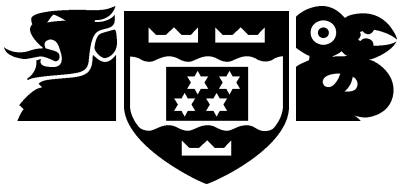
VICTORIA UNIVERSITY OF WELLINGTON
Te Whare Wananga o te Upoko o te Ika a Maui



School of Mathematical and Computing Sciences
Computer Science

Neural Networks for Mining Multiple Class Objects in Image Data

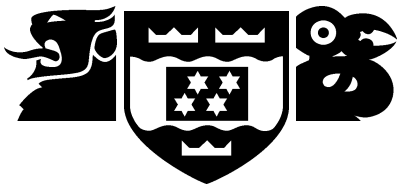
Mengjie Zhang

Technical Report CS-TR-02/01
January 2002

School of Mathematical and Computing Sciences
Victoria University
PO Box 600, Wellington
New Zealand

Tel: +64 4 463 5341
Fax: +64 4 463 5045
Email: Tech.Reports@mcs.vuw.ac.nz
http://www.mcs.vuw.ac.nz/research

VICTORIA UNIVERSITY OF WELLINGTON
Te Whare Wananga o te Upoko o te Ika a Maui



School of Mathematical and Computing Sciences
Computer Science

PO Box 600
Wellington
New Zealand

Tel: +64 4 463 5341, Fax: +64 4 463 5045
Email: Tech.Reports@mcs.vuw.ac.nz
http://www.mcs.vuw.ac.nz/research

Neural Networks for Mining Multiple Class Objects in Image Data

Mengjie Zhang

Technical Report CS-TR-02/01
January 2002

Abstract

Neural networks have been widely applied to data mining since the late 1980s. However, they are often criticised and regarded as a "black box" due to the lack of interpretation ability. This paper describes a domain independent approach to the use of neural networks for mining multiple class objects in large images.

During the object mining process, both the classes and locations of the objects are determined. The results suggest that this approach can be used to mine simple and regular objects with translation and limited rotation invariance in large images against a relatively uniform background.

The network behaviour is interpreted by analysing the weights in learned networks. Visualisation of these weights not only gives an intuitive way of representing hidden patterns encoded in neural networks for object mining problems, but also shows that neural networks are not just a black box but an expression or a model of hidden patterns extracted/discovered during the data mining process.

Keywords Data mining, object recognition, object classification, target detection, weight visualisation.

Author Information

The author is a lecturer in computer science, Victoria University of Wellington, New Zealand.

Neural Networks for Multiclass Object Mining

Mengjie Zhang

School of Mathematical and Computing Sciences
Victoria University of Wellington
P.O. Box 600, Wellington, New Zealand

mengjie@mcs.vuw.ac.nz

Abstract

Neural networks have been widely applied to data mining since the late 1980s. However, they are often criticised and regarded as a "black box" due to the lack of interpretation ability. This paper describes a domain independent approach to the use of neural networks for mining multiple class objects in large images and shows neural networks are not just a black box.

In this approach, the networks use a square input field which is large enough to contain every single object of interest and are trained by the back propagation algorithm on examples which have been cut out from the large images. The trained networks are then applied, in a moving window fashion, over the large images to mine/detect the objects of interest. During the mining process, both the classes and locations of the objects are determined. This approach has been examined on three multiple class object mining problems of increasing difficulty. The results suggest that this approach can be used to mine simple and regular objects with translation and limited rotation invariance in large images against a relatively uniform background.

The network behaviour is interpreted by analysing the weights in learned networks. Visualisation of these weights not only gives an intuitive way of representing hidden patterns encoded in neural networks for object mining problems, but also shows that neural networks are not just a black box but an expression or a model of hidden patterns extracted/discovered during the data mining process.

Keywords Network training, network testing, network sweeping, data mining, object detection, object recognition, object classification, target detection, weight visualisation.

1 Introduction

As more and more data is collected as electronic form, the need for data mining is increasing extraordinarily. Due to the high tolerance to noisy data and the ability to classify unseen data on which they have not been trained, neural networks have been widely applied to data mining [2, 5, 7].

However, neural networks have been criticised for their poor interpretability, since it is difficult for humans to interpret the symbolic meaning behind the learned network weights and the network internal behaviour. For this reason, a neural network is often regarded as a black box classifier or a prediction engine [5]. In this paper, we argue that it is not always true, particularly for data mining in image data. We use the "weight matrices" to represent/interpret the "hidden patterns"[1] in learned networks for multiple class object mining problems.

This paper addresses the problem of mining a number of different kinds of small objects in a set of large images. For example, it may be necessary to find all tumors in a database of x-ray images, all cyclones in a database of satellite images or a particular face in a database of photographs. The common characteristic of such problems can be phrased as "Given $subimage_1$, $subimage_2$, ..., $subimage_n$ which are examples of the object of interest, find all images which contain this object and its location(s)". Figure 1 shows examples of problems of this kind. In the problem illustrated by Figure 1 (b), we want to find centers of all of the 5 cent and 20 cent coins and determine whether the head or the tail side is up. Examples of other problems of this kind include target detection problems [4, 14, 15] where the task is to find, say, all tanks, trucks or helicopters in an image. Unlike most of the current work in the object mining/detection area, where the task is to find only objects of one class [4, 10, 11], our objective is to mine objects from a number of classes.

