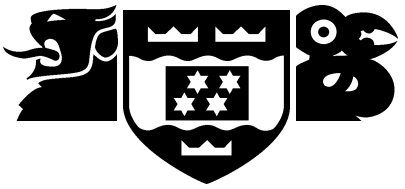
VICTORIA UNIVERSITY OF WELLINGTON
Te Whare Wananga o te Upoko o te Ika a Maui



Computer Science

PO Box 600
Wellington
New Zealand

Tel: +64 4 463 5341
Fax: +64 4 463 5045
Internet: office@mcs.vuw.ac.nz

Multi-class Object Classification and Detection Using Neural Networks

Bunna Ny, BSc, BCA

Supervisor: Dr. Mengjie Zhang

October 25, 2003

Submitted in partial fulfilment of the requirements for
Bachelor of Science with Honours in Computer Science.

Abstract

Two problems in computer vision are object classification and detection. Object classification is the determination of what category an object belongs to and object detection is the determination of where suspicious objects are in a large picture and what class they belong to. Given the advantageous of an automated recognition system, a solution to this problem has always been a desirable objective. This project investigates the application of two domain independent approaches to solve four multi-class object recognition problems ranging in difficulty. Using a pixel statistics and a raw pixel values based approach with a neural network within a recognition system, a preliminary methodology was applied. Results were promising, showing that using concentric local region pixel statistics for object classification problems, can outperform a raw pixel values based approach. A powerful new algorithm, the *Donut* algorithm, is introduced as a false alarm filter to improve the detection performance and results illustrate a markedly significant improvement when used with pixel statistics. The use of a more extensive training set is also investigated and results indicate that these can also significantly improve the performance when used with a raw pixel values based approach.

Acknowledgements

This project could not of been completed without the guidance and wisdom of my supervisor, Dr. Mengjie Zhang, his knowledge in this area is astounding and I was in awe working with him. Much accredit must also go towards Will Smart for his altruism in graciously assisting me by providing the Donut algorithm, for which this project investigates its use as a False Alarm Filter. Special thanks to Jeromé Dolman and Ben Palmer for proof reading my work as well as the rest of the Memphis kru (Simon, Urvesh, Justin, Annie, Richard and Donald) for their input, support and just being there. To my family, friends, Carl and my workmates at Plum Café, thank you. Finally to my girl, Jenna Ward, for all her love and tolerance.

Contents

List Of Figures

List Of Tables

Chapter 1

Introduction

1.1 Motivation

Everyday humans perform the trivial tasks of **object classification** and **object detection** without a second thought. Object classification is the determination of what category an object belongs to. We may perform this task in the exemplification of Figure 1.1, where it might be paying for a purchase of $2.95. From a handful of coins, the buyer must classify each coin corresponding to its value and total it to the purchase amount.

Object detection may be easily confused where it has several meanings or interpretations. In the context of this project, it is the determination of where suspicious objects are in an image. This can be considered a similar circumstance of walking along the street and identifying a $2 coin glistening in the grass, the lucky finder first detects the object and classifies it as money. Here we see the relationship between the two actions.

(a) (b)

Figure 1.1: (a) Object classification could be identifying between the different values of coins, (b) Object detection could be locating a coin in the grass

Just as object classification and detection are tasks performed by humans everyday, the ability to reproduce this within the realm of computer science would have limitless advantages in a real life scenario. Recent terrorism has illustrated security as an area in which face detection could be applied. An automated process for scanning and detecting known terrorists in an airport security image sweep would be an invaluable tool, freeing human resources to deal with other matters. The recent war in Iraq has also illustrated the critical nature of misclassification,

with the media reporting many accounts of allies being shot by friendly fire [13]. While it is important to develop automatic computer systems or programs for object classification and object detection, they remain difficult tasks in the current state of the art.

In many cases of object detection systems, all the objects of interest are considered to be part of a **single class** [21]. The task is therefore an **object** vs. **non-object** problem where the system has to distinguish objects from the background. In contrast, **multi-class object detection** refers to cases where there is more than one class of objects of interest and both their classes and locations must be determined. Generally this is a much harder task than single class detection problems and as in the context of this report, attempting to do multi-class detection using a single trained program is even more difficult.

In recent years, **neural networks** have attracted attention as effective methods for solving automatic target recognition problems [3, 11, 20, 22]. Neural networks are structures within a computer attempting to mimic the function of the human brain and in the context of object detection has three main approaches:

Raw Pixel Based: The raw pixel values of an image are directly used as inputs to neural networks [19]. This approach has the advantage of being domain independent, avoiding the hand crafting of features. The disadvantage of this approach is the high number of inputs needing to be processed. This will lead to a neural network of a large size and therefore induce longer training times and a general inefficiency on larger images.

Feature Based: Various features such as brightness, colour, size and perimeter are extracted from sub-images of the objects of interest and used as inputs to the networks [19]. The advantage of this approach is that the number of inputs that need to be processed is much less than raw pixel values and this leads to a smaller size neural network. However, this approach is domain dependent and therefore selecting and extracting good features is usually time consuming. Programs for feature extraction and selection also often need to be hand crafted.

Pixel Statistics Based: Pixel statistics is statistical information derived from the raw pixel values of an image, such as the mean and standard deviation of an image region [15]. This approach attempts to utilise the advantages from both raw pixel values and feature based approaches by being both domain independent and by using a relatively small number of inputs, accordingly removing the disadvantages of both approaches above.

This project aims to use the pixel statistics based approach for multi-class object recognition problems. It will also compare its performance against the raw pixel based approach applied to the same problems.

1.2 Goals

The intent of this project is to develop and demonstrate the usage of pixel statistics based neural networks in classifying and detecting objects in images. To achieve this, a sequence of increasingly difficult object detection problems will be applied. The results will provide the necessary information needed to confirm the effectiveness of the approach. In detail, the research questions that this project aims to answer are:

1. Which pixel statistics set is good for object classification and/or detection? [Chapter 5]
2. Can pixel statistics perform better for object classification and object detection than raw image pixels? [Chapter 5]

3. Can the false alarm rates for relatively difficult object detection be improved by a false alarm filter? If so, can a new false alarm filter improve the object detection performance? [Chapter 6]

4. Can an off-centre training method improve the object classification and object detection performance? [Chapter 7]

5. Will the object classification and detection performance deteriorate as the degree of difficulty for classification and detection problems increases?

1.3 Contributions

This project shows some interesting contributions in the field of object classification and detection in computer science.

- Concentric regions that capture the rotational invariance of an object are generally the best set of local regions in classification and detection problems. In combination with the use of an effective false alarm filter, the system can be an effective domain independent approach to recognition problems.

- A new algorithm that can be used as a false alarm filter can effectively interpret potential target locations where it is able to distinguish between true object locations and false alarms.

These contributions into the field show the potential in applying a domain independent neural network approach to object recognition problems.

1.4 Report Organisation

The remainder of this project report is organised as follows.

Chapter 2 is a literature survey introducing the structure of neural networks and how it can be applied for processing images in object classification and detection problems.

Chapter 3 introduces the databases and their detail of the images used in this project.

Chapter 4 outlines the methodology used for processing images and how this is fed as inputs into a neural network. This chapter also details how the trained network is used for classification on a test set of object cutouts and for detection on entire images.

Chapter 5 outlays the performance of a range of local regions on a very easy database. These results are used to identify well performing regions for further use on more difficult databases also described in this chapter.

Chapter 6 attempts to improve upon the derived results from Chapter 5, by implementing a more complex false alarm filter that can capture and attempt to distinguish between the conceptual objects detected.

Chapter 7 attacks the problem from the opposite angle of chapter 6, at the source of training rather than at the outcome of the results. This is by including off-centre object cutouts to produce a better representation of exemplars for training the network.

Chapter 8 concludes this project with an analysis of the findings and future work that can be carried on from this project.

Chapter 2

Literature Survey

2.1 Neural Networks

2.1.1 What is a Neural Network?

In 1911, Ramón y Cajál pioneered the idea of **neurons** (nerve cells) as structural substances of the brain [5]. Typically, neurons are five to six orders of magnitude slower than silicon logic gates, the brain makes up for this by having a staggering number of neurons with massive interconnections between them; it is estimated to be in the order of 10 billion neurons in the human brain and 60 trillion synapses or connections. The result is the brain being an amazingly efficient, highly complex, nonlinear and parallel information processing system that has the capability of organising neurons to perform certain computations (e.g. pattern recognition, perception and motor control) many times faster than the fastest super-computer in existence today.

In an attempt to mimic the human brain, we can use a neural network made up of artificial neurons as information processing units. A neural network is a massively parallel distributed processor that has a natural propensity for storing experiential knowledge and making it available for use [5]. It resembles the brain in two respects:

1. Knowledge is acquired by the network through a learning process.
2. Interneuron connection strengths known as synaptic weights are used to store the knowledge.

The procedure used to perform the learning process is called a **learning algorithm** and its purpose is to modify the synaptic weights of the network in a fashion as to achieve the desired design objective. This can be done a number of ways and in this project the **backpropagation algorithm** is used [17]. This will be detailed in section 2.1.6.

2.1.2 The Artificial Neuron

A neuron is an information processing unit fundamental to the operation of a neural network [5]. Typically, a neuron will have more than one input and an example of a neuron with R inputs is shown in Figure 2.1 [10].

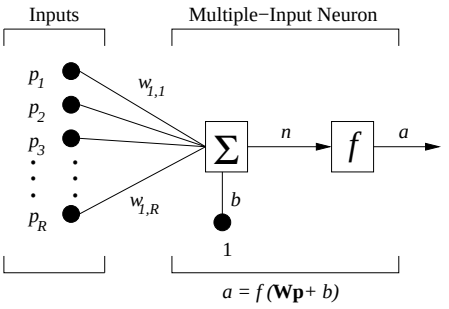


Figure 2.1: Multiple-Input Neuron

The neuron has a bias b, which is summed with the weight inputs to form the net input n:

$$n = w_{1,1}p_1 + w_{1,2}p_2 + \cdots + w_{1,R}p_R + b$$

This expression can be written in matrix form:

$$n = \mathbf{W}\mathbf{p} + b$$

where the matrix $\mathbf{W}$ for the single neuron case has only one row. Now the neuron output can be written as

$$a = f(\mathbf{W}\mathbf{p} + b)$$

where f is an activation function that computes the output value of the neuron, restricted to lie between 0 and 1.

