VICTORIA UNIVERSITY OF WELLINGTON

Te Whare Wananga o te Upoko o te Ika a Maui

School of Mathematical and Computing Sciences

Computer Science

Pixel Statistics Based Neural Networks for Domain Independent Multiclass Object Detection

Mengjie Zhang, Peter Andreae, Roy Chow

Technical Report CS-TR-02/15
November 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

Pixel Statistics Based Neural Networks for Domain Independent Multiclass Object Detection

Mengjie Zhang, Peter Andreae, Roy Chow

Technical Report CS-TR-02/15
November 2002

Abstract

The paper describes a domain independent approach to multiclass object detection problems in which the classes and locations of relatively small objects in large images must be determined. The approach uses low-level pixel statistics rather than domain dependent features as inputs to neural networks. The networks 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 detect the objects of interest. The paper reports on a series of experiments on three object detection problems of increasing difficulty, and with different sets of pixel statistics. The results suggest that this approach can be used successfully to detect simple and regular objects in large pictures against a relatively uniform background. Using *moment* statistics and local region pixel statistics improves multiclass object detection performance significantly.

Keywords Object recognition, object classification, computer vision, multiclass target detection, domain independent detection.

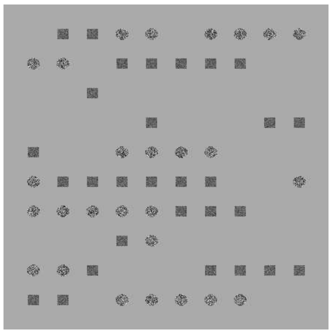
Author Information

Mengjie Zhang and Peter Andreae are members of academic staff in computer science at Victoria University of Wellington. Roy Chow is a graduate student in computer science at Victoria University of Wellington.

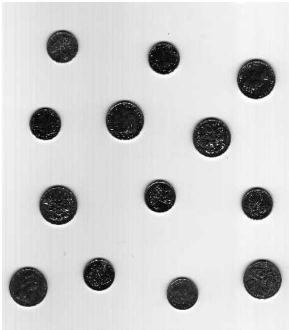
1 Introduction

As more and more image data is captured in electronic form, the need for programs that can find objects of interest in a database of images is increasing. Example tasks include finding tumors in a x-ray images, cyclones in satellite weather images, particular faces in a security video, and identifying different kinds of targets in a photo from a spy plane [10, 12]. Many of these tasks involve objects in visual images that would be easy for humans to identify; others (particularly in medical imaging domains) involve other kinds of images for which human object detection is not very effective.

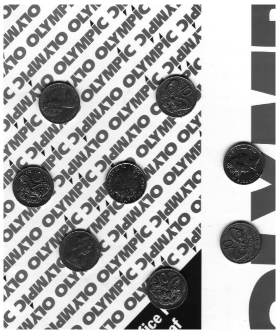
Figure 1 shows some simple examples of problems of this kind. Figures 1 (b) to (e) contain collections of coins. In (b) and (d) the task is to distinguish, and locate the centers of, the 5 cent and 10 cent coins; in (c) and (e) the task is to distinguish and locate the heads-up coins and the tails-up coins. The task in (b) and (d), is very straightforward for humans, even in the low resolution version; the task in (c) is more difficult, and the low resolution version in (e) is almost impossible for a human.



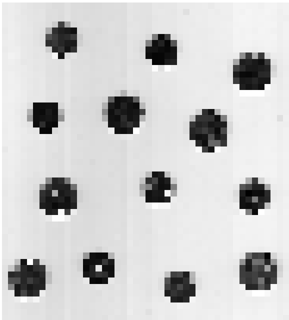
Artificial
(a)



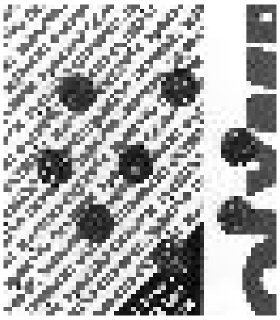
Easy
(b)



More Difficult
(c)



Easy (Low Resolution)
(d)



Low Resolution for (c)
(e)