Neural networks have been applied to object classification and mining problems [1, 9, 10, 13]. In these

[1] The term *pattern* appeared in this paper refers to some high level description of knowledge, such as features, rules, or other expression or model of data. It is quite different from the term "pattern" commonly used in pattern recognition.

approaches, various features/attributes such as brightness, colour, size and perimeter are extracted from the sub-images of the objects and used as inputs to the networks. These features are usually quite different and specific for different problem domains. Extracting and selecting good features is very time consuming and programs for feature extraction and selection often need to be hand-crafted. The approach described in this paper directly uses raw pixels as inputs to the networks.

1.1 Multiclass Object Mining

Multiclass object mining here refers to the detection of small objects of a number of classes in large images. It consists of both *object classification*, which determines the classes of the objects of interest, and *object localisation*, which identifies the positions of all the objects in the large images. This problem is also known as *multiclass object detection*.

Performance in object mining/detection is measured by detection rate and false alarm rate. The detection rate is the number of objects correctly reported as a percentage of the total number of real objects and false alarm rate is the number of non-objects incorrectly reported as objects as a percentage of the total number of real objects. For example, a mining/detection system looking for grey squares in Figure 1 (a) may report that there are 25. If 9 of these are correct the detection rate will be $(9/18) * 100 = 50\%$. The false alarm rate will be $(16/18) * 100 = 88.9\%$. It is important to note that mining/detecting objects in images with very cluttered backgrounds is an extremely difficult problem and that false alarm rates of 200-2,000% are common [11, 13]. Also note that most research which has been done in this area so far either only presents the results of object classification where all the objects have been properly localised and segmented, or only gives the object localisation results where all objects of interest belong to a single class. The results presented in this paper are the performance for the whole mining/detection problem (both object localisation and classification). Thus, it is inappropriate to simply compare the results in this paper with those obtained by other systems which are developed only for the classification problem. It is also inappropriate to directly compare the performance achieved in this paper for multiclass detection problems with the performance for one class detection problems reported in other papers.

1.2 Goals

The overall goal of this paper is to investigate a domain independent approach to the use of neural networks for mining multiple class objects in large images and to investigate a way of interpreting weights in learned networks. Instead of using specific image features, this approach directly uses raw image pixels as inputs to neural networks. This approach is detailed in Section 3. This approach will be examined on a sequence of object mining/detection problems of increasing difficulty (see Section 2.1). Specifically, we investigate:

- Will this approach work for a sequence of multiclass object mining problems of increasing difficulty?
- Will the performance deteriorate as the degree of difficulty of the detection problems increases?
- Can the weights in learned networks be interpreted in some ways and "hidden patterns" be successfully discovered and represented?
- Will the number of training examples affect object mining performance?

In the remainder of this paper, we first describe the image databases and our neural network approach, then present a set of experimental results. After making a detailed analysis and interpretation of the learned network weights, we conclude this approach and give a number of future directions.

2 The Image Databases

2.1 Three Databases

We used three different databases in the experiments. Example images and key characteristics are given in Figure 1. The images were selected to provide object mining problems of increasing difficulty. Database 1 (Easy) was generated to give well defined objects against a uniform background. The pixels of the objects were generated using a Gaussian generator with different means and variances for each class. There are three classes of small objects of interest in this database: black circles ($class1$), grey squares ($class2$) and white circles ($class3$). The coin images (database 2) were intended to be somewhat harder and were taken with a CCD camera over a number of days with relatively similar illumination. In these images the background varies slightly in different areas of the image and between images and the objects to be detected are more complex, but still regular. There are four object classes of interest: the head side of 5 cent coins (class $head005$), the head side of 20 cent coins (class $head020$), the tail side of 5 cent coins (class $tail005$) and the tail side of 20 cent coins (class $tail020$). All the objects in each class have a similar size. They

Number of Images: 7	Number of Images: 20	Number of Images: 15
Number of Objects: 240	Number of Objects: 600	Number of Objects: 164
Number of Classes: 4	Number of Classes: 5	Number of Classes: 3
Image Size: 700x700	Image Size: 640x480	Image Size: 1024x1024
Input Field Size: 14x14	Input Field Size: 14x14	Input Field Size: 14x14
Easy (Circles and Squares)	Medium Difficulty (Coins)	Very Difficult (Retinas)
(a)	(b)	(c)

Figure 1: Object Detection Problems of Increasing Difficulty

are located at arbitrary positions and with different rotations. The retina images (database 3) were taken by a professional photographer with special apparatus at a clinic and contain very irregular objects on a very cluttered background. The objective is to find two classes of retinal pathologies – haemorrhages (class *haem*) and micro aneurisms (class *micro*). To give a clear view of representative samples of the target objects in the retina images, one sample piece of these images is presented in Figure 2. In this figure, haemorrhage and micro-aneurism examples are labeled using white surrounding squares. Note that in each of the databases the background (*non-object*) counts as a class (class *other*), but not a class of interest.