2.1.3 The Activation Function

Various activation functions can be used to define the output of a neuron [5]. This project uses the most common form used in artificial neural networks, the log-sigmoid function. It is a strictly increasing function that exhibits smoothness and asymptotic properties, defined by [10]:

$$a = \frac{1}{1 + e^{-n}}$$

2.1.4 Network Architecture

To develop the neural network, commonly a single neuron may not be sufficient. Multiple neurons, operating in parallel can be used. This is called a "layer".

1. **Single-Layer Feed-forward Networks**

   A single-layer network of S neurons is shown in Figure 2.2 [10]. It consists of only an input layer and an output layer of neurons. Each element of the input vector $\mathbf{p}$ is connected to each neuron through the weight matrix $\mathbf{W}$. Each neuron has a bias b_i, a summer Σ, a transfer function f and an output a_i. Together, the outputs form the output vector $\mathbf{a}$. Since the input layer projects directly onto the outer layer of neurons, we have a feed-forward effect.

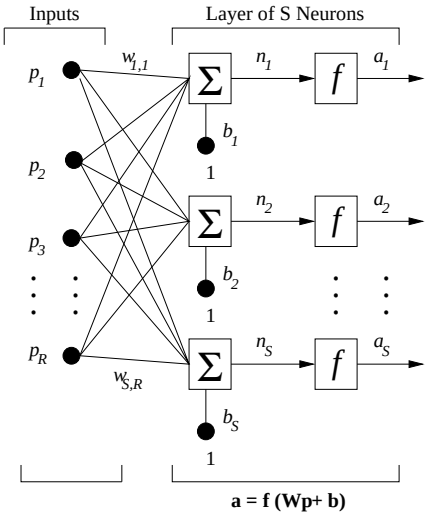


Figure 2.2: A single-layer feed forward network with a layer of S neurons

2. **Multi-layer Feed-forward Networks**

A multi-layer network has several layers, each with its own weight matrix **W**, its own bias vector **b**, a net input vector **n** and an output vector **a** [10]. As shown in Figure 2.3, there are R inputs, S^1 neurons in the first layer, S^2 neurons in the second layer etc. Different layers can have different numbers of neurons. The final layer which is the network output is called the **output layer**. The other layers are called **hidden layers**. Figure 2.3 has an output layer (layer 3) and two hidden layers (layers 1 and 2).

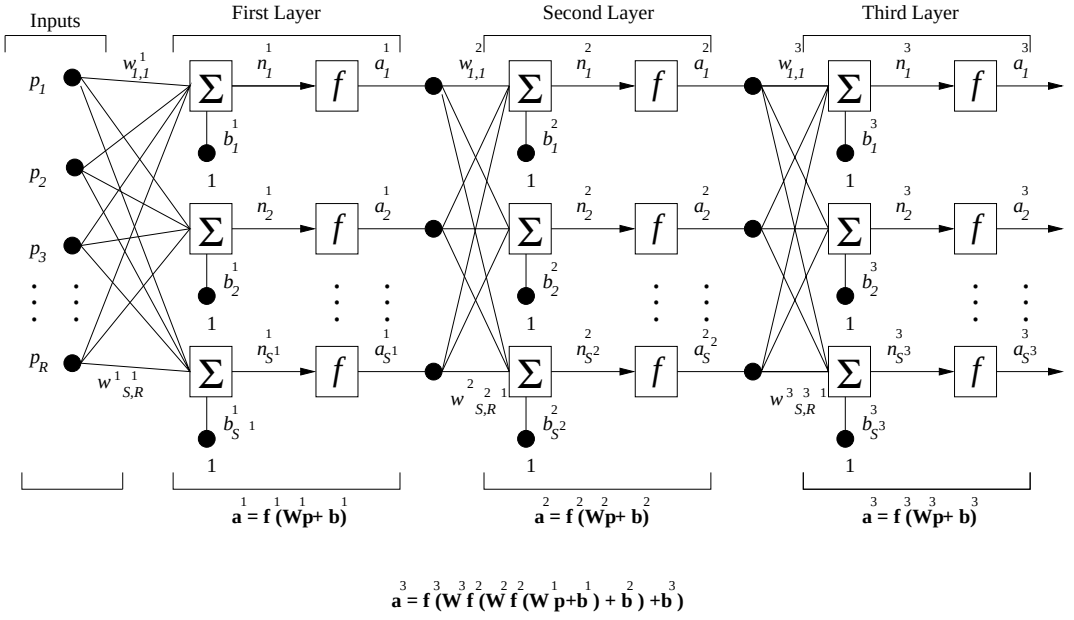


Figure 2.3: A multi-layer feed forward network with a layer of S neurons

Multi-layer networks are more powerful than single-layer networks [10]. This project aims to use trained multi-layered neural networks for multi-class object recognition problems.

3. **Other Networks**

   There are other network architectures including **Recurrent Networks** and **Lattice Structures** which are not used in this project. Details on these structures can be found in Haykin [5] or Hagan et al [10].

2.1.5 Neural Network Learning

One of the key features of a neural network is the ability of the network to learn from its environment, and to improve its performance through learning [5]. More specifically Haykin defines learning in the context of neural networks [5](p. 45) as:

> "*Learning is a process by which the free parameters of a neural network are adapted through a continuing process of stimulation by the environment in which the network is embedded. The type of learning is determined by the manner in which the parameter changes take place.*"

In the general context of neural networks, the 'free parameters' are defined to be the weights of the network, stimulated by the environment (the inputs and the target outputs of the problem set) [5]. 'The type of learning is determined by the manner in which the parameter changes take place' and in this project, this is performed through the **backpropagation** procedure which computes how much performance (error reduction) improves with a weight change.

In this procedure, the weight changes for the output layer are computed first [17]. Weight changes for the previous layer are then computed, which is considered the input to the final layer. This continues until the first layer is reached, hence the term 'Backpropagation'.

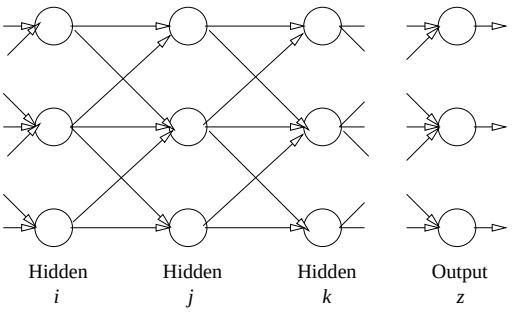


Figure 2.4: An Example of a Multi-layer Feed Forward Neural Network

From the visual representation in Figure 2.4, we can see that using backpropagation, a change in the input to node j depends on the output of node i. Therefore the weights from the hidden nodes in layer i to the hidden nodes in layer j should change substantially if the output of the hidden nodes in layer i is large.

- i.e. $\Delta w_{i \to j} \propto o_i$

We then have to consider how beneficial the change is (in terms of lower error) which we label as β. As node j is connected to the nodes in the kth layer, a change in o_j will be a benefit to each node. We can therefore deduce:

- Hidden: $\beta_j = \sum_k w_{j \to k} o_k (1 - o_k) \beta_k$
- Output: $\beta_z = d_z - o_z$

- and $\Delta w_{i \to j} \propto \beta_j$

In a processing the formulae, this produces:

$$\Delta w_{i \to j} \propto o_i o_j (1 - o_j) \beta_j$$

with the insertion of η as the constant or *learning rate* we have the **Back-propagation formulas:**

- $\Delta w_{i \to j} = \eta o_i o_j (1 - o_j) \beta_j$
- $\beta_j = \sum_k w_{j \to k} o_k (1 - o_k) \beta_k$ (Hidden Units)
- $\beta_z = d_z - o_z$ (Output Units)

2.1.6 The Backpropagation Algorithm

Until the total Mean Squared Error (MSE) is small enough [17]:

- For each input vector
  - Feed forward pass to get outputs
    $\beta_z = d_z - o_z$
  - Compute β for hidden nodes, working from the last layer to the first layer
    $\beta_j = \sum_{j \to k} o_k (1 - o_k) \beta_k$
  - Compute and store weight changes for all weights
    $\Delta w_{i \to j} = \eta o_i o_j (1 - o_j) \beta_j$
- Add up weight changes for all input vectors and change the weights.

For neural network training, the network analyses the input data and applied activations to each class that sum to 1. The higher the activation of a class, the class is considered to be more likely by the network. The largest activated class is checked against the actual class of the object and if wrong, the weights are updated in order to achieve the correct result in the next epoch (cycle). This is called supervised learning, as the network has its answers confirmed as to whether they are correct or not [10].

2.1.7 Terminating Network Training

The termination of network training is important as in the circumstance that the network performed well in classifying almost all of its training set and not as much of its testing set - the network is over-trained and has lost its generality in its final evolved state, i.e. the network has trained to much on the training set and will only be able to classify these examples. If the network has not sufficiently learnt to generalise between the different classes, this is under-training. Figure 2.5 shows where an ideal network would have its training terminated.

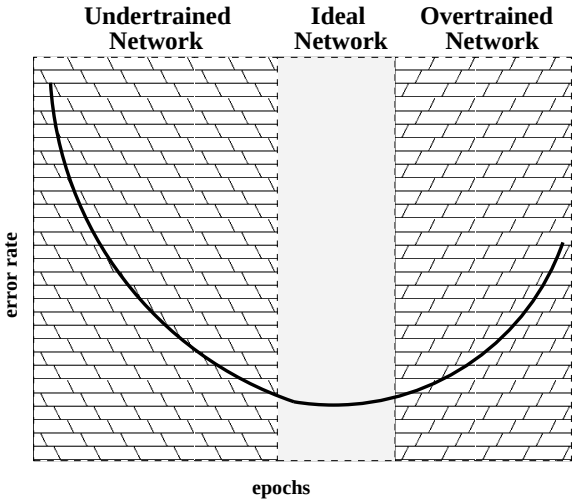


Figure 2.5: Visualisation showing the fitting of data

There are four different strategies that can be used to terminate network training. These can be used independently or in combination with each other [20]:

- **The User Control Strategy**

  The user manually terminates training whenever they feel necessary.

- **The Error Control Strategy**

  An error level is specified and used as the stopping criterion. In this project, the mean squared error (MSE) is used and whenever the network produces an error less than that specified, network training is terminated.

- **The Epoch/Cycle Control Strategy**

  A number of epochs/cycles is specified and used as the stopping criterion. Training will continue until the training epochs/cycles reaches the given number.

- **Proportion Control Strategy**

  When the proportion of the number of patterns "correctly" classified amongst the number of the total training set has reached a pre-defined percentage, training will be terminated.

