Automatic Target Recognition using Neural Networks

Dr Trevor Clarkson
Department of Electronic and Electrical Engineering
King's College London
Strand, London WC2R 2LS, UK
tgc@kcl.ac.uk

ABSTRACT

Two applications of automatic target recognition (ATR) using artificial neural networks are presented. These are, target position detection and target classification. The neural networks are based on the probabilistic RAM (pRAM) neuron which is briefly described. The pRAM has been built using VLSI techniques and includes learning on-chip which allows the pRAM to be used as an adaptive embedded controller in robust systems.

1. Target Position Detection

Given an image or scene, S, and a target image, P, the neural system is to find the coordinates of the target image, P, in the scene, S. Additionally, given any sub-scene S' containing the target, P, the system is expected to find the image P and to return its coordinates with respect to S'. It is assumed that a reference scene S_0 and a target image, P_0, are known *a priori*. It is also assumed that the range, azimuth and elevation of the observation point of scene, S, from the target, P, are known to a reasonable accuracy. This information will be used to transform any sub-scene S' to the same scale as S_0, so as to produce a scene S for analysis of where the target P is located in S.

The target image is typically derived from a photograph and the image to be matched may come from a video camera or another photograph.

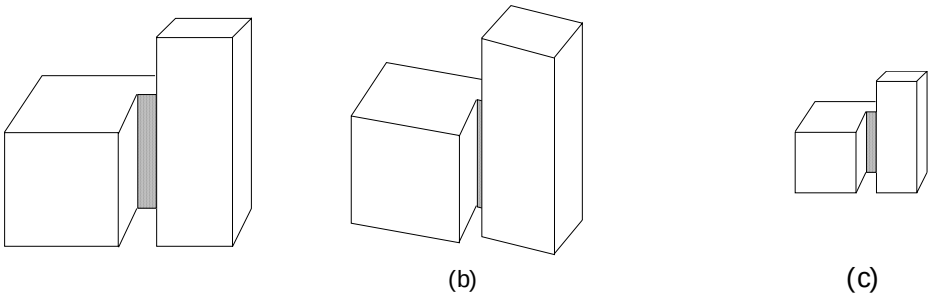

Figure 1. (a) the target image, (b) change of viewpoint, (c) change of scale

In the simple images in Fig. 1, if (a) is the target image, P_0, then (b) and (c) are the same target seen from a different angle or range. It can be seen that (b) and (c) can be made to approximate (a) by the application of suitable geometric transforms. It is not possible to obtain a perfect reconstruction of (a) from (b) owing to the three-dimensional nature of the image since these images are available to the system in two-dimensional form.

However, in real images, the target will not be readily segmented from its background. Therefore the scene in which the target is placed is significant. It is essential that additional information concerning the viewpoint of the reference scene and the observed scene is known, otherwise the transformation of the observed scene to match the reference scene cannot be performed accurately. The accuracy of this transformation is limited, in any case, and the maximum difference in the angle of view between the reference and observed images is around 30°. Standard correlation techniques do not give good performance for these complex images in arbitrary scenes, which is why a neural network is used to handle the non-linear characteristics of this problem.

There are additional problems inherent in using these techniques for outdoor scenes, which are the time of day and the time of year. Only one reference scene may exist and this will be for a certain time of day. If the scene is later observed at a different time of day, then the effects of shadows or the lack of shadows will be significant. Shadows can distort the apparent outline of objects, examples of which might be buildings or vehicles. Other problems will be caused by night/day or summer/winter differences in the images. Substantial changes, such as a change from full foliage to absence of foliage or a thick covering of snow cannot reasonably be accommodated.

In the example described later, the reference scene was derived from a high-resolution photograph and the observed images were received from an infra-red sensor. Here, there is a cross-spectral problem where parts of the image which are optically dark may appear to be light in the infra-red image owing to their high temperature. Therefore, this recognition system must be insensitive to colour.