2.2 Image Data Sets

To avoid confusion, we define a number of terms related to the image data.

A set of entire images in a database constitutes an *image data set* for a particular problem domain. In this paper, it is randomly split into two parts: a *detection training set*, which is used to learn a detector, and a *detection test set*, which is used for measuring object detection performance. *Cutouts* refer to the object examples which are cut out from a detection training set. These cutouts form a *classification data set*, which is randomly split into two parts: a *classification training set* used for network training, and a *classification test set* for network testing. The latter set plays two roles:

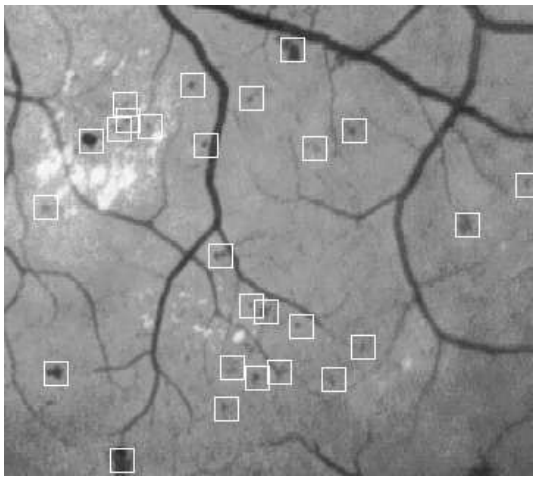


Figure 2: An enlarged view of one piece of the retina images

one is to measure object classification performance, the other is used as a "tuning/validation set" for monitoring network training process in order to obtain good parameters of the learned network for object mining in large images. An *input field* refers to a square within a large image. It is used as a moving window for the network sweeping (detection) process. The size of the input field is the same as that of the cutouts for network

training. The relationships between the various data sets are shown in Figure 3.

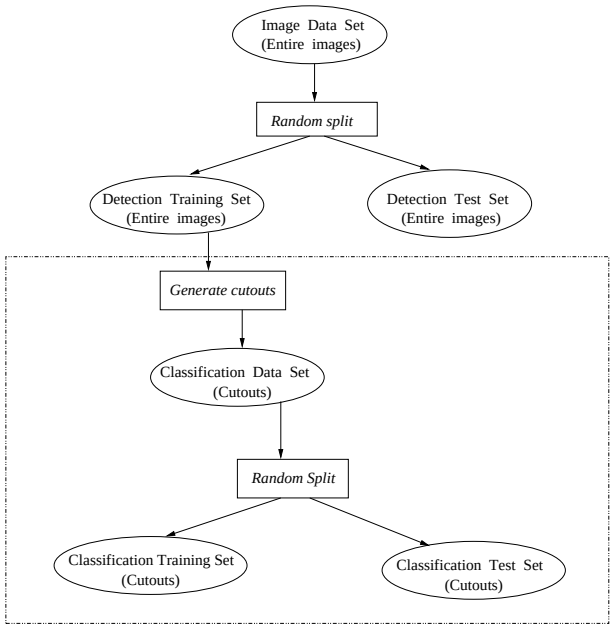


Figure 3: Relationship between the classification and detection data sets

3 The Method

3.1 Overview

An overview of the approach is shown in Figure 4. A brief outline of the method is as follows:

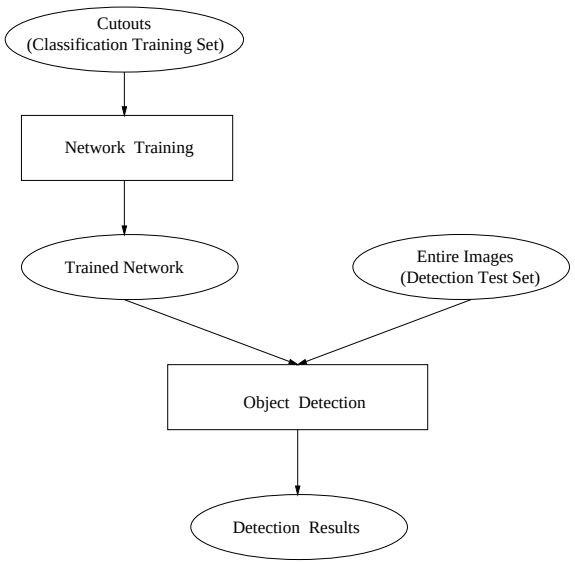


Figure 4: Overview of the approach

1. Assemble a database of images in which the locations and classes of all the objects of interest are manually determined. Divide these full images into two sets: a *detection training set* and a *detection test set*.

2. Determine an appropriate size ($n \times n$) of a square which will cover all single objects of interest and form the input field of the networks. Generate a classification data set by cutting out squares of size $n \times n$ from the detection training set. Each of the squares, called *cutouts* or *sub-images*, only contains a single object and/or a part of the background. Randomly split these cutouts into a classification training set and a classification test set.

3. Determine the network architecture. A three layer feed forward neural network is used in this approach. The $n \times n$ pixel values form the inputs of the training data and the classification is the output. One one hidden layer is used in this approach.[2] The number of hidden nodes is empirically determined.