2.2 Object Classification vs. Object Detection

> "*In reality, current-day machine vision systems are vastly different from, and in most ways inferior to, biological vision systems. No vertebrate animal employs such constant resolution, single snapshot imaging. Instead, vertebrates build their mental image of a scene sequentially through hundreds of twitches and tens of slow drifts of their foveated eyeballs at each of tens of saccade fixation points in the scene.*"

Quoted from Hecht-Nielsen [6](p. 449), in discussing machine vision applications, we must almost always avoid "*erroneous and misleading anthropomorphisms*" [6](p. 449). We must first understand machine vision and its potential from a totally different perspective which allows us to build a solid foundation. Only once we understand the fundamental structure are we able to use this information to contribute to the process of understanding and replicating vertebrate vision.

2.2.1 Object Classification

Object classification is an important field and a mistake can be tragic. When newspapers report headlines such as "Friendly fire is all too common" [13], where in the context of the 2003 war on Iraq, British fighter jets were being shot down by their American allies, we can see the devastating consequences of misclassification. In the 1991 Gulf War (Iraq), American fire killed more British allied troops than the Iraqis did.

The Oxford English Dictionary defines the word *classification* as [1]:

> "*1. The action of classifying or arranging in classes, according to common characteristics or affinities; assignment to the proper class.*
>
> *2. The result of classifying; a systematic distribution, allocation, or arrangement, in a class or classes; esp. of things which form the subject-matter of a science or of a methodic inquiry.*"

In the context of this project, we can derive object classification to be the allocation of given objects into their corresponding classes. An example of the success with a neural network classification system is in a real-time, on-line apple product inspection line [6]. Apple inspection involved measurement of four properties for each apple: colour grade (overall colour), discolouration, the presence of defects, and size. These attributes are then combined to determine an overall grade for the apple.

In early experimental testing of the completed system, its performance was compared with that of human apple inspectors in Japan. It was found that the performance of the machine is roughly comparable to that of a human apple inspector in terms of the technical grade coloration and defect decisions made on a large varied lot of apples. The sufficient speed and accuracy enable the machine to replace a human inspector, freeing up human resources with a tireless and efficient machine.

2.2.2 Object Detection

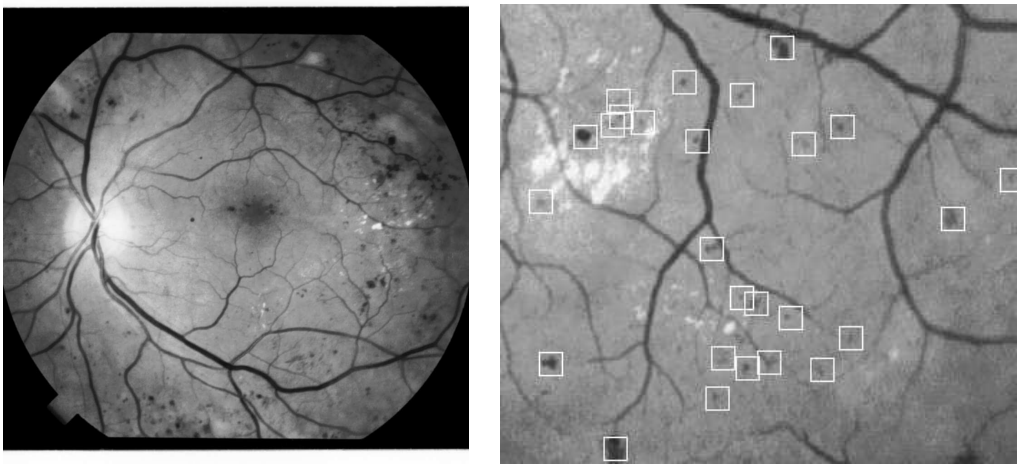
Object detection can derived in many different ways. In the context of this project and as quoted from the Oxford English Dictionary, the definition of detection is [1]:

> "...
>
> *2. a. Discovery (of what is unknown or hidden); finding out. Obs. exc. as in b.*
>
> *b. spec. The finding out of what tends to elude notice, whether on account of the particular form or condition in which it is naturally present, or because it is artfully concealed; as crime, tricks, errors, slight symptoms of disease, traces of a substance, hidden causes, etc.*
>
> ..."

Specifically, given a large image, it is desirable to determine the positions of objects of interest. This is called object localisation and in taking a further step in analysis, we can use object classification to allocate the objects into predetermined classes. The amalgamation of both these tasks is called object detection [3].

Much research has been done in object detection, where of a competent computer system that can replicate the human trait could be a far more effective method. A prime example is in the medical field where finding haemorrhages and micro aneurisms in an image of a retina is extremely difficult, even for trained professionals. An example of how difficult this problem

is can be seen in Figure 2.6, the locations of haemorrhages and micro-aneurisms are labelled using white surrounding squares.



(a) (b)

Figure 2.6: (a) An example of a retina image. (b) an enlarged view of one piece of the retina image with haemorrhages and micro-aneurisms labelled using white surrounding squares.

This specific problem has been investigated by Zhang [21] and though the results are poor, the success of using neural networks on easier problems recognises that this problem is solvable and an effective and efficient system would be revolutionary.

2.3 Neural Networks for Object Classification and Detection

There have been a variety of approaches in object classification and object detection problems. Two of the main methodologies that have attributed reasonable success are:

- **Neural Networks**
- **Genetic Programming**

2.3.1 Neural Networks

Since neural networks has been recognised as an effective approach for object classification and detection problems, a great deal of investigation and application into it has been performed over the past couple of decades. Some of the more successful applications are the classification of apples in a product line [6], handwritten digit recognition for zip code processing in a postal service [4] and target detection and classification of military targets in infrared images by the U.S. Army Centre for Night Vision and Electro-Optics [6].

Two of the main approaches that have achieved reasonable success are [21]:

- **Raw Pixel Inputs** in which the raw pixel values are used directly as inputs. By using the intensity values of both luminance and chrominance components of an image, raw pixel inputs were used to detect face locations in videophone images. This managed to achieve a 100% detection rate.

- **Feature Based** in which various features such as brightness, colour, size and perimeter are extracted from the sub-images of the objects of interest and used as inputs to the networks. This approach was used to extract moving targets from a real-time video stream, classifying them into predefined categories according to image-based properties and then robustly tracking them. Once classified, targets were tracked by a combination of temporal differencing and template match with the resulting system robustly identifying targets of interest and rejecting background clutter.

Extensive research done by Mengjie Zhang [11, 19, 20, 21, 22] has identified **pixel statistics** as an alternative approach that has shown reasonable and encouraging success. Pixel statistics is a combination of the raw pixel based approach and the feature based approach. Using raw values, it computes some mathematical properties that it uses as features from local regions, such as the mean and standard deviation. This approach is investigated in this project and the methodology is described in detail in Chapter 4.

There have been many other image analysis applications that have used neural networks with some considerable success. Two case studies discussed by Hecht-Nielsen [6] in the context of using neural networks for image analysis showed considerable success.

2.3.2 Apple Sorting

A classic machine vision problem is in real-time product inspection. For products such as apples, this has never been practical as it involves complex operations such as the measurement of colour, texture and three-dimensional shape. In an inspection system built by HNC, Inc. and Sumitomo Heavy Industries and installed in Japan, this system can inspect three apples per second on a sustained basis. This involves the measurement of the overall colour, discolouration, the presence of defects and size. These attributes are then combined to determine an overall grade for the apple.

The apples ride single-file on a conveyor belt where they enter a small brightly illuminated tunnel. In the middle of this tunnel are three colour CCD television cameras that will analyse the apples from above and the sides. Determination of the grade of the apple is done by simple measurements of the edge profile of the apple. Colouration is carried out using colour spatial frequency filter techniques and a simple rule based system.

For defect classification, a coalition of specialised neural networks carry out local analysis on small square tiles of colour pixel values within an image. These mapping networks are trained (through backpropagation or counterpropagation) to identify the presence or absence of each specific type of defect. The outputs of these classifiers are then used as the inputs into other neural networks which carry out regional defect classifications within larger tiles. The outputs of these classifiers and other image features (including the size and shape of the apple) are submitted to a rule-based system which makes the final defect detection and classification decisions.

In early experimental testing of the completed system, its performance was compared to that of human apple inspectors in Japan. It has sufficient speed to operate in real-time and sufficient accuracy to replace a human inspector.

2.3.3 Automatic Target Recognition

An end-to-end automatic target recognition system (ATR) system constructed and tested at HNC sponsored by DARPA's *Comparative Measurements: Neural Networks for Target Recognition* Program. The HNC ATR system carries out target detection and classification for infrared images.

The imagery used in this project was simulated IR imagery produced on a terrain board by the U.S. Army Center (sic) for Night Vision and Electro-Optics. The advantage of this is that the images are both realistic and relatively inexpensive to produce and therefore significant numbers of training and testing examples were able to be supplied by this method. The ATR system developed using this simulated imagery was tested on real IR imagery and the results were essentially the same as testing on simulated imagery.

The HNC ATR system consists of three functional modules that are sequentially applied to each image:

1. Potential Target Locator: is the first stage that will identify regions in the image which are good candidates for target locations.

2. False Alarm Filter: is the second stage that screens the target locations identified by the Potential Target Locator to reduce the false alarms.

3. Target Classifier: is the final stage that analyses the target locations which survive the False Alarm Filter in order to classify the target type.

All three of these modules operate on a Gabor profile representation of an image region and these profiles consist of a number of Gabor jets which have been computed at selected points in the region. The jets consist of the coefficients generated by correlating different Gabor wavelets with the image at the selected points.

The potential target locator will identify potential target locations after comparing their Gabor profiles to the profiles of one or more filled polygons - polygons serves as the centre point of the profile and are intended to be roughly similar to human made objects. The entire image is checked by examining the region centred on each point in a grid of equally spaced points. For each region, a similarity measure is computed between the regions profile and that of each polygon. If the similarity measure for any of the polygons exceeds a pre-selected threshold, the centre of the image region is deemed to be a potential target location. Multiple potential target locations in a small area of the image are eliminated by choosing the one with the highest similarity measure.

In the second and third ATR modules, the profiles are used as inputs into a feed-forward neural network trained with back-propagation. In the false alarm filter, a Gabor profile representing the potential target location is used as a network input and the state of the output processing element is a measure of the networks confidence that it is centred on a target. A pre-selected threshold will enable a decision to be made as to whether the profile represents a target.

In the target classifier, the input to the network consists of an entire Gabor profile and thus the network contains one input processing element for each coefficient in the profile. When a profile is input into the network, the target classifier's predication is taken to be the target type corresponding to the output processing element with the highest state.