Figure 1: Object Detection Problems of Increasing Difficulty

The traditional approach to object recognition and detection task [4, 13] usually involves several stages of *preprocessing*, *segmentation*, and *feature extraction* before the final stage of *classification*. The goals of the earlier stages are to transform the image into a representation that makes the classification simpler. Feature extraction, in particular, transforms the very high dimensional pixel based representation of the original image into a much lower dimensional representation in which the objects are much more easily separated from each other and from the background. The design of the preprocessing and feature detection stages is a major component of the work involved in constructing an object detection system using the traditional approach. Unfortunately, the design is usually dependent on the domain, and therefore may have to be repeated for each different task. The features will often be dependent on the particular classes of objects to be detected, and even the early preprocessing stages will depend on the image domain. Typically, using domain independent feature detection and preprocessing produces less effective object detection systems.

The goal of our research is to explore domain independent methods for object detection that exploit the power of learning systems, particularly genetic programming and neural networks, to compensate for the weaknesses of domain independent features. We do not expect these general methods to outperform highly tailored, domain specific systems on any single task. However, the justification for our approach is that the domain independent systems should be cheaper to develop than the domain dependent ones, and therefore may be valuable for the wide range of applications that cannot justify the high development cost of a domain dependent system.

Another goal of our research is to explore the power of genetic programming techniques. Object detection tasks provide a challenging, high dimensional problem domain for exploring the power of this learning techniques. To evaluate the genetic programming techniques, we must be able to compare their performance with appropriate benchmarks. The traditional image processing, domain dependent techniques do not provide an appropriate benchmark; we are therefore using the better understood learning techniques of neural networks[2, 9] to generate benchmarks against which we can compare genetic programming. A side benefit of this approach is developing a better understanding of the power and limitations of neural networks for domain independent object detection. This paper reports on a series of experiments in using neural networks for domain independent object detection.

Neural networks have been applied in object detection by a number of researchers. Some have taken a traditional feature based approach [1, 11] in which a set of domain specific, hand designed, feature detectors are used to transform the image into a low dimensional vector. This feature vector is then used as input to the classification network.

Other researchers [2, 9] have explored a domain independent approach by using the raw pixel values of the image as inputs to the neural network classifier. Although this approach avoids the cost and domain specificity of the preprocessing and feature extraction, it has a problem that the input vector to the neural network is typically very large. The effect of this is to require a large number of training examples and long training time. Although long training time is not necessarily a critical problem, generating a large number of correctly classified training examples may be infeasible in many applications.

Our approach is to investigate an alternative approach using domain independent, statistically based features. Like domain specific features, these *pixel statistics* transform an image to a lower dimensional vector. However, the features are based on standard statistical measures, such as means and moments, of parts of the image, and do not depend on any particular properties of the image domain. They are therefore equally relevant for any kind of image and class of object. A further feature of our approach is that, unlike most of the current work in the object detection area, where the task is to detect only objects of one class [5, 7], our objective is to detect objects from a number of classes.

1.1 Multiclass Object Detection

Object detection refers to the detection of small objects in large pictures. It consists of both object classification, which involves determining the classes of objects of interest, and object localisation, which identifies the positions of all the objects in the large pictures.

In most automatic object detection systems, all the objects of interest are considered to be a *single class*. Such systems address the single class object detection task, which is the problem of distinguishing *objects* or *targets* from *non-targets* or the *background*. In contrast, *multiclass object detection* refers to the case where there is more than one class of objects and both the classes and locations of all the objects must be determined. In general, multiclass object detection problems are harder than single class detection problems, and multiclass detection using a single trained program (such as a neural network) is an even more difficult problem.

1.2 Experimental Goals

The overall goal is to develop a domain independent approach to the use of neural networks for multiclass object detection problems and to investigate what simple pixel statistics can do a good enough job for these tasks. To achieve this goal, we ran a sequence of experiments on multiclass object detection problems of increasing difficulty with different sets of pixel statistics. The specific goals of the experiments were

- To determine whether a neural network using pixel statistics can achieve acceptable performance on multiclass object detection problems.
- To identify which sets of pixel statistics are sufficient for these problems.
- To determine whether *local* regional pixel statistics are required for these problems.