Because of the above artefacts in the observed image, preprocessing is essential in order to remove, or at least reduce, these unwanted features. Most of the problems are caused by scene illumination; however, it is assumed that the structure of the target image will not change. This suggests that classification based on some form of feature detection will give the best results, rather than template matching techniques alone. The exact features to be extracted are dependent upon the structure of the target.

8.1 An example of position detection

In the example described below, a photograph of a building was used as the target image. The observed image was part of a set of infra-red images at different ranges.

As stated above, the relevant features for position detection are dependent upon the structure of the target. For a building, these features might be the corners or edges of the building with architectural features such as windows providing additional information. The simplest form of feature extraction is to use edge-detection. This method works well with high-resolution and high-contrast images. When low-resolution and low-contrast infra-red images are used, an edge may not be completely represented.

Therefore, a combination of conventional image processing and non-linear principal component analysis was investigated in the solution to this problem. The extracted features were passed to a pyramidal neural structure and noise was injected during training to give greater tolerance to variations in the observed image.

8.2 Preprocessing

In this example, the parameters of the reference scene viewpoint are known *a priori*. It is assumed that navigational information is available to perform second-order geometric transformations [1] on the observed image to give an approximate match to the reference scene in terms of target size and angle-of-view.

Two methods of further processing were then used and compared.

8.3 Principal Components Analysis

Principal Components Analysis (PCA) [2] of an image yields an ordered set of masks which represent the most common features in that image and their order gives their frequency of occurrence. Experiments were conducted to see how many components were required to reconstruct the original image to a given accuracy, which was normally taken to be better than 95%. In this example, six PCA masks of size 8 by 8 pixels were used.

The image is multiplied by the six PCA masks which results in six matrices. These matrices are input to the neural network. Since each set of 64 pixels yields one vector, and with six components used, a useful reduction in the dimension of the input of 64/6 was achieved.

To train the neural network, the PCA masks were extracted from the reference scene. A target point on the building was then marked manually. The image was then scanned in 8 x 8 pixel segments at 2 pixel increments. At each step, the image segment was convolved with the PCA masks and the 6-element vector applied to a neural network. The network was trained to give an output of "0" for all areas outside the marked segment and to give an output of "1" at the marked point only. The network was assessed on a geometrically-corrected infra-red image containing the same object. If successful, the network should give a maximum response when the 8 x 8 pixel area containing the previously marked point in the photograph is seen in the infra-red image.

The results of using PCA were disappointing. Although there was a peak in the response at the desired point, there were a number of other peaks in the response, some of which were larger in amplitude than the desired response. This is mainly due to the use of PCA masks derived from the photograph being used for the infra-red image. These masks are clearly not sufficient to discriminate the spatial features in the infra-red image.

8.4 Edge detection

In place of PCA, edge-detection using eight preferred orientations was used. The absolute value of the edge-detected images was used and a single image was produced by summing the eight outputs. It is clear that the discrimination performance will be improved if each edge-detected feature is separately processed, but the advantages expected will be small. It is noted that this method of edge-detection is a special case of PCA - where the components (or masks) are predefined.

The reference photograph was processed to produce an edge-detected image. Again one point on the building was marked manually. An 16 x 16 pixel window centred on this point was used to train the neural network to give an output of "1" and the complement of this area was applied to the network and trained to give an output of "0". Training noise [3] was used to improve generalisation of the network.

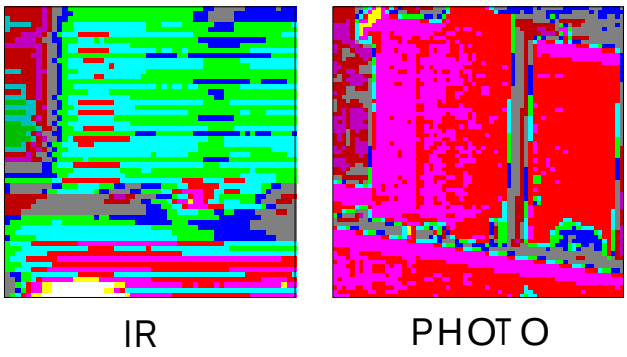

Figure 2. A section of the edge-detected infra-red and photographic images (64x64 pixels)

When the network of Fig. 3, trained on the edge-detected photographic data, was used to search for the marked point in the geometrically-corrected infra-red image, the maximum response was found at the target point. The infra-red image was searched by scanning across the entire image (256 x 256 pixels) using the 16 x 16 mask moving in 2 pixel increments.