The overall performance demonstrated over 95% probability of detection with a false alarm rate of 5 for every target. The false alarm filter was able to reduce the false alarm rate from 5:1 to 1:9 while degrading the detection performance by approximately 10%. Testing the target classifier in a six class problem yielded an overall classification accuracy of 97.5%. In conclusion the use of neural networks in automatic target recognition systems showed promise in this case study.

2.3.4 Genetic Programming

Genetic Programming is an evolutionary computing process, introduced by John R. Koza in 1992 [8, 14]. It is based on Darwin's theory of natural selection in that the strongest individuals of a population will survive. Extending genetic algorithms, a search algorithm also based on the mechanics of natural selection and natural genetics, genetic programming uses tree structured programs as individuals in a population. Each program is evaluated and given a fitness level, a measure of how well the program performs in training. Genetic operators are then applied to a number of programs to produce a new generation of programs. This continues for a certain number of generations or until a user defined fitness level has been achieved.

Genetic programming is a relatively new approach that has produced reasonable results when applied to object recognition problems. Further information can be found in recent research done by Will Smart, in "Genetic Programming for Multi-Class Object Classification" [18], or by Urvesh Bhowan, in "A Domain Independent Approach to Multi-Class Object Detection using Genetic Programming" [2].

2.4 Performance Evaluation

There are countless methods in evaluating the performance of systems in object recognition. It is important that this is well defined and consistent throughout a project to be a good representation of the system.

2.4.1 Performance Evaluation in Object Classification

In any application of machine learning techniques to a classification problem, the measure of the performance of the methodology in comparison to another classification methodology involves many considerations, including understandability, training times and execution times [9]. In the work of this project, a statistical comparison between classification methodologies is outside of the scope and rather, the primary aim is to demonstrate the feasibility of the various approaches developed and tested. A simple but accurate analysis is done on the performance of classifying networks.

The primary measure of classification is the classification accuracy, which is the number of objects in the testing set correctly classified as a percentage of the total number of desired objects in the test set.

$$Accuracy = \frac{N_{classified}}{N_{total}} \times 100$$

This will give the accuracy of the network as a percentage, where $N_{classified}$ is the number of objects correctly classified and N_{total} is the number of desired objects in the data set.

The second measure used is the mean squared error. This is the expected value of the square of the error, where the error is the amount by which the estimator differs from the quantity to be estimated.

$$MSE = \frac{1}{2n} \sum_{p=1}^{n} \sum_{i=1}^{m} (t_{pi} - o_{pi})^2$$

where

- p is the index of the patterns in the data set.
- n is the total number of the patterns.

- i is the index of the object classes.
- m is the total number of object classes.
- t_{pi} is the target output of the ith class for pattern p.
- o_{pi} is the actual output of the ith class for pattern p.

2.4.2 Performance Evaluation in Object Detection

As with classification, there is also a large number of evaluation measures that can be used for object detection. This project uses two basic measures that are very effective at representing the network. They can be used for further analysis in comparing the competency of the approaches used. This project does not take time into consideration.

The **Detection Rate (DR)** is a measure indicating the number of desired objects correctly detected by the system, for a single class it is [21]:

$$DR_i = \frac{\sum_{j=1}^{n} N_{true}(i,j)}{\sum_{j=1}^{n} N_{known}(i,j)} \times 100\%$$

For the overall system it is

$$DR = \frac{\sum_{j=1}^{n} \sum_{i=1}^{m} N_{true}(i,j)}{\sum_{j=1}^{n} \sum_{i=1}^{m} N_{known}(i,j)} \times 100\%$$

where

- DR_i is the detection rate for class i.
- DR is the overall detection rate (for all classes) i.
- n is the total number of pictures in the image database.
- m is the total number of object classes of interest.
- $N_{known(i,j)}$ is the number of actual known objects for the ith class in the jth image in the database.
- $N_{true(i,j)}$ is the number of the objects correctly reported by a detection system.

If all that we cared about was finding all the object of interest and it didn't matter how many objects were misclassified, the detection rate measure would be sufficient. But to give a true indication of the performance of a system, the amount of misclassifications is as important as the misclassifications in object classification. This is measured in object detection by the **False Alarm Rate (FAR)** [21]:

For a particular class i

$$FAR_i = \frac{\sum_{j=1}^{n} N_{reported}(i,j) - \sum_{n=1}^{n} N_{true}(i,j)}{\sum_{j=1}^{n} N_{known}(i,j)} \times 100\%$$

For the overall system it is

$$FAR_i = \frac{\sum_{j=1}^{n} \sum_{i=1}^{m} N_{reported}(i,j) - \sum_{n=1}^{n} \sum_{i=1}^{m} N_{true}(i,j)}{\sum_{j=1}^{n} \sum_{i=1}^{m} N_{known}(i,j)} \times 100\%$$

where

- $N_{reported}(i, j)$ is the number of the objects reported by a detection system for class i in picture j.

An ideal detection system would be one that achieves a 100% detection rate with a 0% false alarm rate.

2.4.3 Standard ROC vs. Extended ROC

To precisely state information, it must be clearly defined and displayed in context of the information [12]. The terms "true-positive faction (TPF)" and "true-negative fraction (TNF)" are respectively synonymous with the "sensitivity" and "specificity" of data. In the context of object recognition, the TPF is the fraction of desired objects in a database that are correctly classified/detected by a classified/detector and the TNF is the fraction of non-objects in a database that are correctly classified/detected as non-objects/background. Complementing this, the "false-positive fraction (FPF)" and the "false-negative fraction (FNF)" can be used as measures where the FPF is the fraction of non-objects in a database that are incorrectly classified/detected as objects and the FNF is the fraction of objects in a database that are incorrectly classified/detected as non-objects.

The pairing of sensitivity and specificity describes performance more meaningfully than the single index "percent correct" [12], but a dilemma often arises in which one system provides higher sensitivity, but lower specificity than the other. A solution is suggested by the fact that an observer can intentionally change the confidence threshold, thereby causing sensitivity and specificity values to vary inversely and thus a smooth curve would be "swept out"[12]. Essentially this constitutes a **receiver operating characteristic** curve or a **relative operating characteristic** curve (ROC). Each ROC curve indicates the tradeoffs between sensitivity and specificity that are available from the performance of a system.

Standard ROC curves conventionally take the FPF as the x axis and the TPF as the y axis. A typical standard ROC curve can be seen in Figure 2.7

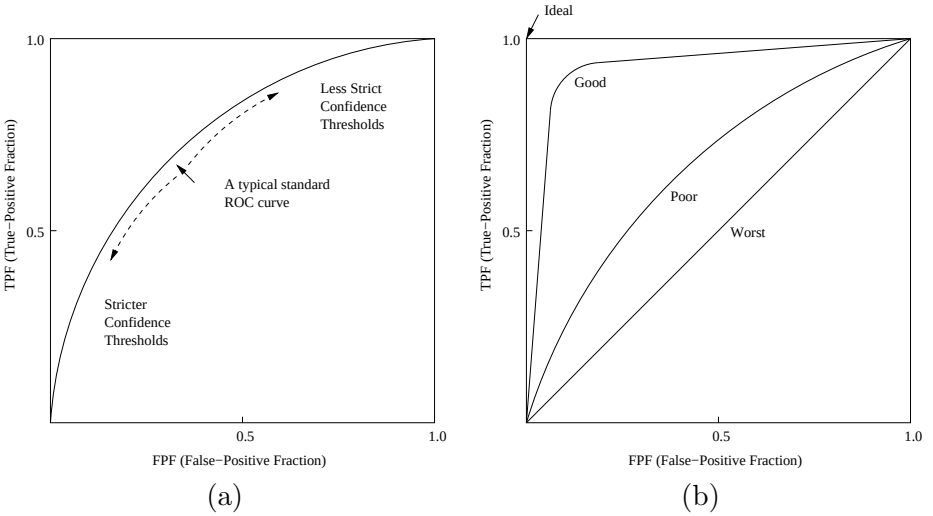


Figure 2.7: Detail of standard ROC curves

As specified in Section 2.4.2, the measures used for object detection in this project is the detection rate and false alarm rate, not the TPF and FPF used in standard ROC curves. This can be expanded to use the DR and FAR measures in what is called an extended ROC curve [3].

In extended ROC curves, the FAR forms the x axis and the DR forms the y axis. As seen in Figure 2.8, the ideal case corresponds to the top left hand corner where 100% detection rate and 0% false alarm rate is achieved. A poorer recognition system tends towards the x axis where the worst case would bea flat curve along the x axis with a 0% detection rate with any number of false alarms.

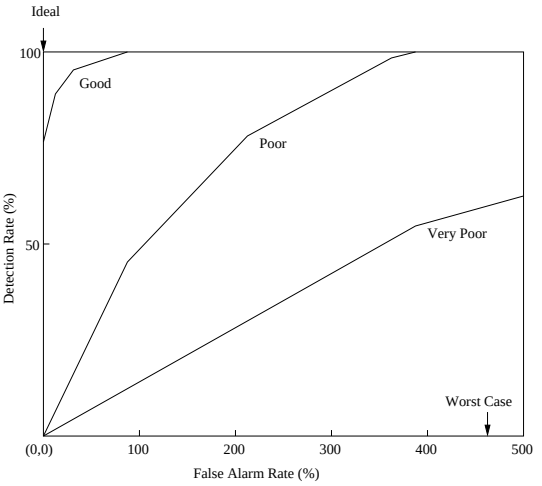


Figure 2.8: Typical Extended ROC showing ideal to worst case curves

This project uses extended ROC curves as a measure for a representation of the performance of the recognition system used. The standard ROC curve is not considered.

Chapter 3

Image Databases

This chapter will outline the details of the databases used in this project. The elimination database is considered to be a very easy task and is therefore used as a basic filter to stop ill-performing local regions from being used later for harder tasks. It is considered that if a set of local regions cannot perform well for this task, there is no point in further use as they will produce poor results later on.

3.1 The Elimination Database: 4 class problem

This database is considered trivial for object classification and detection, as each class is easily distinguishable. The entire image is artificially generated, where the noisy background is first created and then the 3 different classes of objects are placed on top. The main goal of this database is to filter local regions through their performance, where local regions that have difficulty in this task will be of little use in more complex databases. The only real difficulty is that it is quite hard to distinguish between the class of grey squares and the background. This class will prove to be the real test for the local regions.

An important note to consider is that as these objects are computer generated, the positions containing the centres of objects are also used to cut out the objects. This will mean that every object will be exactly on centre, creating a consistent set of exemplars for training. For all other databases, the centres of the objects are located manually and due to human error, not all objects will be consistently centred.