1.3 Performance Evaluation

Performance in object 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 the false alarm rate is the number of objects incorrectly reported as a percentage of the total number of real objects. For example, a detection system looking for grey squares in Figure 1 (a) may report that there are 35. If 15 of these are correct the detection rate will be $(15/29) = 51.7\%$. The false alarm rate will be $(20/29) = 69.0\%$. It is important to note that detecting objects in pictures with very cluttered backgrounds is an extremely difficult problem and that false detection rates of 200%–2,000% (that is, the detection system suggests that there are up to 20 times as many objects as there really are) are common [7, 9].

2 The Image Databases

We used three different databases in the experiments. Example pictures and key characteristics are given in Figure 1. The pictures were selected to provide problems of increasing difficulty. Database 1 ("Artificial") 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 two classes of small objects of interest in this database: squares ($class1$) and circles ($class2$). Database 2 ("Easy") consists of scanned images of New Zealand coins. There are two object classes of interest: the 5 cent coins and 10 cent coins. The objects in each class have a similar size but are are located at arbitrary positions and with different rotations. Database 3 ("More difficult") was created in a similar way to database 2, but the background was not uniform and the objects are all 10c coins. The objective is to detect objects of the two classes: "heads" and "tails". The task is more difficult than that of database 2, both because there is a cluttered background and the system must distinguish the heads and tails of the same sized coins, given a low resolution image. Note that for all the detection problems, the background is also counted as one class for network training, but it is not a class of interest.

3 The Methodology

This section describes the methodology for the use of pixel statistics in neural networks for domain independent, multiclass object detection problems. An overview of the approach is shown in Figure 2.

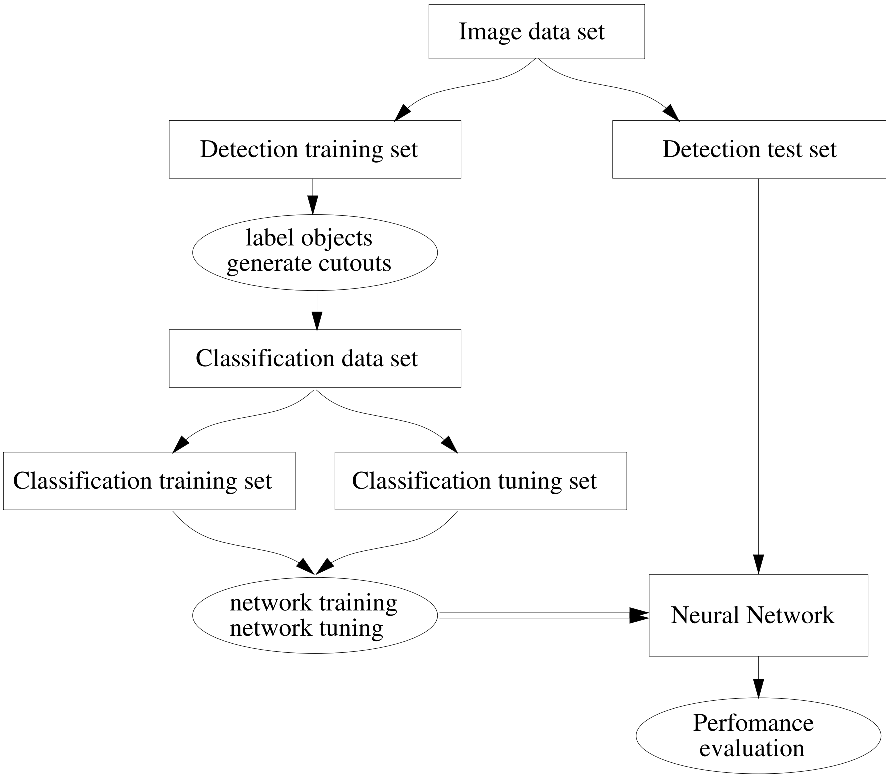


Figure 2: Overview of the approach

3.1 Image Data Preparation

First, the locations of all objects of interest in an image database are manually determined and their classifications are labelled and recorded along with their locations. Then this image database is split into two data sets: one is the *detection training set*, which is used for training the network, and the other is the *detection test set* reserved for evaluating the object detection performance of the network.