4. Train the network by the back propagation algorithm [12] on the classification training data. The trained network is then tested on the classification test set to measure the object classification performance. The classification test data is also used to avoid overtraining the network. This step is designed to find the best trained network for object mining/detection.

5. Use the trained network as a moving window template to mine/detect the objects of interest in the detection test set. If the output of the network for a class exceeds a given threshold then report an object of that class at the current location. It is important to note that the thresholds for different classes are different.

6. Evaluate the object mining performance of the network by calculating the detection rate and the false alarm rate.

3.2 Object Mining/Detection

While object classification corresponds to network training and network testing on the cutouts in the classification data sets, object mining/detection corresponds to network sweeping on the entire images in the detection test set, which were not used in any

[2] The theoretical results provided by Irie and Miyake [6] and Funahashi [3] have proved that any continuous mapping can be approximated by a network with a single hidden layer.

form for network training. *Network sweeping* involves both object classification and localisation.

During network sweeping, the successfully trained neural network is used as a template matcher, and is applied, in a moving window fashion, over the large images to detect the objects of interest. The template is swept across and down these large images, pixel by pixel in every possible location.

After the sweeping process is done, an *object sweeping map* for each detected object class will be produced. An object sweeping map corresponds to a grey level image. Sample object sweeping maps for *class1*, *class2* and *class3* together with the original image for the easy detection problems are shown in Figure 5. During the sweeping process, if there is no match between a square in an image and the template, then the neural network output is 0, which corresponds to black in the sweeping maps. A partial match corresponds to grey on the centre of the object, and a good match is close to white.

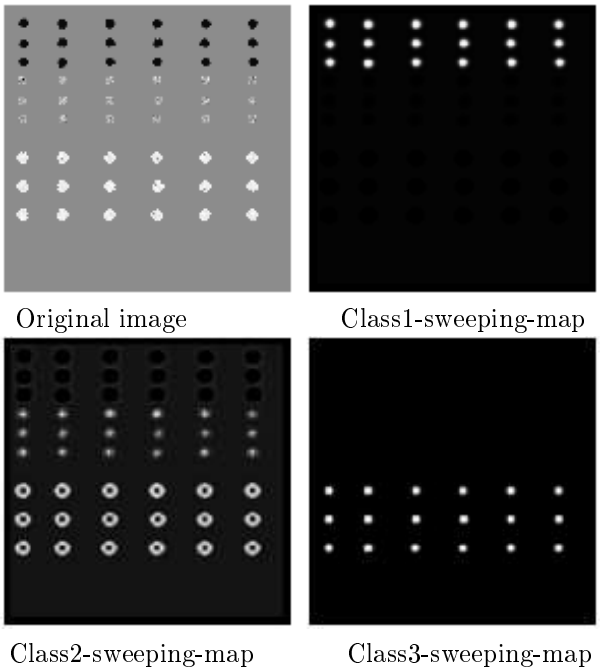


Figure 5: Sample object sweeping maps in object mining/detection.

In the sweeping maps, if a pixel value at a location is greater than or equal to a *threshold*, then report an object at that location. If two or more "objects" for different classes at the same position are found, the decision will be made according to the network activations at this position. For example, if one object for *class2* and one for *class3* at position *(260, 340)* for the easy detection problem are reported and the activations for the three classes of interest and the background at this position are `(0.27, 0.57, 0.83, 0.23)`, then the object for *class3* will be considered the mined/detected object at this position since the activation for this class is the biggest (`0.83`).

4 Results

4.1 Object Classification Results

To classify the object cutouts for a particular problem, the number of hidden nodes of the network needs to be empirically determined. We tried a series of numbers of hidden nodes for the three object mining problems: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 13, 15, 20, 30, 50, 100, 150, 200, 300, and 500. The experiments indicated that 196-4-4,[3] 576-3-5 and 256-5-3 with a set of learning parameters gave the best performance for the easy, the coin, and the retina problems, respectively.

Network training and testing results for object classification are presented in Table 1. In all cases, the network training and testing procedure was repeated 15 times and the average results are presented. Line 1 shows that the best network for the easy images is 196-4-4, the average number of epochs used to train the network and get it converged is 199.40, and the trained network can achieve 100% accuracy on the cutouts of both the classification training set and test set. For the coin images, we can also achieve the ideal performance for object classification. However, this is not the case for the retina images, where only 71.83% accuracy was obtained on the test set of the cutouts.

4.2 Object Mining/Detection Results

This section describes the detection performance of this approach on the three detection problems. For each problem, the 15 trained networks obtained in object classification are used to mine/detect objects in the detection test set and the average results are presented. During the object mining procedure, various thresholds result in different detection results. The higher the threshold, the fewer the objects that can be detected by the trained network, which results in a lower detection rate but also a lower false alarm rate. Similarly, the lower the threshold is selected, the higher the detection rate and the higher the false alarm rate are produced. Thus there is generally a trade-off between the detection rate and the corresponding false alarm rate.

The best detection rates and the corresponding false alarm rates for the three classes in the easy images are presented in Table 2. The best detection rates

[3] 196-4-4 refers to a feed forward network architecture with 196 input nodes, 4 hidden nodes and 4 output nodes. In this paper, we use a similar way to express other network architectures.