An example image and the main characteristics are presented in Figure 3.1 and Table 3.1.

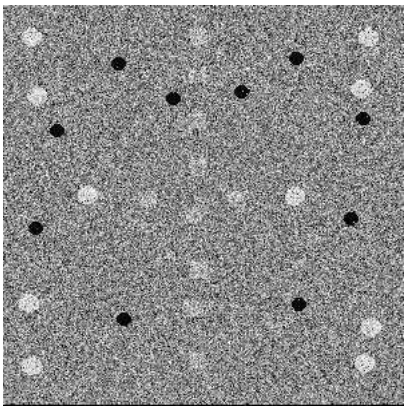


Figure 3.1: Example image from the Elimination Database

Table 3.1: Detail of the Elimination Database

Total Number of Images	24
Number of Images used for Training	10
Number of Objects used for Training	400
Number of Images used for Testing	10
Number of Objects used for Testing	400
Number of Images used for Detection	4
Total Number of Object Classes	4
Input Field Size	18×18
Total Number of Objects	800
Approximate Image Size	300×300

3.2 Database 1: 3 class problem

The images in this database consist of scannings of 16 New Zealand 5 cent coins where 8 coins have their head side showing and the remaining 8 with their tail side showing. This database is considered relatively easy for object classification and detection. It is a harder than the elimination database as it is a real scanned image of coins with a noisy background created to increase the difficulty of the task.

An example image is showing in Figure 3.2. The main characteristics of the database of images are given in Table 3.2

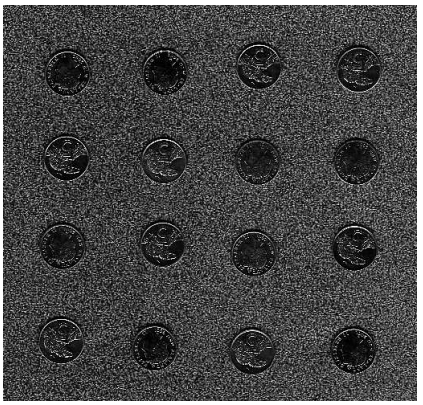


Figure 3.2: Example image from Database 1

Table 3.2: Detail of Database 1

Total Number of Images	24
Number of Images used for Training	10
Number of Objects used for Training	240
Number of Images used for Testing	10
Number of Objects used for Testing	240
Number of Images used for Detection	4
Total Number of Object Classes	3
Input Field Size	70×70
Total Number of Objects	480
Approximate Image Size	573×557

3.3 Database 2: 5 class problem

This database consists of scanned images of 16 New Zealand 5 and 10 cent coins where four 5 cent coins show heads, four 5 cent coins show tails, four 10 cent coins show heads and the final four 10 cent coins show tails. This database should be easier for distinguishing these objects from the background since the background is relatively uniform. However, the real challenge of this database lies in the handling of multiple classes. Detecting objects from the background in this database is trivial given the highly contrasting features between the coins and background. The larger number of classes in this database over database 1 and the complex nature between the classes should therefore make this database relatively difficult. An example image is shown in Figure 3.3 and the main characteristics of the database of images are given in Table 3.3.

An important point must also be considered where the size of the cutout corresponds to the ability to capture the object of interest. In the situation with multiple classes where a class of objects is larger than another, the cutout size will be set to accommodate the largest cutout. This means that a substantially larger area of background will be in the cutout of the smaller object and this may detract away from being able to capture as much information about the object.

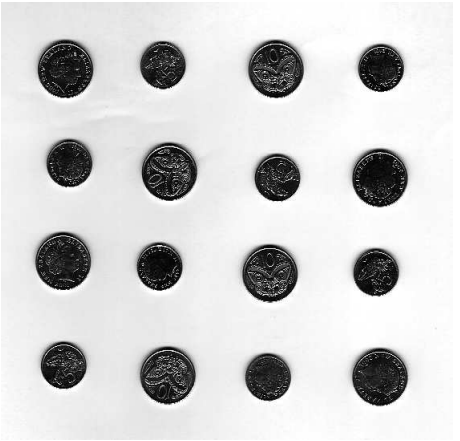


Figure 3.3: Example image from Database 3

Table 3.3: Detail of Database 2

Total Number of Images	24
Number of Images used for Training	10
Number of Objects used for Training	200
Number of Images used for Testing	10
Number of Objects used for Testing	200
Number of Images used for Detection	4
Total Number of Object Classes	5
Input Field Size	90×90
Total Number of Objects	400
Approximate Image Size	564×546

3.4 Database 3: Face Detection problem

This database represents a real life task of face classification and detection. This detection problem is extremely difficult given the computer vision techniques in the current state of the art. There are 10 portraits for each of the 4 people in this database, where each image is taken from a different angle. These images are 92×112 pixels, but as current algorithms are based on images of an even square size, preprocessing was performed to modify the images to a size 112×112 pixels, an example can be seen in Figure 3.4. For the detection process, 5 images from each class were manually placed onto a uniform black background to create a detection image for sweeping, as shown in Figure 3.5. The characteristics of this database is in Table 3.4.

While most face detection approaches use very domain specific high-level features, a goal of this project is to investigate whether domain independent pixel statistics can do a reasonable job or not.

(a) (b)

Figure 3.4: Example from Database 6, original size (a) and preprocessed size (b)

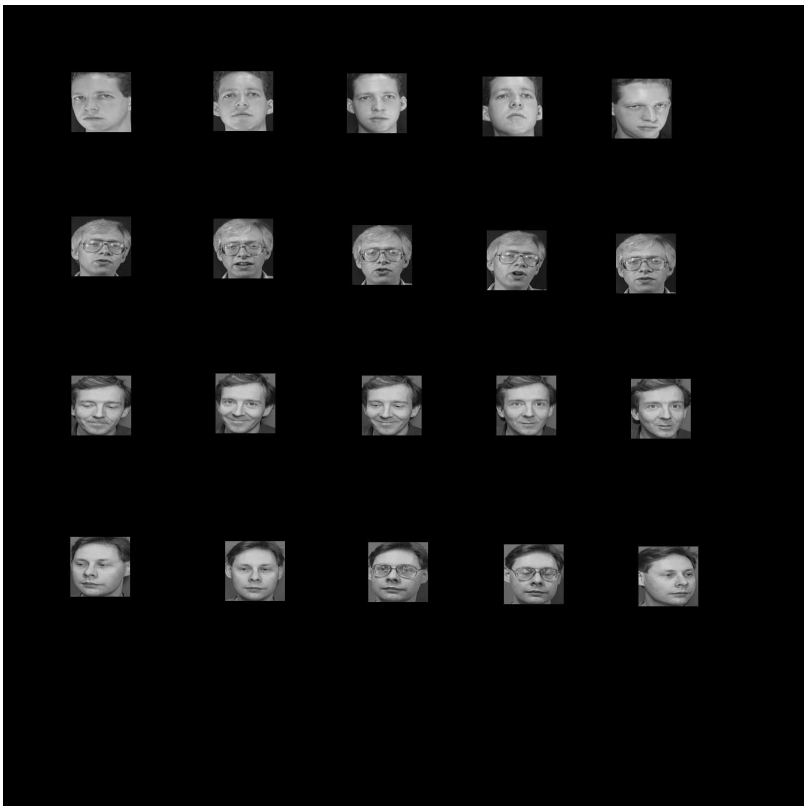


Figure 3.5: Face Detection Task for Database 6

Table 3.4: Detail of Database 3

Total Number of Images	51
Number of Images used for Training	50
Number of Objects used for Training	50
Number of Images used for Testing	50
Number of Objects used for Testing	50
Number of Images used for Detection	1
Total Number of Object Classes	5
Input Field Size	112×112
Total Number of Objects	40
Approximate Image Size	1500×1500

Chapter 4

Preliminary Methodology

4.1 Overview

Figure 4.1 shows the typical structure of using neural networks for object detection. It is a 5-phase process consisting of preparing cutouts of objects for network training and classification, using the trained network for object detection and applying a false alarm filter to the potential target locations.

This chapter will describe each phase in more detail, as well as explaining the differences between pixel statistics and raw pixel values and where they are used.

Figure 3.1 shows an overview to the approach that is used for the object detection process. First, given a database containing objects of multiple classes against a background, we need to first segregate the objects into a classes designated for network training, network testing and for object detection (network sweeping). Generally in this project, this was done in the percentage of 40:40:20 from the database for training, testing and sweeping respectively.

4.2 Phase 1: Allocate Cutouts

4.2.1 Cutout Analysis

The initial step in this methodology is to first process the database by allocating them into appropriate sets. Given a sufficient number of images, this project allocated objects into the training, testing and sweeping sets using the ratio 40:40:20. This ensures that the objects in each set remain independent from each other and so the network does not end up detecting an object that it has already seen in training, as this would be unfair. A database without many objects can pose a problem as this would mean that there is not sufficient data to train the network. This is the situation with database 3 - faces, where there are only 10 images of each class. 10-fold cross validation was used to get around this problem and though this meant that exemplars are in the detection set, every image had an opportunity to be in the testing set. 10 fold cross validation will be explained in further detail in the Training section.

For the images designated for the purpose of network training and network testing, all the objects of interest are manually cut out and classified. Then, depending upon the method used, the cutout is analysed based upon the intensities of pixels in the cutout image. When using raw pixel values, each pixel in the cutout will have an intensity value ranging from 0 to 255. The darker intensities will tend towards 0 and the lighter intensities toward 255. Pixel statistics uses mathematical calculations from these intensities based on regions in the cutout image.