The neural network is never applied to an entire image, but only to sub-images of a fixed size. We refer to these sub-images as *input images*. For the network training stage, the input images are created by cutting out samples from images in the detection training set. For the evaluation stage, the system sweeps a window (or *input field*) of the appropriate size over the entire image, creating an input image for each possible position. In both stages, a set of pixel statistics (low level image features) are calculated from the input image and used as inputs to the neural network.

The size of the input images is determined by the size of the object of interest in each of the image databases. Following the heuristic in [14], the input field is chosen to be sufficiently large to characterise the background but small enough to contain only a single object of interest. The largest object in each of the image databases can be used to determine this size (14×14 for the first data set, and 20×20 for the other two).

The set of input images constructed from the detection training set is called the *classification data set.* This data set consists of input images centred on each of the objects in the images in the detection training set, labeled with the object class, plus a collection of input images that are not centred on the objects, and are labeled class *other*. Some of these "other" input images will include regions of the background that do not contain any objects, but it is important that there are also sufficient input images that include parts of objects or objects that are not centred in the input image, since these are necessary for training the network to identify the centres of objects. We construct the input images in the classification data set by hand labeling the centres of all objects in the detection training set, and then automatically constructing one collection of cutouts centered at those points and another collection of cutouts centered at random other positions, with a bias towards positions near but not on the centres of the objects. The classification data set is randomly partitioned into a *classification training set* and a *classification tuning set*.

3.2 Neural Network Architecture

We use three layer, feed forward neural networks with one input layer, one output layer and one hidden layer. The number of input nodes is the number of pixel statistics computed from an input image (see below). The number of output nodes depends on the number of object classes in the image database. The number of hidden nodes is empirically determined through experiments, using the classification tuning set to choose the optimal number of hidden nodes.

3.3 Pixel Statistics

Pixel statistics are low-level, domain independent image features, computed from the pixel intensities in a region of the input image. We varied the statistical properties that we computed, as well as the the set of regions in an input image, as shown in table 1.

Pixel Stat Set	Regions	Pixel Stats	Number of Pixel Stats
I	entire input image	mean, sd, max, min	4
II	entire input image	mean, sd, max, min, moment	5
III	6 local regions	mean, sd, max, min, moment	30

Table 1: Three pixel statistics sets used in neural networks. (sd: standard deviation)

3.3.1 Pixel Statistics Set I

Pixel Statistics Set I was limited to four basic statistical properties computed over the whole input image: *Mean*, *standard deviation*, *Maximum*, and *Minimum*. We note that all four of these measures are translation and rotation invariant, which is significant for all kinds of images. However, they are also invariant to any permutation of the pixels over the region — they contain no location-dependent information at all. We hypothesise is that Set I will be too limited for any interesting object detection problems.

3.3.2 Pixel Statistics Set II

Pixel Statistics Set II augments the statistics in the set I with a first order *moment* statistic:

$$moment = \frac{1}{m}\sum_{i=1}^{m} f(x_i, y_i) \cdot d(x_i, y_i)$$

where $f(x_i, y_i)$ is the value of pixel (x_i, y_i), m is the number of pixels in the region, and $d(x_i, y_i)$ is the distance of the pixel from the centre of the input image. The effect of the moment statistic is to provide some location-dependent information.

3.3.3 Pixel Statistics Set III

Pixel Statistics Set III uses the same statistics as Set II, but the statistics are computed on on 5 "local regions" within the input image (as shown in figure 3) as well as on the whole input image. The local regions provide more location-dependent information, at the cost of increasing the size of the input vector.

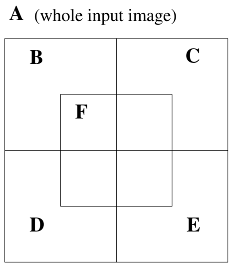


Figure 3: Pixel statistics with local regions.

3.4 Network Training and Testing — Object Classification

We use the backward error propagation algorithm [8] with the variations of online learning and the fan-in factor [3, 6] to train the networks. In the *online learning* procedure, also called the *stochastic gradient procedure*, weight changes are applied to the network after each training pattern. The fan-in factor normalises the weights by dividing them by the number of inputs of the node to which the connection belongs. The normalisation is applied before network training and also to the weight changes during network training.

The training is terminated when the classification accuracy in the classification training set reaches a pre-defined percentage. When training is terminated, the trained network weights and biases are saved for the use in network testing or subsequent resumption of training.