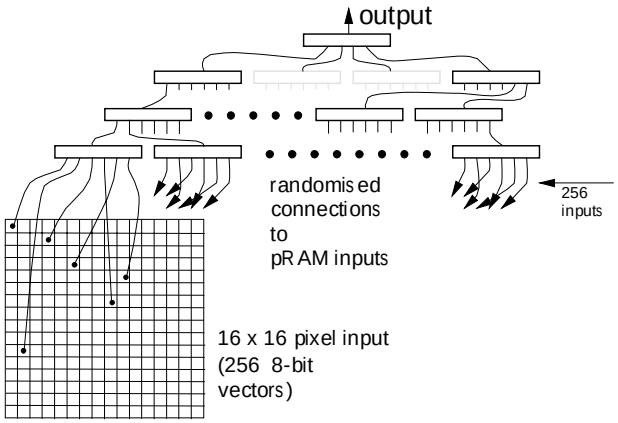

Figure 3. The pRAM neural network used in the position detection system.

Since the pRAM neuron produces an output in the form of a spike-train, and receives real-valued inputs in the same form, each input was presented for a number of iterations (typically 1000) and the output response was accumulated. The results are shown as the firing rate in Fig. 4, where the maximum response is seen at an offset of zero pixels from the marked position.

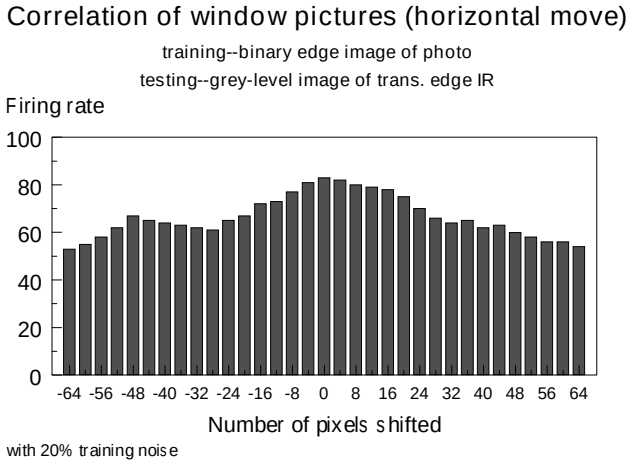

Figure 4. The search results for the target position in an infra-red image.

8.5 Discussion of results

The results in Fig. 4 represent a single horizontal scan across the infra-red image, passing through the target point. A similar scan for the vertical direction also shows a unique maximum response. In the final system, the response is a 2-D map of the response of the net as it scans the received image, in which a unique (maximum) response is required. However, a maximum response does not indicate any certainty of having found the target. If the peak of the response is not sharp, the confidence of the result is low.

A circle of error probability (CEP) estimate is required in order to assess the accuracy of the result. The CEP estimate can be made using the variance of the spike trains with the formula

$$\sigma^2 = \frac{<\alpha>(1-<\alpha>)}{R}$$

where $<\alpha>$ is the mean response at the maximum output of the net and R is the spike train length. We take the CEP as defined by the intersection of the horizontal line at 2σ below the maximum with the response curve. Thus a broad response gives a large CEP and a sharp response, a small CEP. We find that, with $R=10^3$, $\alpha \sim 0.8$, so $\sigma \sim 1\%$, that for the data of Fig 4, that the CEP $\approx \pm 6$ pixels in the horizontal direction.