These values (raw or statistical) are stored along with an indication of the class the cutout belongs to, into a text or binary file called a pattern file. Also recorded into this file are which

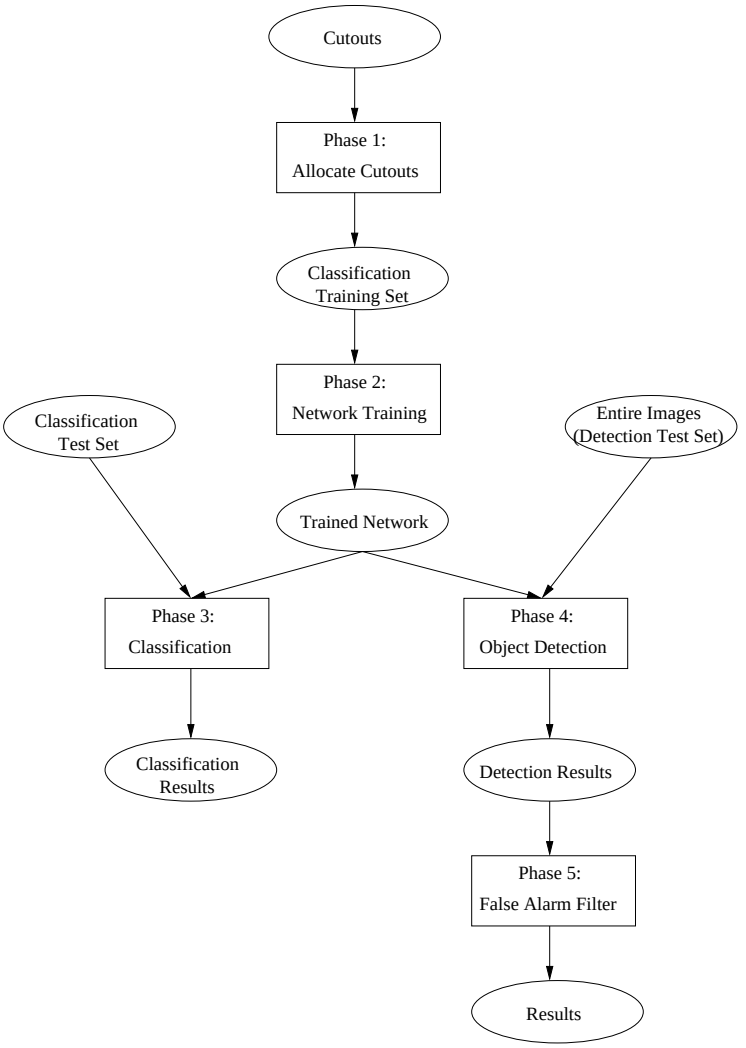


Figure 4.1: An overview of the approach

of the cutouts are to be considered for the training set and which for the testing set.

The classification training set refers to the set of cutouts split from the classification data set. This data set is used for network training in object classification and detection. The classification test set is similar to the classification training set except that it is used for measuring the object classification performance by applying the trained network during network testing. The detection test set refers to a set of entire images split from the image classification test. This is used for measuring the object detection performance through the images in this set being unseen to the training process, and are used to investigate the generalisation ability of the learning methods.

4.2.2 Pattern Files

For the object cutouts to be interpretable by the neural network, information must first be extracted for input into the neural network. This is the creation of the pattern file and will contain information on the pixel intensities of the cutout to be used as network input and the class of the object cutout to be used as network output.

4.2.3 Network Input - Raw Pixel Values

The neural network input of a cutout is based on the pixel intensity values in the cutout where the brightest (whitest) pixel will have a value of 255 and the darkest (black) pixel will have a value of 0. The simplest method is to use raw pixel values which will consider each pixel in the cutout as an input into the network.

The advantage of this method is that no cutout processing needs to be performed. The cutout is read in and the pattern file extracted directly. This makes raw pixel values domain independent. The disadvantage of this method though is that it causes the neural network to be quite large. The smallest cutout size used in this project is 18×18 and this means that the neural network will have 324 inputs. This is over 40 times larger than the number of inputs when using pixel statistics with four local regions and this may cause a considerable impact.

4.2.4 Network Input - Pixel Statistics

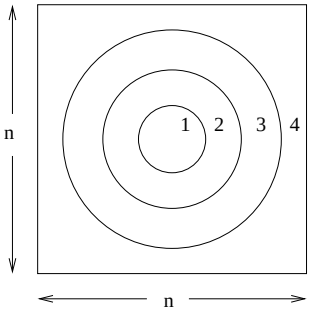
Pixel statistics are low-level, domain independent image features. They are computed from the pixel intensities in a region of the input image. Compared with domain specific features, they are not relevant to any specific shape of objects nor any specific characteristics of those objects.

There are two important aspects that must be considered in the use of pixel statistics:

- What local regions of an input image should be considered?
- What properties of these regions could be used?

In order to investigate the goals outlined in section 1.1, several different sets of local regions are applied.

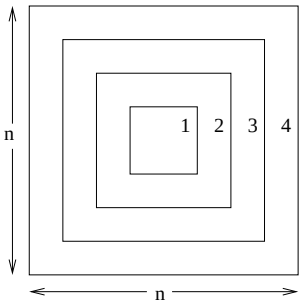
4.2.5 Pixel Statistics Local Regions - Concentric Circular Regions



Size of window = n
Number of regions = 4
Percentage covered by region:
1. 5.94%
2. 13.58%
3. 22.22%
4. 59.26%

This set of regions is based on three inner rings, designed to capture information on the object of interest, and an outer region designed to capture all other data surrounding the object. This may also work in capturing background information where a background cutout may have partial objects around the edges which will be captured by the outer region. The geometric shape of this region and the various rings should be able to capture all information on circular object of interests - especially useful with different sized coin objects.

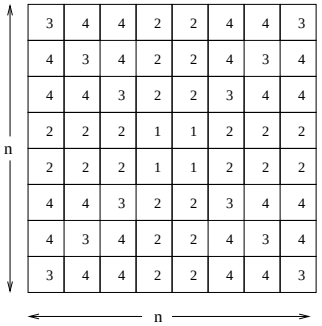
4.2.6 Pixel Statistics Local Regions - Concentric Square Regions



Size of window = n
Number of regions = 4
Percentage covered by region:
1. 4.94%
2. 14.81%
3. 24.69%
4. 55.56%

The concentric square regions is based on three inner square rings and an outer square region. This is designed in the same vein as the concentric circular regions though the geometric shape of these features will allow for objects that are not completely on centre or non circular objects to be captured more effectively.

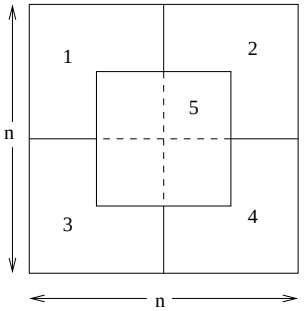
4.2.7 Pixel Statistics Local Regions - Diagonal Regions



Size of window = n (n is even)
Number of regions = 4
Percentage covered by region:
1) Central region. 1.23%
2) Horizontal/Vertical. 19.75%
3) Diagonal. 9.88%
4) Other. 69.14%

The diagonal set of regions is based on capturing information based on neighbouring pixels, rather than small areas of the cutout. This set of regions should perform best in capturing objects which take up almost all of the square (not much background) by considering outer pixels of the cutout in the same region as the inner pixels.

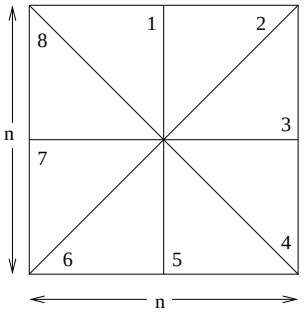
4.2.8 Pixel Statistics Local Regions - Rectilinear Regions



Size of window = n
Number of regions = 5
Percentage covered by region:
1. 25%
2. 25%
3. 25%
4. 25%
5. 25%

From the set of local regions used in this project, rectilinear regions is the only set that considers the central area of pixels twice. Regions 1-4 capture the entire cutout while region 5 is designed to concentrate on the object of interest (as this is generally in the centre of the cutout). In having the central pixels considered twice, this applies a greater amount of consideration onto the object of interest and should increases the likelihood of classification and detection.

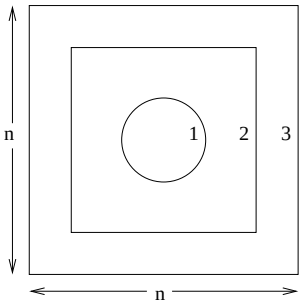
4.2.9 Pixel Statistics Local Regions - Triangular Regions



Size of window = n
Number of regions = 8
Percentage covered by region:
1. 12.5% 5. 12.5%
2. 12.5% 6. 12.5%
3. 12.5% 7. 12.5%
4. 12.5% 8. 12.5%

The triangular set of regions is designed to capture the object cutout by segregating the cutout into even size regions and consider each portion individually. This set of regions should work best in cutouts where half an object's properties will equate to another object's properties, e.g. black circle + white background = grey square. Therefore this set of regions is aimed at classifying and detecting objects where more than one object exists in a window.

4.2.10 Pixel Statistics Local Regions - Hybrid Regions



Size of window = n
Number of regions = 3
Percentage covered by region:
1. 11.11%
2. 33.33%
3. 55.56%

This hybrid set of regions was created after initial training and testing results showed that concentric square and concentric circles generally had the best performance from the set of regions created. The inner circle in this set of regions is designed to capture the object of interest, especially in the circumstance where a circular object of interest is does not take up a large amount of the cutout. The larger square input is designed to capture any discrepancies the inner region may have missed. The outer region is designed to capture the background surrounding the object of interest.

4.2.11 Pixel Statistic Properties

For each of the regions in an input field, properties need to be extracted and used as inputs into the neural network. These act as the features of a particular region and may include: maximum and minimum pixel values, mean, standard deviation, moment, median gradient, etc.

This project uses the **mean** and **standard deviation**, on the basis that these values will provide enough information to distinguish between the objects of interest in an image. Research has already investigated the effectiveness of different pixel statistics and therefore, is not investigated in this project [3, 22].

The mean and standard deviation provide the main average and contrast of the pixel intensities in the region. These can be computed from the formulae:

$$mean = \mu = \frac{\sum_{i=1}^{n} f(x_i)}{n}$$

$$standard\,deviation = \sigma = \sqrt{\frac{\sum_{i=1}^{n}(f(x_i) - \mu)^2}{n}}$$

where n is the number of pixels in that region and $f(x_i)$ is the value of the pixel at the location x_i.

These results are stored into the pattern file, along with the corresponding class for each pattern. This pattern file is used as data for the neural network system in training and testing the network, which will be explained in the Methodology chapter.

4.2.12 Input Pattern File

The general format of a pattern file is:

```
feature1 feature2 ... featureN Class
```

where the features correspond to either the features computed from pixel statistics where N is the product of the number of properties used and the number of regions, or for raw pixel values, each pixel is a feature and thus N is the size of the object squared. The 'Class' component represents an output node for each class. A '1' signifies that the feature vector represents the class of that particular output node.

(a) (b) (c)

Figure 4.2: Examples of (a) a Black Circle cutout (b) a Grey Square cutout (c) a White Circle cutout

Table 4.1 shows some examples of the different patterns extracted from Figure 4.2 for some pixel statistic regions and raw pixel values. From analysis, the values are reflective of the region chosen to extract the patterns. In observing the patterns of the pixel statistic features, just as the values in this pattern are vastly different from each other, the results from performance are generally quite different. This stresses the notion that different local region features perform differently in object classification and object detection.