The trained network is then applied to the classification tuning set. If the performance is reasonable, then the trained network is ready to be used for object detection. Otherwise, the network architecture and/or the learning parameters need to be changed and the network re-trained, either from the beginning or from a previously saved, partially trained network. The classification tuning set is also used as a "validation set" for monitoring network training process in order to obtain good parameters of the learned network for object detection.

During network training and tuning, the classification is regarded as correct if the output node with the largest activation value corresponds to the desired class of a pattern.

3.5 Object Detection

While object classification corresponds to network training and network testing on the cutouts in the classification data sets, object detection corresponds to network sweeping on the entire images in the detection test set, which were not used in any form for network training. This involves both object classification and localisation. In this approach, object detection consists of *network sweeping* and *object center finding*.

3.5.1 Network Sweeping

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 pictures to detect the objects of interest. The template is swept across and down these large pictures, pixel by pixel in every possible location. The relationship between the full image, the sweeping window, the input image, the pixel statistics and the trained network is shown in figure 4. As shown in this figure, the sweeping window selects a portion of a full image (in the detection test set) as the input image and passes the raw pixel values of the input image to the pixel statistics computation described earlier. The computed pixel statistics are then provided as inputs to the trained neural network and the network classifies the input image.

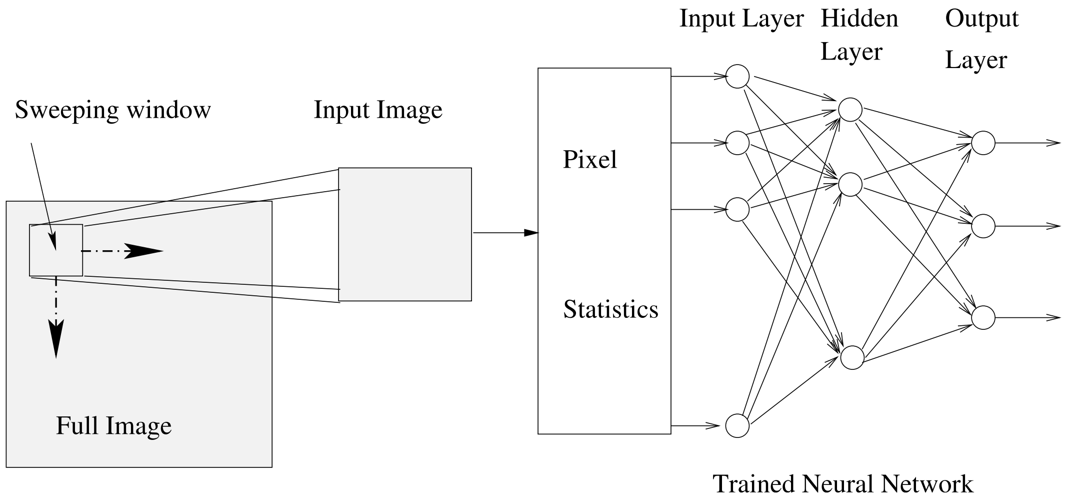
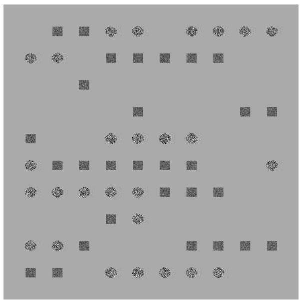


Figure 4: Network sweeping using a training network with pixel statistics as inputs.

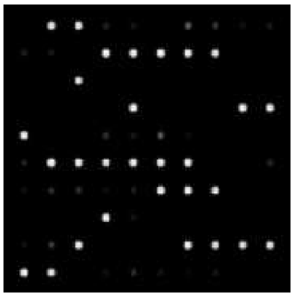
For each object class, the sweeping process will generate an *object sweeping map* — a bitmap representing the output value of the neural network for the given class at each position in the original image. A sample object sweeping map for *class1* (circles) with the original picture for the “Artificial” detection problem is shown in Figure 5. A black pixel in a sweeping map indicates a neural network output of 0 (“object not present”) at that position; a white pixel indicates an output of 1 (“object definitely present”). Grey pixels indicate a partial match to the class at that position.

