in a test image.

5.3. Sensitivity to background clutter

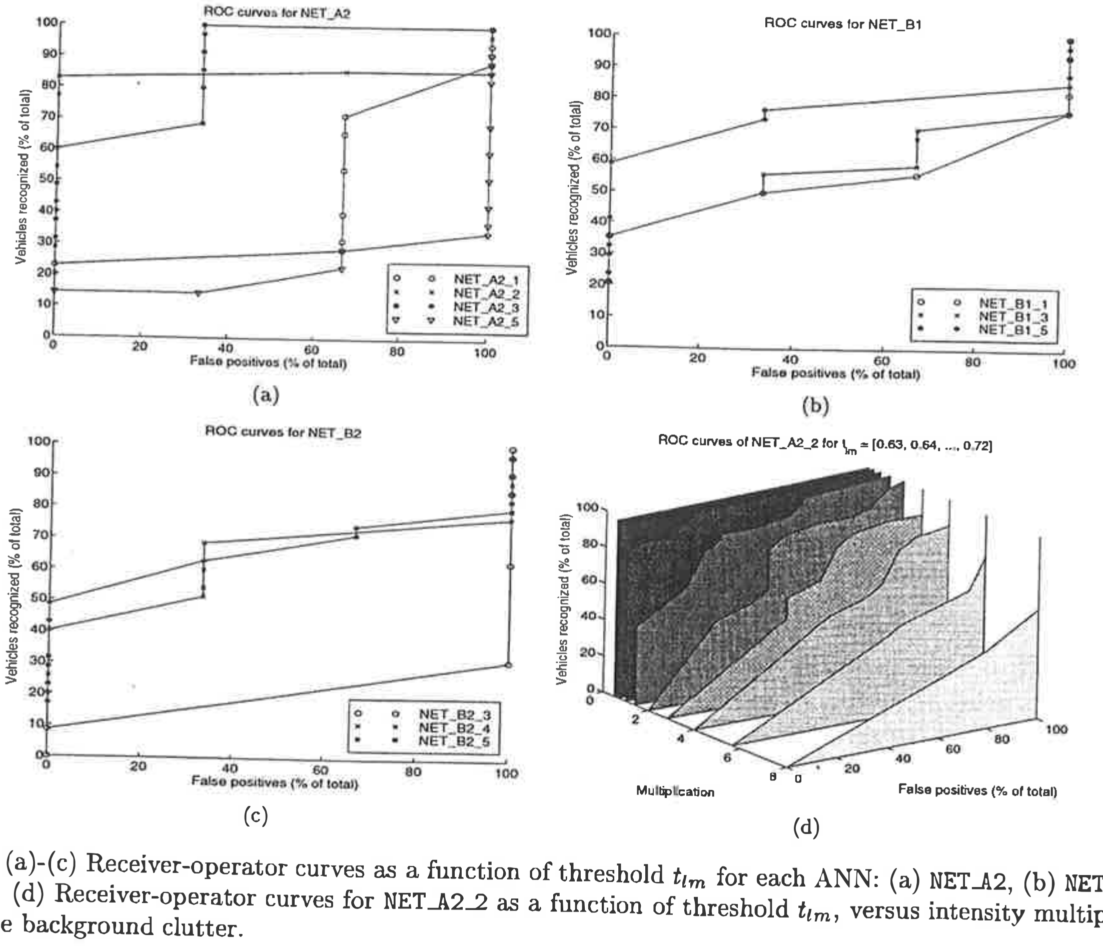
Figures 5.3 (a)-(c) give receiver-operator curves for each ANN. Note that, since there were only three clutter images, only three levels of false alarm are possible. We believe however that the trend is clear.

A problem with the evaluation of clutter images was the normalization step. Recall that each image was normalized by subtracting its mean and dividing by 2.0 × its standard deviation. Since images containing vehicles have a small number of pixels with relatively high values, this results in a very nonsymmetric gray value distribution. In clutter images however, the result is quite symmetric and the resulting normalized image has a lower maximum gray value than an image containing a vehicle. To investigate in how far the ANNs simply respond to image intensity variation, we tested one ANN (NET_A2_2) on the clutter images for various levels of intensity multiplication. Figure 5.3 (d) shows that for small multiplications, performance remains well. For large multiplications however (larger than 2), performance drops to a point where the system begins to produce a large number of false alarms. Although this is undesirable, this may not be a problem as long as the condition is met that the vehicle to be recognized has a relatively high variance with respect to its surroundings.