Table 4.1: Excerpt from a Pattern File from Figure 4.2

Region Type	Input Pattern of a Black Circle Object	Class
Black Circle Object		
Circular	0.034314 0.007779 0.039572 0.010130 0.350327 0.072577 0.507864 0.344901	1 0 0 0
Square	0.034314 0.007779 0.066013 0.036739 0.378186 0.100381 0.505839 0.341954	1 0 0 0
Diagonal	0.040196 0.026316 0.228002 0.021751 0.424265 0.260573 0.431670 0.047003	1 0 0 0
Rectilinear	0.400823 0.098386 0.364222 0.070575 0.408376 0.434611 0.370080 0.329317 0.102687 ...	1 0 0 0
Triangular	0.410675 0.281920 0.392941 0.128635 0.408540 0.184526 0.308824 0.000000 0.363834 ...	1 0 0 0
Hybrid	0.041068 0.014622 0.300871 0.058567 0.505839 0.348997	1 0 0 0
Raw values	0.583000 0.571569 0.145750 0.699143 0.533115 0.758337 0.780833 0.510349 0.506096 ...	1 0 0 0
Grey Square Object		
Circular	0.034314 0.032169 0.039572 0.064132 0.350327 0.732834 0.507864 1.750128	0 1 0 0
...		
White Circle Object		
Circular	0.700000 0.656250 0.630927 1.022495 0.699510 1.463275 0.576511 1.986693	0 0 1 0
...		

4.3 Phase 2: Network Training

4.3.1 Network Architecture

Before training the neural network, the network architecture and parameters need to be considered. The network architecture consists of the number of input nodes, hidden nodes and output nodes of that the network will be working on. The number of input nodes corresponds to the type of pixel statistics chosen (processing the pattern file created by the rectilinear set of regions will have 10 input nodes, the mean of region 1, the standard deviation of region 1, the mean of region 2, etc). The number of output nodes corresponds to the number of object classes in the database. In the simple example of classifying the sides of a 5 cent coin, there are three classes, the head side of the coin, the tail side of the coin and the background.

There is no real heuristic in deciding upon the number of hidden nodes. These were chosen through trial and error, where they may be adjusted depending upon the difficulty of the database and the type of features in use. In a general sense, the smaller the number of hidden nodes, the better the generalisation, but also the higher the likelihood the network is not able to find a solution. In contrast, a higher number of hidden nodes maybe able to form a more efficient solution, but this takes a longer training time and increases the chance of over training on the training data set. An example of a neural network used in database 1 is given in Figure 4.3. The set of pixel statistic regions are circular, therefore there are 8 input nodes. All networks are fully connected, therefore every node in each layer is fully connected to every node in the next layer.

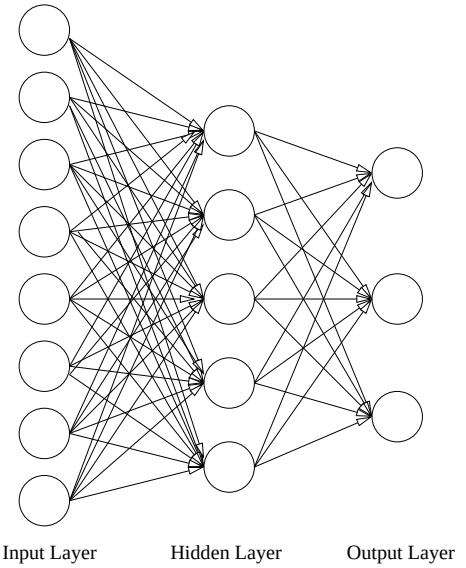


Figure 4.3: Schematic of the neural network used in database 1 by circular regions

4.3.2 Network Learning Parameters

Network parameters are required by the neural network to determine how it should learn and when it should stop.

- η is the learning rate, the constant in true gradient descent, where the larger the learning rate, the larger the changes in the weights.
- The critical error is the desired error (mean squared error) for stopping network training, i.e. if the actual error is equal or less than this error, training will cease.
- The random range is set to [-1.0, 1.0] for all problems. This is the range in where the weights are randomly initialised.
- The percent is the desired percentage of patterns correctly classified/learnt in the training set. If the actual correct percentage of training patterns is equal to or greater than this pre-defined value, training will cease.

4.3.3 Network Training

Reading in the patterns from the training set in the pattern file, each value in every pattern corresponds to an input node of a neural network and each class of interest corresponds to an output node. Through backpropagation [17], the network is trained to distinguish between the different classes given the input pattern.

This project will use a combination of the error control strategy and the proportion control strategy when in use with pixel statistics. The proportion control strategy was set to 100% and the error control strategy was determined through trial and error. When training raw pixel values based neural networks, only the error control strategy is used. Raw pixel values are generally able to converge to a 100% training accuracy rate considerably quickly in relation to the mean square error, therefore the proportion control strategy was not used or else the network would generally be under-trained.

4.3.4 10-Fold Cross Validation

Most work in this project uses the training and testing sets of equal sizes. This method is fine for databases with a sufficient number of training examples, but where there is dataset is limited as with that in Database 3, the network will be less than ideal as there will not be enough training examples for the network to sufficiently train on.

This introduces the method of 10-fold cross validation, which is implemented by dividing the entire data set into 10 equal sized sections [16]. Nine of those sections are then used for training and the excluded data is set aside so it can be used as an independent validation set for measuring the learners performance. Training is then continued with the training set designated to include the validation set and another section set to be the new validation set and so on, until all sections have had one chance to be considered as a validation set. The learners performance is then considered to be the average result in testing each of the validation sets.

Some of the dataset is also needed for object detection. For this project, in database 3, 5 from the 10 images of each class was manually pasted onto an image file containing background for the object detection process. A graphical representation of 10 fold cross validation can be seen in Figure 4.4.

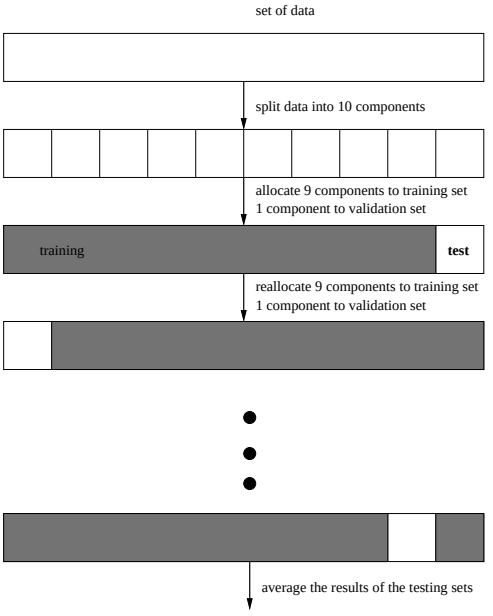


Figure 4.4: Methodology of Cross Validation

It is important to bear in mind that the cross validation method may have possible limitations where:

- It is only an estimate and may be quite noisy.
- The independence of the testing and detecting sets from the training example is lost.

10-fold cross validation should therefore best considered as an estimate and not a precise representation of the neural network.

4.4 Phase 3: Object Classification

Using the weights file as a record of the trained neural network, the patterns of objects from the test set are read by the network and an output node corresponding to the class that this network considers the pattern to belong into is activated. This is object classification and is evaluated through having the activated node checked against the true class of the object. The total of correct and incorrect classifications are used as data for evaluating the performance of the neural network for object classification.

Object classification performance is measured by the classification accuracy detail in 2.4.1. This stage is used for tuning the network architecture and learning parameters. If the classification accuracy on the (classification) test set is good, then the trained network is ready for object detection. Otherwise, the network architecture and learning parameters will be adjusted and the network retrained until the classification result is reasonably good.

4.5 Phase 4: Object Detection

4.5.1 Sweeping

In this stage, the trained neural network is applied as a template, in a moving window fashion, over a large image in order to locate objects of interest. The template is swept across and down the large picture pixel by pixel, in every possible position. This template window must match the set of pixel statistics regions or the raw pixel values which were used to train the network, so for instance if the network was trained using concentric square regions, the window swept across the image will also use the same set of concentric square regions.

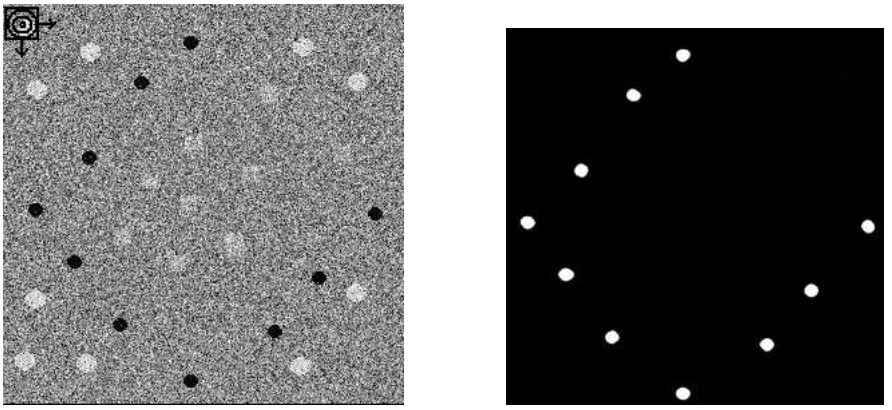


Image being swept Potential target locations of black circles

Figure 4.5: Application of Circular Regions Sweep

A PGM format image and a positions file are produced for each class of interest as well as for the background class. In these images, potential targets for each class are identified by the activation levels from the network. These activation levels correspond to the pixel intensities and thus, a position that highly corresponds to a certain class will have a very bright pixel intensity value (white). Figure 4.5 shows an example of this with a circular region sweep for black circles.

Do note that as the pixel replaced is in the centre of the moving window, the area bordering the image that is not evaluated, hence the image result being slightly smaller than the original image. The width of this area is exactly half of the moving window.

4.6 Phase 5: False Alarm Filter

Network sweeping will identify potential locations of objects of interests. Unfortunately this may contain a number of false alarms. From manual observations of resulting image sweeps, we can see a clear distinction between false positions that have been activated and activations over true objects positions. An example of this can be seen in Figure 4.6, where white pixels show activations by the network and we can see clearly, to different types of objects, filled in white circles and white rings. As the network is sweeping with an entire window, we know that these rings cannot be true, for when the sweeping window would be positioned directly over the centre of the object, it would be activated. In the case of rings, it is indicating that a window containing a portion of an object (in this case not of the class of interest) and containing background is enough for the values of the pixel to be considered as the class we are interested in.