3.5.2 Finding Object Centres

Typically, a well trained neural network will detect an object not only when the sweeping window is centred over an object, but also when it is within a few pixels of the centre of an object. A simplistic interpretation of the neural network output would therefore suggest that there are multiple objects present in the image within a few pixels of each other, which



Original image



Object sweeping map for class1

Figure 5: Sample object sweeping maps in object detection.

is obviously false. The system contains a *centre-finding algorithm* (figure 6) to analyse the object sweeping maps to find the centres of all objects detected by the trained network.

For each object sweeping map:
• Set all values less than *threshold* to 0
• **while** there are non-zero values in map
– Find largest value, and label position as an object centre.
– Set all values within *delta* of center to zero.

Figure 6: Centre-finding algorithm

It is possible that "object centres" for two (or more) different classes may be found at the same position. Only the class with the highest activation level at the position will be selected. For example, if the centre-finding algorithm finds one object centre for *class2* and one for *class3* at position *(260, 340)* and the activations for the three classes of interest and the background at this position are *(0.27, 0.57, 0.93, 0.23)*, then the object for *class3* will be considered the detected object at this position since the activation for this class is the biggest one.

3.6 Object Detection Performance Measurement

The set of objects detected by the trained network is reported by the centre-finding procedure. To measure the performance of the network, the system compares the reported object centres and classes against the desired known object centres and reports the number of objects correctly detected. We allow a tolerance of 4 pixels in the x and y directions. For example, if the coordinates of a known object are *(21, 19)* and the coordinates of a detected object are *(22, 21)*, we still consider them the same object. In this case the matching is successful, even though there is a gap of one and two pixels in the x and y directions respectively.

We use the *detection rate* and *false alarm rate* to measure the object detection performance. It is important to note that the detection performance here refers to the performance achieved by a trained network on the detection test set, not on the detection training set.

4 Results

In this section, we first briefly describe the network training and testing results for object classification, then present the object detection results in detail.

4.1 Object Classification Results

Network training and testing results for object classification are presented in Table 2. 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 database 1 ("Artificial") with pixel statistics set I is 4-4-3 (input-hidden-output nodes), the average number of epochs used to train the network and get it converged is 243.4, and the trained network can achieve 100% accuracy on the cutouts of both the classification training set and test set. With pixel statistics set II, the best network architecture is 5-4-3 which also achieved 100% accuracy on the classification test set. For database 2 ("Easy"), we can also achieve the ideal performance for object classification, no matter which pixel statistics set was used. However, the training time (number of epochs) is much longer than for database 1. However, this is not the case for database 3 ("More difficult"). If pixel statistics set I was used, only 87.62% accuracy was obtained on the classification training set and 71.83% on the classification test set. When pixel statistics set III was used, the classification (test) accuracy was increased to 86.45%.

Image Databases	Pixel Stat Set	Network Architecture	Training Epochs	Training Accuracy	Test Accuracy
Database 1 (Artificial)	I	4-4-3	243.4	100%	100%
	II	5-4-3	246.3	100%	100%
Database 2 (Easy)	I	4-6-3	834.6	100%	100%
	II	5-6-3	1186.7	100%	100%
	III	30-6-3	1253.6	100%	100%
Database 3 (more difficult)	I	4-8-3	5634.5	87.62%	71.83%
	II	5-8-3	6737.1	90.12%	75.54%
	III	30-8-3	12654.5	95.34%	86.45%

Table 2: Object classification — Network training and testing results for the three databases.

These results suggest that for the "Artificial" and the "Easy" problems here, pixel statistics set I (mean, standard deviation, max, and min) would be sufficient for object classification. For more difficult classification problem such as database 3, more pixel statistics particularly with the local region features (Pixel statistics III) can do a better job.

4.2 Object Detection Results

This section describes the detection performance of this approach on the three detection problems. In all cases, the 15 trained networks described in Table 2 are used for object detection and the average results are presented. During the object detection procedure, different 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, the higher the detection rate and the higher the false alarm rate. Thus there is generally a trade-off between the detection rate and the corresponding false alarm rate.