Image Databases	Network Architecture	Training Epochs	Training Accuracy	Test Accuracy
Easy Images	196-4-4	199.40	100%	100%
Coin Images	576-3-5	234.6	100%	100%
Retina Images	256-5-3	475.8	81.62%	71.83%

Table 1: Object classification — Network training and testing results for the three databases.

achieved for all the three classes are 100%, showing that this approach can successfully detect all the objects of interest in this database. At this point, detecting *class1* (black circles) and *class 3* (white circles) did not produce any false positives (false alarm rates are zero), while detecting *class2* (grey squares) resulted in a 91.2% false alarm rate on average.

Easy Images	Object Classes		
	Class1	Class2	Class3
Detection Rate(%)	100	100	100
False Alarm Rate(%)	0	91.2	0

Table 2: Object mining/detection results for the easy images.

The detection results for the coin images are described in Table 3. As in the easy images, the trained neural networks achieved 100% detection rates for all the classes, showing that this approach correctly detects all the objects of interest from different classes in the coin images. In each run, it was always possible to find a threshold for the network output for class *head005* and *tail005* which resulted in detecting all of the objects of these classes with no false alarms. However, detecting classes *head020* and *tail020* was a relatively difficult problem. The average false alarm rates for the two classes at a 100% detection rate were 182% and 37.5% respectively.

Compared with the performance of the easy and coin images, the results of the two classes *haem* and *micro* in the very difficult retina images are disappointing. All the objects of class *micro* were correctly detected (a detection rate of 100%), however, with a very high false alarm rate. The best detection rate for class *haem* was only 73.91%. Even at a detection rate of 50%, the false alarm rate was still quite high.

To give an intuitive view, the extended ROC curves (Receiver/Relative Operating Characteristic curve [8, 16]) for class *class2* in the easy images, class *head020* in the coin images and classes *haem* and *micro* in the retina images are presented in Figure 6 (a), (b), (c) and (d) respectively.

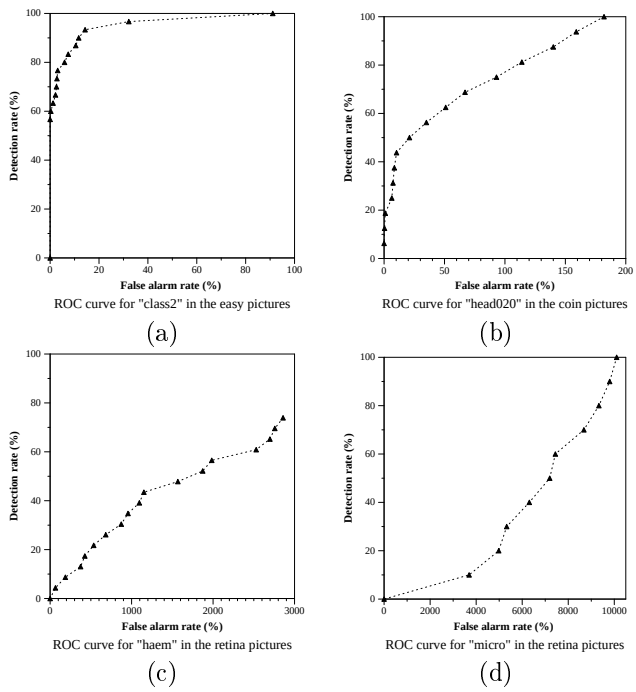


Figure 6: Detailed results (extended ROC curves) for some "difficult" classes in the three databases

4.3 Analysis of Results

As can be seen from the detection results obtained here, it was always possible to detect all objects of interest in the easy and the coin images. This reflects the fact that the objects in the two databases are simple or regular and the background is uniform or relatively uniform. While detecting objects in the easy images only resulted in a few false alarms, detecting objects in the coin images resulted in a relatively higher false alarm rate. This is mainly because the mining/detection problems in the coin images are more difficult than in the easy images.

Due to the high degree of difficulty of the detection problems, the results for the retina images are not good. For class *micro*, while all objects were correctly detected, a very high number of false alarms were pro-

Coin Images	Object Classes			
	head005	tail005	head020	tail020
Detection Rate (%)	100	100	100	100
False Alarm Rate (%)	0	0	182	37.5

Table 3: Object mining/detection results for the coin images.

duced. This is mainly because these objects are irregular and complex and the background is highly cluttered. For class *haem*, it was not possible to detect all objects of interest (the best detection rate was 73.91%). This is mainly due to the size variance of these objects (from *7×7* to *14×14* pixels). Another reason that we did not obtain good results on the retina images is the insufficient number of training object examples (only 164, see Figure 1).

Comparing the results obtained in object classification and object mining/detection, it can be found that object classification results are better than the object mining results for all the three databases. In the coin images, for example, the trained networks achieved perfect object classification results, that is, 100% accuracy on classifying all the objects in the four classes. While mining 5 cent coins achieved ideal performance, that is, all objects of the two classes (*head005, tail005*) were correctly mined with zero false alarm rate, mining the two 20 cent coin classes (*head020, tail020*) produced a number of false positive objects. The results for the other databases showed a similar pattern. This is due mainly to the fact that multiclass object mining/detection task is generally much more difficult than only the classification task on the same problem domains since multiclass object mining includes both object classification and object localisation.