(a) (b)

Figure 5. Detection performance for NET_A2_2 using $t_{lm} = 0.67$: (a) Number of detected patterns versus scale; here a scale of 1.0 means the original, not scaled image. (b) Number of detected patterns around the correct scale; here a scale of 1.0 means the image scaled with the average factor used when the wheel samples were extracted (i.e. the wheels have the same size as those used to train the ANN). The graphs indicate the average number of patterns detected over all 40 test images; the bars indicate the standard deviation.



(a) (b) (c) (d)

Figure 6. (a)-(c) Receiver-operator curves as a function of threshold t_{lm} for each ANN: (a) NET_A2, (b) NET_B1 and (c) NET_B2. (d) Receiver-operator curves for NET_A2_2 as a function of threshold t_{lm}, versus intensity multiplication factor of the background clutter.

6. CONCLUSIONS AND RECOMMENDATIONS

6.1. Conclusions

We created a system for the detection of vehicles in IR images, using a more or less standard ANN technique and some dedicated yet simple algorithms for processing the ANN output. We believe that the use of an ANN as a feature detector, followed by some traditional system can deal with more a priori knowledge, e.g., the simple rule that there must be four or more wheels present on a single line to detect a vehicle, than would be possible using an all-neural system. The system performs reasonably well.

Our hypothesis that some form of rejection would be necessary was disproved by the experiments, since the networks with an RBF layer did not perform significantly better than the other systems. Only in the case without an added hidden layer did the addition of an RBF layer contribute something. Also, the addition of an RBF layer to an ANN having an extra hidden layer does not seem to contribute anything. This can be the result of these ANNs being complex enough to solve the problem as well as possible, or of the data set size being insufficient to train these more complex ANNs. The latter is more likely, since the more parameters a pattern recognition system has, the more training samples are necessary to estimate the right set of parameters.

A general conclusion about the ANN feature detection part is that various trained instances of the ANNs show great variation in performance. This may be due to an extreme sensitivity to the exact initialization of the ANN, which indicates that a larger architecture or a more conservative learning rule should be used, or a sensitivity to the division of the data set, which indicates that more training data should be used. At any rate, this makes it difficult to judge the ANNs performance well; any result may be an artifact.

We also tested the ANNs toward sensitivity to scale and background clutter. The ANNs seem to be sensitive to scaling differences in very small areas around the right scale, yet they give responses over a very large range of scales. Although the ANNs are trained to respond only to wheels, this indicates that they also react on other parts of the vehicle. While a more selective detector could be useful as performance will increase in even lower signal-to-clutter ratios, we currently use the response of the ANNs to parts of vehicles other than wheels to recognize the vehicles with orientations such that wheels are not visible, or are severely distorted.

Finally, the clutter sensitivity seems to be reasonable on the three images we used. With the best network, NET_A2_2, it is possible to reach a 0.0% false alarm rate and still recognize 83% of the vehicles present in our test set.

6.2. Recommendations

The following topics are worth investigating in future research:

- A larger training set could be used to characterize the influence of the training set size.
- The number of RBF kernels in RBF ANNs could be varied to get a better idea of their performance, compared to non-RBF ANNs.
- More complex methods of outlier detection, i.e., rejection of the second kind, could be implemented.
- The influence of ANN complexity (such as the number and size of the feature maps and sub sampling maps) in general can be investigated in more depth.
- Within the current set of experiments we used the fact that the observer and plane of rotations of the vehicle coincide. Under general conditions, including air to ground scenarios, this is not true, and the experiments should be expanded to cover all possible vehicle orientations.
- Currently we consider only a single ANN, which is trained to recognize wheels of a broad range of vehicles. Using in parallel a bank of ANNs to recognize more specific wheel types could improve performance.
- Within this research the only features used are wheels. Adding (ANN) detectors for other features, such as, for example, tank barrels or wind shields of trucks, in parallel with the wheel feature detectors could help to reduce the false alarm rate.