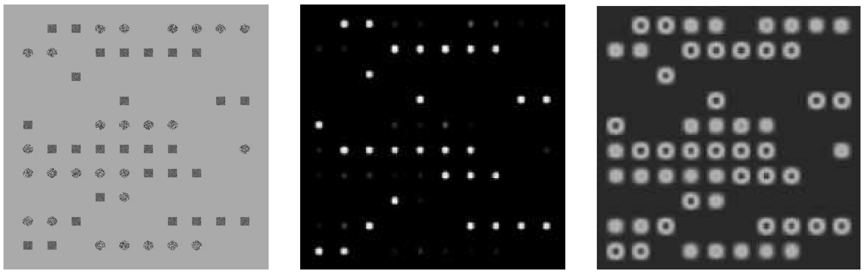
4.2.1 "Artificial" Detection Problem — Database 1

The best detection rates and the corresponding false alarm rates for the two classes of interest in database 1 based on the first two pixel statistics sets are presented in Table 3. With pixel statistics set I as inputs to neural networks, the best detection rates achieved for the two classes are 100%, showing that this approach can successfully detect all the objects of interest in this database using the first pixel statistics set. At this point, while detecting *class1* (squares) did not produce any false positives (false alarm rates are zero), detecting *class2* (circles) resulted in a 300% false alarm rate on average. This is because a small square partially in the window has a very similar mean and standard deviation as a small circle completely in the window. The invariance of the pixel statistics to the relative positions of the pixels is the problem. This situation was greatly improved when the second pixel statistics set replaced the first set as inputs to the networks, where the ideal detection result was achieved for both classes. Clearly, mean and standard deviation for an input image may be useful statistics (they are adequate for *classifying* the squares and circles), but they are not sufficient for object detection, even in a trivial, artificial data set.

"Artificial" (Database 1)		Object Classes	
		Class1	Class2
Best Detection Rate(%)		100	100
False Alarm Rate(%)	with pixel stat set I	0	300
	with pixel stat set II	0	0

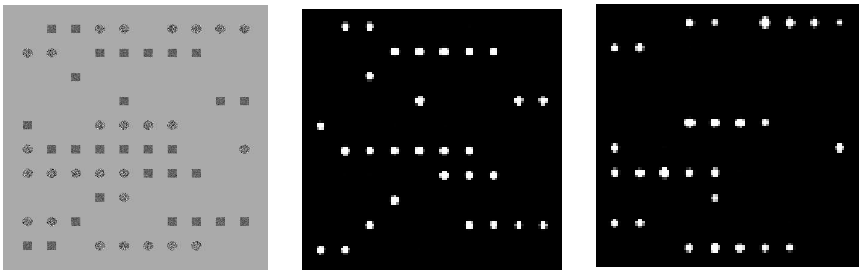
Table 3: Object detection results for the "Artificial" detection problem (database 1).

To give an intuitive view of why we obtain these results, figure 7 shows a typical example of object sweeping maps for *class1, class2* and class *other* using the two pixel statistics sets.



Original image | class1 sweeping map | class2 sweeping map

(a)



Original image | class1 sweeping map | class2 sweeping map

(b)

Figure 7: A typical example of object sweeping maps for database 1: (a) using pixel statistics set I; (b) using pixel statistics set II.

The object sweeping maps shown in figure 7 explain the detection results in table 3. The major reason that pixel statistics set I produces a lot of false positives for class2 (circles) is that none of the 4 pixel statistics (mean, sd, max, min) contain object location distribution information. Since moment contain such information, pixel statistics set 2 achieved ideal performance on this database.

4.2.2 "Easy" Detection Problem — Database 2

The detection results for the "Easy" detection problems (database 2) are described in Table 4. As in the "Artificial" images, the trained neural networks achieved 100% detection rates for all the classes, showing that this approach correctly detect all the objects of interest from different classes in this database. In each run, it was always possible to find a threshold for the network output for class *coin010* which resulted in detecting all of the objects of these classes with no false alarms. However, detecting classes *coin005* was more difficult. The average false alarm rates for this class with the use of the three pixel statistics sets were 400%, 160% and 0% respectively. This suggests that the "moment" feature can improve the detection performance achieved by the mean and standard deviation, but the also using local region pixel statistics were required to give adequate performance.