9. Target Classification

A more conventional use of ATR is in target classification. Here an imaging system is used to detect the presence of features in a scene which may constitute a target. Small 'patches' are extracted from an image of size 16x16 pixels each containing a possible target. A probabilistic Random Access Memory (pRAM) neural network is used for the classification of objects in a video sequence of FLIR (Forward Looking Infra Red) images into two classes, target and clutter [4]. For simplicity, the objects are taken to be vehicles in the following description.

The image sequences used for training and testing were gathered from real scenes. These sequences of frames were first passed through a hot-spot detection system which identified points in the image that have a high probability of corresponding to a target. 16 x 16 pixel patches were centred on each hot-spot. These hot-spots have a high probability of corresponding to a target as high contrast areas in the thermal image indicate a possible vehicle engine. This stage reduces the amount of information that has to be processed subsequently. This is also the most critical part of the algorithm since an object missed will never be selected in any subsequent ATR operation.

The second stage is the feature extractor. The targets are noisy and only a few pixels in size. Hence the use of a shape-based feature extraction technique [5, 6] was not possible nor was it possible to extract any handcrafted features [7].

Then feature extraction was done on the image patches surrounding these hotspots using Principal Component Analysis (PCA). These features serve as input to a reinforcement-trained pRAM net which produces values of (1 0) for targets and (0 1) for clutter.

Each patch contains a grey-scale sub-image where the target size may vary from a few pixels across to the full width of the patch. A neural network is trained on a large number of patches taken from reference video sequences where the targets and clutter have been previously classified visually.

9.1 Principal Component Analysis

Based on the data above, Petersen [4] showed that for 90% reconstruction accuracy 6 PCA features are needed, for 95%, 15 features and for 99%, 47 features are required. An inspection of the PCA masks shows that there is little structure above the 20th eigenvector. This analysis confirmed his intuitive impression that sixteen or so features are necessary in order to maintain classification accuracy.

9.2 The pRAM classifier

In this application, a pRAM neural network was used for the classification of objects into two classes. pRAM neural nets have been successful in discriminating digits with a moderate amount of noise [3]. They also have good discrimination and generalisation properties [8].

A pyramidal network structure (Fig. 5) is adopted for the classifier as it provides a good compromise between generalisation and storage capacity. The input to the network is the

PCA data for one patch and the output is (1 0) for targets and (0 1) for clutter. The pyramidal structure consists of two layers of pRAM neurons preceded, during training, by a layer of 16 1-pRAMs (not shown) for noise-injection. The weights of the noise-injection layer are changed according to the percentage of noise applied during training. Various other architectures were also experimented with during simulations, for example one output layer pRAM (binary coded output) or 32 pRAMs in the input layer.

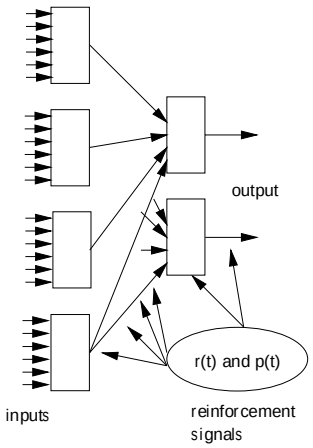

Figure 5. The pRAM neural network

9.3 Results

After training, the following results were obtained [4] where S1 and S2 were two sequences of images taken on separate occasions. Training was performed using every 10th frame of the sequences in S1 or S2. S1' and S2' are subsets of sequences S1 and S2 which do not contain any of the training set.

Training Set	Test Set	P_r Performance	P_{fa} False Alarm	P_d Detection
S2	S2	98%	2.7%	100%
S2	S1'	71%	33.6%	87%
S1	S1'	93%	8.5%	99%
S1	S2'	93%	0	84%

Table 1. Classification performance of the pRAM ATR system.