5 Visualisation and Analysis of Learned Network Weights

To analyse why this approach can be used for multiclass object mining/detection, this section interprets the network internal behaviour through visual analysis of the weights in the trained networks. For presentation convenience, we use the trained networks for regular object detection in the coin images. Most other networks contained similar patterns.

Figure 7 shows the network architecture (576-3-5) used in the coin images. The weight groups between the input nodes and the hidden nodes and between the hidden nodes and the output nodes are also presented. The weight matrices (a), (b), (c) and (d) shown in Figure 8 correspond to weight groups (a), (b), (c) and (d) in the network architecture in this figure.

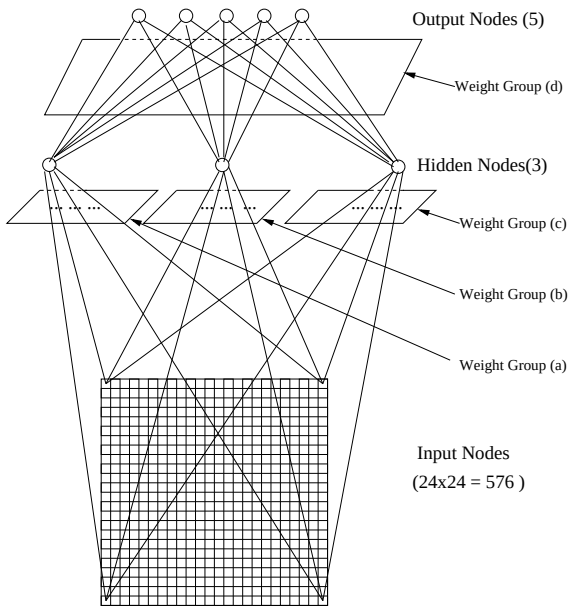
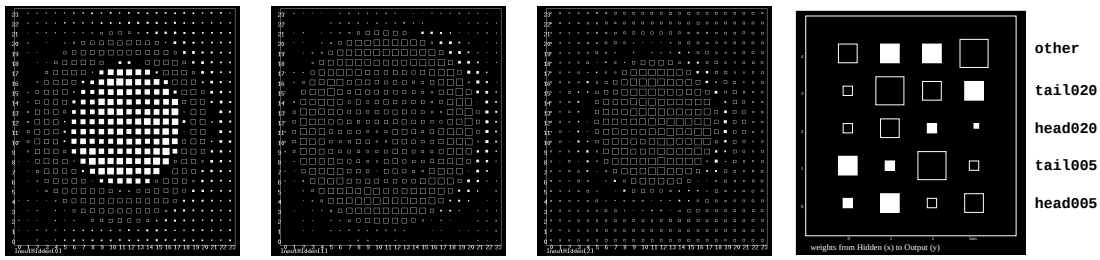


Figure 7: Network architecture with four weight groups for object mining/detection in the coin images.

Figure 8 shows the weights from a trained 576-3-5 network which has been successfully applied to the coin images. In this figure the full squares represent positive weights and the outline squares represent negative weights, while the size of the square is proportional to the magnitude of the weight. Matrices (a), (b) and (c) show the weights from the input nodes to the first, the second and the third hidden nodes. The weights are shown in a *24×24* matrix (corresponding to the *24×24* input field) to facilitate visualisation. Figure 8 (d) shows the weights from the hidden nodes to the output nodes and the biases of these output nodes. The five rows in this matrix correspond to the classes *other, tail020, head020, tail005* and *head005*. The first three of the four columns correspond to weights from the three hidden nodes (associated with weight matrices (a), (b) and (c)) to the five output nodes. The last column corresponds to the biases of the five output nodes.

Inspection of the first column of Figure 8 (d) reveals that weight matrix (a) has a positive influence on 5 cent coins and a negative influence on 20 cent coins. It has a



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

Figure 8: Weights in a trained network for object mining/detection in the coin images.

strong influence on class *tail005* but a weak influence on class *head005*. The same matrix has a strong negative effect on class *other* (background). Inspection of the second column reveals weight matrix (b) has a positive effect on the 5 cent coins and a negative effect on the 20 cent coins. Moreover, it has a strong influence on class *head005* but a week influence on class tail005. Also it has a very strong positive influence on class *other*. This indicates that the combination of matrices (a) and (b) not only can separate the 5 cent coins from the 20 cent coins and the background, but also can discriminate the 5 cent tails from the 5 cent heads. Inspection of the third column reveals that the weight matrix (c) has a strong negative influence on the tails of both 5 cent and 20 cent coins and a week influence on the heads of both 5 cent and 20 cent coins. It also strongly supports the background and has a strong negative influence on the tails of 5 cent coins. The fourth column suggests that the biases also play a complementary role for mining objects, particularly for class *tail020*, class *other* and class *head005*. If we regard the nodes of the hidden layers as representing feature detectors learnt by the network, then Figures 8 (a)-(c) are a visual representation of these features. Visually these features 'make sense' as there are regions corresponding to the 5 cent coins, the annulus remaining to the background when a 5 cent coin is 'removed' from the centre of a 20 cent coin.