"Easy" (Database 2)		Object Classes	
		coin005	coin010
Best Detection Rate(%)		100	100
False Alarm	with pixel stat set I	400	0
	with pixel stat set II	160	0
Rate(%)	with pixel stat set III	0	0

Table 4: Object detection results for the "Easy" detection problem (database 2).

4.2.3 "More Difficult" Detection Problem — Database 3

The preliminary results on this task are shown in table 5. Like the previous two databases, the trained neural networks with pixel statistics set III as inputs achieved the best performance, but, not surprisingly, the results are not as good as on the "Easy" problem.

More difficult Database 3		Object Classes	
		head	tail
Best Detection Rate(%)		100	100
False Alarm	with pixel stat set I	500	460
	with pixel stat set II	320	340
Rate(%)	with pixel stat set III	80	120

Table 5: Object detection results for the more difficult detection problems (database 3).

4.3 Discussion

As can be seen from the detection results obtained here, different pixel statistics as inputs to neural networks resulted in very different object classification and detection performance. While showing adequate performance on some object classification tasks, pixel statistics set I performed badly on object detection problems. This demonstrates that the object detection task is more difficult than the corresponding object classification task. For all the three detection problems described here, pixel statistics set II always resulted in better performance than pixel statistics set I, showing that the "moment" feature is important for object detection. Pixel statistics set III further improved the detection performance, suggesting that local region features play an important role in object detection. As expected, the object detection results on the three image databases deteriorated as the degree of difficulty of the detection problems increased.

5 Conclusions

This paper presented a pixel statistics based approach for detecting multiple class objects in large images using multilayer feed forward neural networks. The backward propagation algorithm was used for the network training on the sub-images which had been cut out from the large images in the detection training set (classification set). 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 pictures in the detection test set. In this approach, three pixel statistics sets were used to determine the effectiveness of different pixel statistics for object classification and detection problems. The experimental results showed that this approach performed very well for detecting simple and regular objects against a relatively uniform background. It did not perform as well on the difficult detection problems with a very cluttered background. As expected, the performance degrades when the approach is applied to detection problems of increasing difficulty.

The approach has the following advantages:

- The simple image pixel statistics are used as inputs to neural networks, and accordingly complex feature extraction is avoided. These pixel statistics can be chosen from a general preexisting library.
- It is a domain independent approach and can be directly applied to multiclass object detection problems in different areas.
- Multiple class objects can be detected (classified and localised) from large pictures with a single trained neural network.

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

- This approach produced a number of false alarms on detecting difficult objects against a cluttered background, which suggests that more local region pixel statistics need to be investigated.
- 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 some false positives in the difficult detection problem. We are investigating ways of refining the trained networks based on the entire images to improve the detection performance.
- This approach needs some manual work for labelling the objects of interest for data pattern preparation. We are investigating ways of automating this process.

References

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

[2] V. Ciesielski and J. Zhu. A very reliable method for detecting bacterial growths using neural networks. In *Proceedings of the International Joint Conference on Neural Networks*, pages 62–67, Beijing, November 1992.

[3] Dick de Ridder. Shared weights neural networks in image analysis. Master's thesis, Delft University of Technology, Lorentzweg 1, 2628 CJ Delft, The Netherlands, Feb 1996.

[4] Olivier Faugeras. *Three-Dimensional Computer Vision – A Geometric Viewpoint.* The MIT Press, 1993. ISBN 0-262-06158-9.

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

[6] Y. LeCun, 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*, 1:541–551, 1989.

[7] Michael W. Roth. Survey of neural network technology for automatic target recognition. *IEEE Transactions on neural networks*, 1(1):28–43, March 1990.

[8] 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.

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

[10] 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*, 8(7/8):1029–1051, 1995.

[11] Philip Winter, Shahab Sokhansanj, Hugh C.Wood, and William Crerar. Quality assessment and grading of lentils using machine vision. In *Agricultural Institute of Canada Annual Conference*, Saskatoon, SK S7N 5A9, Canada, July 1996. Canadian Society of Agricultural Engineering. CASE paper No. 96-310.

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

[13] A. Yli-Jaaski and F. Ade. Grouping symmetrical structures for object segmentation and description. *Computer Vision and Image Understanding*, 63(3):399–417, May 1996.

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