- P_r is the total number of objects correctly classified, divided by the total number of objects presented to the net,
- P_d is the number of targets correctly identified as targets divided by the total number of targets,
- P_{fa} is the number of clutter-objects incorrectly identified as targets divided by the number of clutter objects.

So the probability of detection is high, in the range 84 - 100% and false alarms are low, except when tested on S1'.

9.4 False-alarm rate

It is possible to make the false-alarm rate arbitrarily low by setting a high threshold. However, this is likely to make the detection rate low as well. In the same way, the detection

rate can be made very close to 100% in most cases, thus ensuring that all targets are noted. But this is likely to cause an unacceptably high false alarm rate. Therefore P_{fa} and P_d must never be viewed in isolation, but as a pair. P_r shows the true performance of the classifier in terms of the percentage of correctly-classified objects.

9.5 Training sets

The need to have an adequate training set is exemplified in the results above. The poor test result for S1' was analysed by investigating the nature of the failures. It was found that there were features present in set S1' which were not present in set S2 and this gave rise to false alarms. In other words, the neural network had not been told how to classify these new objects in S1' and by generalisation, it classified them as targets.

When the network was trained using a set which included the new objects (line 3 above), the performance of the network improved considerably. What this means is that the network was given the information, during training, which enabled it to discriminate between the new clutter objects and real targets.

When there was an imbalance in the number of available samples between the classes, it was necessary to adopt a differential training rate to compensate [9].

9.6 Feature extraction

In the system above, the hot-spot detector and the PCA process were used to reduce the dimensionality of the input and to enhance the classification performance of the neural network. This degree of processing was chosen so that the recognition system could operate in real-time. Other ATR systems have used a more comprehensive set of features as shown in the table below which is extracted from Priddy et.al. [10], for example.

The table shows the features that were used by the authors in two systems, forward-looking infra-red (FLIR) and laser radar (LADAR) images.

FLIR	LADAR	Feature	Description
•	•	Complexity	Ratio of border pixels to total object pixels
•	•	Length/width	Ratio of object length to width
•	•	Mean contrast	Contrast ratio of object's mean to local background mean
•		Maximum brightness	Maximum brightness on object
•		Contrast ratio	Contrast ratio of object's highest pixel to its lowest
•	•	Difference of means	Difference of object and local background means
•	•	Standard deviation	Standard deviation of pixel values on object
•		Ratio bright pixels/total pixels	Ratio of number of pixels on object within 10% of maximum brightness to total object pixels
•	•	Compactness	Ratio of number of pixels on object to number of pixels in rectangle which bounds object
	•	Length	Length of object in pixels
	•	Height	Height of object in pixels

Table 2. Features evaluated by Priddy et.al. [10]

10. pRAM neural networks

The discussion above concentrates on the preprocessing techniques suitable for ATR systems. Both the applications above use the hardware-realisable pRAM neurocomputer. The results presented above have been obtained from pRAM systems using reinforcement training.

The pRAM is a hardware-realisable model of an artificial neuron which generates an output in the form of a spike train. Synaptic weights are realised as multiple stored firing probabilities in the pRAM. These probability values are held in RAM and are therefore readily modified. This probability of firing corresponds to the quantal release of neurotransmitter at each synapse. Being RAM-based, the pRAM can implement non-linear functions within each neuron. The pRAM has also been shown to generalise after training [8], by virtue of the probabilistic spike trains which represent inter-neuron activity.

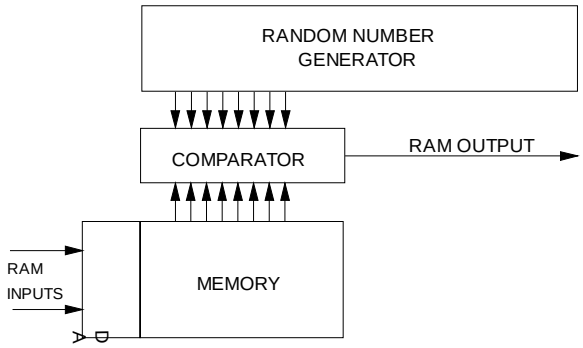

Figure 6. The pRAM neuron architecture

The *p*RAM-256 [11, 12] is a VLSI (integrated circuit) neural network processor with an on-chip learning unit. It offers the flexibility of a software solution with the speed of hardware. Connections between the *p*RAM neurons are reconfigurable so that a network's architecture may be modified at any time.