7. ACKNOWLEDGMENTS

We would like to thank Dr. Ir. R.P.W. Duin of the Pattern Recognition Group for valuable advice and co-operation. Also, thanks go to Ir. M. van Ginkel and Ir. G. van Kempen for helping with some software problems. Parts of this paper are based on preliminary work executed for the START consortium in the framework of EUCLID RTP-8.2 "Intelligent Sensors". Our collaboration and discussions with Signaal, and in particular with the INETI-DOP, in that work is acknowledged. This work was partly supported by the Foundation for Computer Science in the Netherlands (SION) and the Dutch Organization for Scientific Research (NWO).

REFERENCES

1. M. Roth, "Survey of neural network technology for automatic target recognition," *IEEE Transactions on Neural Networks* **1**(1), pp. 28–43, 1990.
2. S.K. Rogers, J.M. Colombi, C.E. Martin, J.C. Gainey, K.H. Fielding, T.J. Burns, D.W. Ruck, M. Kabrisky, and M. Oxley, "Neural networks for automatic target recognition," *Neural Networks* **8**(7/8), pp. 1153–1184, 1995.
3. M.W. Koch, M.M. Moya, L.D. Hostetler, and J.R. Fogler, "Cueing, feature discovery, and one-class learning for synthetic aperture radar automatic target recognition," *Neural Networks* **8**(7/8), pp. 1081–1102, 1995.
4. Ranganath, H.S., Kerstetter, D.E., and Sims, R.F., "Self partitioning neural networks for target recognition," *Neural Networks* **8**(9), pp. 1475–1486, 1995.
5. E. Viennet, *Architectures Connexionistes Multi-Modulaires, Application à l'Analyse de Scène.* PhD thesis, Université de Paris-Sud, Centre d'Orsay, 1993.
6. Y. Le Cun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard, and L.D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural Computation* **1**, pp. 541 – 551, 1989.
7. Y. Le Cun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard, and L.D. Jackel, "Handwritten digit recognition with a back-propagation network," in *Advances in Neural Information Processing Systems 2*, Touretzky, D., ed., Morgan Kaufmann Publishers, San Mateo, CA., 1990.
8. Y. Le Cun, L.J. Jackel, B. Boser, J.S. Denker, H.P. Graf, I. Guyon, D. Henderson, R.E. Howard, and W. Hubbard, "Handwritten digit recognition: applications of neural network chips and automatic learning," *IEEE Communication* **27**(11), pp. 41–46, 1989.
9. D. de Ridder, "Shared weights neural networks in image analysis," Master's thesis, Delft University of Technology, March 1996.
10. P.D. Gader, J.R. Miramonti, Y. Won, and P. Coffield, "Segmentation free shared weight networks for automatic vehicle detection," *Neural Networks* **8**(9), pp. 1457–1473, 1995.
11. J. Hertz, A. Krogh, and R.G. Palmer, *Introduction to the theory of neural computation*, Addison-Wesley, Reading, MA, 1991.
12. M.R. Moya and D.R. Hush, "Network constraints and multi-objective optimization for one-class classification," *Neural Networks* **9**(3), pp. 463–474, 1996.
13. M.R. Moya, M.W. Koch, and L.D. Hostetler, "One-class classifier networks for target recognition applications," in *Proc. World Congress on Neural Networks*, pp. 797–801, International Neural Network Society, INNS, (Portland, OR), 1993.
14. G.A. Carpenter, S. Grossberg, and D.B. Rosen, "ART 2-A: an adaptive resonance algorithm for rapid category learning and recognition," *Neural Networks* **4**(4), pp. 493–504, 1991.
15. J.R. Shewchuk, "An introduction to the conjugate gradient method without the agonizing pain," Tech. Rep. CMU-CS-94-125, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, March 1994.
16. A. Hoekstra, M.A. Kraaijveld, D. de Ridder, and W.F. Schmidt, *The Complete SPRLIB & ANNLIB*, Pattern Recognition Group, Faculty of Applied Physics, Delft University of Technology, 1996.