For object mining in large images described here, the weight matrices can be considered "hidden patterns" encoded in learned neural networks. Visualisation of the learned network weights reveals that patterns contained in learned networks can be intuitively represented, which strongly supports the idea that neural networks are not just a "black box", but a model or an expression of patterns discovered during learning.[4]

[4] A mathematical model of the patterns in learned neural networks can be seen from Appendix A.

6 Conclusions

This paper presented a pixel based approach for mining multiple class objects in large images using multilayer feed forward neural networks. The back propagation algorithm was used to train the network on the sub-images which had been cut out from the large images. The trained network was then applied, in a moving window fashion, over the entire images to detect the objects of interest. Object detection performance was measured on the large images in the detection test set. The experimental results showed that this approach performed very well for mining a number of simple and regular objects against a relatively uniform background. It did not perform well on the difficult detection problems in the retina images, which indicates that it can not be well suited to detecting complex and irregular objects against a highly cluttered background.[5] As expected, the performance degrades when the approach is applied to object mining problems of increasing difficulty.

Visualisation of the weights in trained neural networks resulting from this approach revealed that trained networks contained feature detectors which "made sense" for the problem domain and could discriminate objects from different classes. This provides a way of revealing hidden patterns in learned neural networks for object mining problems. This approach also shows that neural networks are not just a black box, but a model of patterns which were discovered through the learning process.

The approach has the following advantages:

- Raw image pixel data are used as inputs to neural networks, and accordingly traditional specific feature extraction and selection is avoided.

[5] Strictly speaking, we can not make this conclusion, since the number of training examples is not sufficient. This will be addressed later in this section.

- It is a domain independent approach and can be directly applied to multiclass object detection problems in different areas.
- Multiple class objects can be mined/detected (classified and localised) in large images with a single trained neural network.
- Patterns encoded in learned neural networks can be visually represented by using weight matrices.

The approach also has a number of disadvantages, which need to be addressed in the future:

- This approach can mine translation and limited rotation invariant objects but cannot successfully detect objects with the size invariance such as class *haem* in the retina images.
- In this approach, the network was trained based on the object cutouts but the trained network was directly applied to the entire images. This might be one of the main reasons which result in many false positives in some difficult detection problems. We are investigating ways of refining the trained networks based on the entire images to improve the detection performance.
- The insufficient number of training examples led to poor object mining performance in the retina images. It would be very interesting to investigate whether this approach can achieve good results using sufficient training examples in the future — this will reveal whether this approach can be used to mine/detect complex and irregular objects against a highly cluttered background.

Acknowledgement

I would like to thank Dr. Peter Andreae, Dr. Dong Qian Wang and Dr. Victor Ciesielski for useful discussions and Chris Kamusinski who provided and labelled the retina images.

References

[1] David P. Casasent and Leonard M. Neiberg. Classifier and shift-invariant automatic target recognition neural networks. *Neural Networks*, Volume 8, Number 7/8, pages 1117–1129, 1995.

[2] Usama Fayyad, Gregory Piatetsky-Shapiro and Padhraic Smyth. From data mining to knowledge discovery in databases. *AI Magazine*, Volume 17, Number 3, pages 37–53, 1996.

[3] Ken-Ichi Funahashi. On the approximate realization of continuous mappings by neural networks. *Neural Networks*, Volume 2, Number 3, pages 183–192, 1989.

[4] Paul D. Gader, Joseph R. Miramonti, Yonggwan Won and Patrick Coffield. Segmentation free shared weight neural networks for automatic vehicle detection. *Neural Networks*, Volume 8, Number 9, pages 1457–1473, 1995.

[5] Robert Groth. *Data Mining: Building Competitive Advantage*. Prentice Hall PTR, 2000.

[6] B. Irie and S. Miyake. Capability of three-layered perceptrons. In *Proceedings of the IEEE 1988 International Conference on Neural Networks*, pages I(641–648), New York, 1988. IEEE. Vol. 1.

[7] Jiawei Jan and Michaeline Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers, 2001.

[8] Charles E. Metz. ROC methodology in radiologic imaging. *Investigative Radiology*, Volume 21, Number 9, pages 720–732, September 1986.

[9] Steven K. Rogers, John M. Colombi, Curtis E. Martin, James C. Gainey, Ken H. Fielding, Tom J. Burns, Dennis W. Ruck, Matthew Kabrisky and Mark Ocley. Neural networks for automatic target recognition. *Neural Networks*, Volume 8, Number 7/8, pages 1153–1184, 1995.

[10] H. L. Roitblat, W. W. L. Au, P. E. Nachtigall, R. Shizumura and G. Moons. Sonar recognition of targets embedded in sediment. *Neural Networks*, Volume 8, Number 7/8, pages 1263–1273, 1995.

[11] Michael W. Roth. Survey of neural network technology for automatic target recognition. *IEEE Transactions on neural networks*, Volume 1, Number 1, pages 28–43, March 1990.

[12] D. E. Rumelhart, G. E. Hinton and R. J. Williams. Learning internal representations by error propagation. In D. E. Rumelhart, J. L. McClelland and the PDP research group (editors), *Parallel distributed Processing, Explorations in the Microstructure of Cognition, Volume 1: Foundations*, Chapter 8. The MIT Press, Cambridge, Massachusetts, London, England, 1986.