The *p*RAM-256 can complete one pass of the training process, training all 256 *p*RAMs, in less than 0.25 ms when operating at the maximum clock speed of 33 MHz. Because of the high number of *p*RAMs supported by the *p*RAM-256, a typical neural network can be built using a single *p*RAM Module. Several *p*RAM Modules can operate in parallel so that larger networks can be built.

These features have been exploited in the above ATR systems, making these systems fast, compact and hardware-realisable. For example, a pRAM system can classify image patches as targets or clutter within the time taken to acquire a new video frame, at a rate of 50 frames per second.

11. Conclusion

Two methods of automatic target recognition (ATR) using artificial neural networks have been described. These are, target position detection and target classification. The performance of such ATR systems which use pRAM neural networks has been shown to of a high standard.

It is important to understand the significance of the results obtained from an ATR system. The CEP estimate in the first example quantifies the performance of that system. Whilst the results given were correct, the CEP estimate is required to determine how good this result really is.

When poor results are obtained in an ATR system, for example the high false-alarm rate described above, it is necessary to discover the cause of the errors, if possible. In the

example given, it was revealed that the training set was not sufficiently rich in samples of a given class.

It is clear that preprocessing of the input data is the most important factor in obtaining acceptable results and a number of preprocessing techniques have been given above and in the references.

12. References

[1] Wolberg G, Digital Image Warping, IEEE Computer Society Press, New York, 1990.

[2] Fukunaga K, Introduction to statistical pattern recognition, Academic Press, Boston, 1990

[3] Guan Y, Clarkson T G, Taylor J G, Gorse D, "Noisy reinforcement Training for pRAM Nets", *Neural Networks*, Vol 7, 523-538, 1994.

[4] Ramanan S, Petersen R, Clarkson T G and Taylor J G , "pRAM nets for detection of small targets in sequences of infra-red images", *Neural Networks* (to appear) 1995.

[5] Daniell C E, Kemsley D H, Lincoln W P, Tackett W A, Baraghimian G A, "Artificial neural networks for automatic target recognition", *SPIE Optical Engineering Journal*, Vol 31, No 12, 2521-2531, 1992

[6] Gilmore J F, Czuchry A J, "Application of neocognitron in target recognition", Proc. INNC-90, Vol 2, 15-18, 1990.

[7] Katz A J, Gateley M T, Collins D R, "Robust classifiers without robust features", *Neural Computation*, Vol 2, 472-479, 1990.

[8] Clarkson T G, Guan Y, Gorse D and Taylor J G, "Generalisation in Probabilistic RAM Nets", *IEEE Transactions on Neural Networks*, Vol 4, No 2, 360-364, 1993.

[9] Ramanan S, Petersen R, Clarkson T G and Taylor J G, "Adaptive learning rate for training pyramidal pRAM nets", Proc. ICANN'94, Sorrento, Vol 2, 1360-1363, 1994.

[10] Priddy K L, Rogers S K, Ruck D W, Tarr G L, Kabrisky M, "Bayesian selection of important features for feedforward neural networks", *Neurocomputing*, Vol 5, 91-103, 1993.

[11] Clarkson T G, Gorse D, Taylor J G, Ng C K, "Learning Probabilistic RAM Nets Using VLSI Structures", *IEEE Transactions on Computers*, Vol 41, **12,** 1552-1561, 1992.

[12] T G Clarkson, C K Ng, Y Guan, "The pRAM: An Adaptive VLSI Chip", *IEEE Transactions on Neural Networks*, Vol 4, No 3, 408-412, 1993.