[13] Mukul V. Shirvaikar and Mohan M. Trivedi. A network filter to detect small targets in high clutter backgrounds. *IEEE Transactions on Neural Networks*, Volume 6, Number 1, pages 252–257, Jan 1995.

[14] Allen M. Waxman, Michael C. Seibert, Alan Gove, David A. Fay, Ann Marie Bernandon, Carol Lazott, William R. Steele and Robert K. Cunningham. Neural processing of targets in visible, multispectral ir and sar imagery. *Neural Networks*, Volume 8, Number 7/8, pages 1029–1051, 1995.

[15] Yonggwan Won, Paul D. Gader and Patrick C. Coffield. Morphological shared-weight networks with applications to automatic target recognition. *IEEE Transactions on neural networks*, Volume 8, Number 5, pages 1195–1203, September 1997. ISSN 1045-9227.

[16] Mengjie Zhang. *A Domain Independent Approach to 2D Object Detection Based on the Neural and Genetic Paradigms.* Ph.D. thesis, Department of Computer Science, RMIT University, Melbourne, Australia, August 2000.

A Pattern Modelling

Section 5 visualised the learned features as weight matrices, which can be considered hidden patterns extracted from learning. With the analysis of these patterns, we showed that a learned network was an expression of these patterns. In this appendix, we investigate a formal mathematical model. To do so, we define the following symbols:

$\vec{I}$: the network input data. Its dimension is the number of pixels in the input field or the training object cutout. This number is the same as the number of input nodes of the network. In the standard multilayer feed forward back propagation networks, $\vec{I}$ is also the output of the network input nodes.

$\vec{O}$: the network output vector. Its dimension is the number of output nodes defined in the network.

$\vec{H}$: the output of the hidden nodes. As mentioned earlier, only one hidden layer is used in our approach.

f: the transfer function. In the back propagation algorithm, f is the sigmoid function, that is, $f(x) = \frac{1}{1+e^{-x}}$. Here we extend the transfer function as a vector $\vec{f}$:

$$\vec{f}(\vec{x}) = (\frac{1}{1+e^{-x_1}}, \frac{1}{1+e^{-x_2}}, ..., \frac{1}{1+e^{-x_n}}) \quad (1)$$

where $\vec{x} = (x_1, x_2, ..., x_n)$.

$\vec{W}_H$: the weights between the input layer and the hidden layer. The dimension of this vector is the number of hidden nodes times the number of input nodes.

$\vec{W}_O$: the weights between the hidden layer and the output layer. The dimension is the number of output nodes by the number of the hidden nodes.

Based on these definitions, the network output will be:[6]

$$\vec{O} = \vec{f}(\vec{W}_O \cdot \vec{H}) \quad (2)$$

where:

$$\vec{H} = \vec{f}(\vec{W}_H \cdot \vec{I}) \quad (3)$$

Combining Equations 2 and 3, we obtain:

$$\vec{O} = \vec{f}(\vec{W}_O \cdot \vec{f}(\vec{W}_H \cdot \vec{I}) \quad (4)$$

Generally, for any three layer learned network for object mining problems, we have:

$$\vec{W}_H = (\vec{W}_{h_1}, \vec{W}_{h_2}, ..., \vec{W}_{h_m}) \quad (5)$$

where $\vec{W}_{h_1}, \vec{W}_{h_2}, ..., \vec{W}_{h_m}$ are the weight matrices similar to those in Figure 8 (a), (b) and (c), and m is the number of hidden nodes in the learned network. In this case, the network output is:

$$\vec{O} = \vec{f}(\vec{W}_O \cdot \vec{f}((\vec{W}_{h_1}, \vec{W}_{h_2}, ..., \vec{W}_{h_m}) \cdot \vec{I})) \quad (6)$$

For the coin image example, $\vec{I}$ is a 576 dimensional vector which corresponds to a two dimensional array of 24×24 with the values of (normalised) object pixels. $\vec{O}$ is a five dimensional vector which corresponds to the five classes. $\vec{W}_O$ corresponds to the weight matrix $\vec{W}_d$ in Figure 8 (d). $\vec{W}_H$ is a three dimensional vector:

$$\vec{W}_H = (\vec{W}_a, \vec{W}_b, \vec{W}_c) \quad (7)$$

where $\vec{W}_a, \vec{W}_b$ and $\vec{W}_c$ are hidden patterns extracted, corresponding to the weight matrices shown in Figure 8 (a), (b) and (c) respectively. Each of them has 24×24 weights. Thus, for the object mining problems in the coin images, we have:

$$\vec{O} = \vec{f}(\vec{W}_d \cdot \vec{f}((\vec{W}_a, \vec{W}_b, \vec{W}_c) \cdot \vec{I})) \quad (8)$$

Clearly, the network output is a function of $\vec{W}_a, \vec{W}_b, \vec{W}_c, \vec{W}_d$ and $\vec{I}$. Thus, if these weight matrices are regarded as features or patterns extracted from learning, then the learned network is a model of these patterns.

[6] For presentation convenience, we do not put the biases in these equations. This does not change our conclusion.