A simplified example of this can be in the case where we may be looking for grey squares on a black background, by using the mean value of the window as a pixel statistics. In the circumstance that the image is also populated with white circles, a sweeping window that happens to contain half of a white circle and the other half black background, will average out to grey, the colour of the class we are trying to look for. This has identified a need for an algorithm that can effectively consider each object captured by the network and distinguish the true objects from the false alarms, just as we can do when looking at Figure 3.5.

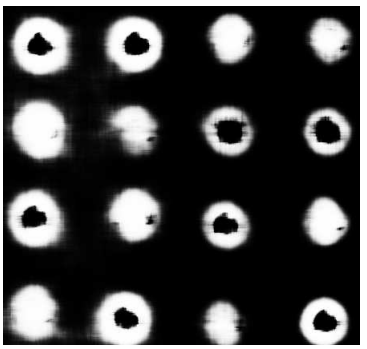


Figure 4.6: Example of a sweep for 5c Tails on a noisy background

4.6.1 The Centre Finding Algorithm

A very simple algorithm, called 'the centre Finding Algorithm', is effective at capturing true objects from false alarms, by considering that true objects found by the network, will have a greater activation than that of false alarms.

The first step is, given a threshold t between 0 and 1, only activations greater than this threshold will be considered. In terms of pixel intensities, it is considered that true objects will have a higher activation by the network than false, therefore an ideal threshold is one that is set high enough to not consider any false alarms, but low enough to capture all true objects.

The algorithm is simple in that it will consider the largest activation of the sweep as an object centre. It will then remove any surrounding activations in an area the size of the object cutout/sweeping window to stop them from being considered any further and take the next largest activation as the next object centre, continuing this process until all activations have been considered. What has been considered as object centres are then saved to file for consideration in the evaluation of the network sweep.

There are obvious limitations to this algorithm. One of them being that it considers the highest activations as the true objects. In a general case this is true, but there is often an object that the network has managed to detect, but not as strongly as other true objects and many cases some false objects. This identifies a tradeoff point, where in reducing the threshold value to a level such that it will still capture the true object will allow for some false alarms to be accepted. If we want to remove these false alarms, it will be at the cost of a 100% detection rate. What is ideal is for an algorithm to not consider the highest activation points as object centres, but one that will distinguish between the filled circles and rings as can be seen in Figure 4.6. We reserve this point and try to improve it in Chapter 6.

4.6.2 Object Detection Evaluation

Targets which survive the False Alarm Filter are considered the true locations of the objects of interest. To evaluate this performance, a manual analysis of the true centres are stored into a file and compared with the location centres returned by the network. The evaluation parameters used in this project is the Detection Rate and False Alarm Rate where an optimal performance would return 100% and 0% respectively. The formulas for computing these values for a single class is given in Table 2.4.2.

Chapter 5

Preliminary Results

5.1 Chapter Goals

The goals of this chapter are to use the outlined methodology, described in Chapter 4, to measure the performances of neural networks using different pixel statistic regions and raw pixel values, for classification and detection on databases of increasing difficulty. In the circumstance of object detection, because we are interested in achieving a perfect detection result with no false alarms, the results of the detection performance will be listed as the minimum false alarm rate from a threshold that returns the maximum detection rate. The performance of each method is evaluated through the computed average of four image sweeps.

This chapter will begin by illustrating the variation between results in using different local regions for pixel statistics and raw pixel values. On a database containing computer generated images of different shapes on a noisy background, it will use the recognition results to identify the best set of local regions for use on more difficult databases. This helps to evaluate which set of local regions are ideal for object detection as regions that struggle with such a simple task will undoubtedly not manage to handle the more difficult databases. This initial region testing will eliminate superfluous testing allowing for more focus in improving the approach.

The better performing local regions are used in object detection for more difficult databases, containing images of New Zealand 5 and 10 cent coins and faces. The first stage in the detection process is to identify regions in the image which are good candidates for target locations. The result is fed through a false alarm filter that screens the target locations to reduce the false alarms rate. Target locations that survive the False Alarm Filter are considered to be the true locations and are analysed to see if they correspond to the manually determined positions. These results can be used to evaluate the performance of the neural network. This chapter will also help to identify areas that need to be focused upon to improve results.

5.2 The Elimination Database: Artificial Shapes

This database is a simple task and is tested with a larger number of local regions. The aim is to identify poorly performing local regions and discontinue the use of them from further consideration in more difficult problem sets.

5.2.1 Neural Network Architecture

Table 5.1 displays the network architecture for local region pixel statistics and raw pixel values used in the elimination database.

Each row corresponds to the particular set of local regions used. The columns are described as follows:

- η is the learning rate of the neural network
- **CE** is the critical error used as a stopping criterion in the Error Control Strategy. If the mean squared error of the training patterns is equal to or greater than this pre-defined value, the training stops.
- **R** is the random range ([**-R**,**R**]) of the initial weights.
- % is the desired percentage of patterns correctly classified/learnt in the training set. This is used as a stopping criterion in the Proportion Control Strategy. If the actual correct percentage of the training patterns is equal to or greater than this pre-defined value, the training stops.
- **Input** specifies the number of features that are used as input nodes in this network.
- **Hidden** specifies the number of nodes in the hidden layer used in this network.
- **Output** specifies the number of object classes that are used as output nodes in this network.

Table 5.1: The neural network architecture for local region pixel statistics and raw pixel values used for the elimination database

Region	η	**CE**	**R**	**%**	**Input**	**Hidden**	**Output**
Circular	1.0	0.001	1.0	100	8	4	4
Diagonal	1.0	0.003	1.0	100	8	4	4
Square	1.0	0.001	1.0	100	8	4	4
Rectilinear	1.0	0.001	1.0	100	10	4	4
Triangular	0.5	0.001	1.0	100	16	4	4
Hybrid	0.5	0.001	1.0	100	6	3	4
Raw	1.0	0.001	1.0	101	324	4	4

5.2.2 Classification

Table 5.2: Results from object classification on the elimination database for local region pixel statistics and raw pixel values

Region Type	**Training**			**Testing**		
	Epoch	**MSE**	**Classification**	**MSE**	**Classification**	**Accuracy**
Circular	264	0.001	399/400	0.003	396/400	99%
Square	391	0.001	399/400	0.000	400/400	100%
Diagonal	399	0.003	397/400	0.017	384/400	96%
Rectilinear	414	0.001	399/400	0.022	377/400	94.25%
Triangular	702	0.002	400/400	0.022	379/400	94.75%
Hybrid	275	0.001	399/400	0.001	400/400	100%
Raw Values	52	0.001	400/400	0.003	399/400	99.75%

It is interesting to note in the classification results seen in Table 5.2, that only square and hybrid pixel statistic regions had an optimal performance. This stems from the difficulty in detecting the class of grey squares against the noisy background, where this task is difficult even for the human eye. As grey squares have the same geometric shape as contained in the local regions, they are able to capture all of the information about the square, allowing for the perfect classification result.

Circular regions, were also able to classify to a high accuracy, but this was not perfect where its misclassifications are due to the grey squares class. Disappointingly, rectilinear regions had the worst performance, which is interesting as the central region is a square and this should capture the statistics needed to classify the grey square objects. The reason for its poor performance may be due to the combination of the four other regions containing only portions of a grey square object and background. When statistics are taken from this, they may easily be confused with the background.

Surprisingly, raw pixel statistics produced an almost perfect classification result in testing. The significance of this result suggests that poorly selected pixel statistic regions are not as effective as raw pixel values.

5.2.3 Detection

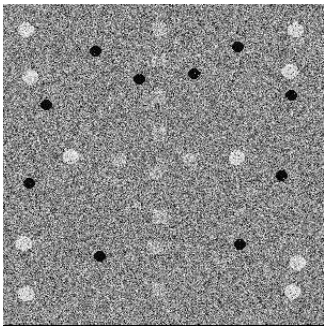


Figure 5.1: Example Image from the Elimination Database being Swept

Table 5.3: Results from object detection on the elimination database for local region pixel statistics and raw pixel values

Region Type	**Black Circles**		**Grey Squares**		**White Circles**	
	DR%	**FAR%**	**DR%**	**FAR%**	**DR%**	**FAR%**
Circular	100	0	100	0[1]	100	0
Square	100	0	100	0	100	0
Diagonal	100	0	100	487.5	100	0
Rectilinear	100	0	100	115	100	0
Triangular	100	0	100	130	100	0
Hybrid	100	0	100	7.5	100	0
Raw Values	100	0	100	180	100	0

[1]The circular regions managed to obtained a 0% false alarm rate on 3 of the images tested upon, but one of the

By performing the object detection process, the results can be seen in Table 5.3. This has revealed that pixel statistics local regions are able to perform both well and poorly for classification and/or detection. The difficulty of the grey squares proved to be an accurate measure in comparing between the different sets of regions as black and white circles appear to be trivial tasks.

Not surprisingly, concentric square regions had the optimal performance of being able to detect 100% of all objects of interest with a 0% false alarm rate. Circular regions also performed to an equally good extent and this gives notion to regions that capture the object of interest without breaking it up are generally the best performers. Rectilinear regions have somewhat reinforced this in its ability to produce a good result in comparison to the other set of regions.

Raw pixel statistics were also found to be a poor performer in the object detection of this database. This is the direct opposite of its performance in object classification.

5.2.4 Analysis

In evaluating the performance of the different types of pixel statistic regions used, it is plausible to confirm that the type of region chosen has a considerable impact on detecting objects of interest in an image. This can be seen where concentric square region pixel statistics performed almost perfectly with the similar concentric circular and hybrid regions performing very well.

There may be two reasons as to why these regions have performed so well. The first is that the geometric shape of these regions match the shapes of the objects of interest, allowing for the better capture of information. The second is that the common feature of these regions capture the rotational invariance of the object. Though these objects are not rotated explicitly, the variance in the noisy background is enough to simulate this. Rectilinear regions do have a central region which attempts to capture rotational invariance and this is recognised as having the next best performance in the next best false alarm rate. The reason why this did not perform as well may contribute to the fact that the other four regions of this type continue to break apart the object of interest.

Though hybrid regions had the third best pixel statistics performance, they were eliminated from use in later sections due to not achieving perfect results. This maybe due to it having the smallest number of regions and therefore having the most difficulty in capturing information with such a small number of features. The remaining databases are too complex for this feature.

Due to the optimal performance of every set of regions in detecting black circles and white circles, the ROC curve in Figure 5.2 represents the curve of the performance each region in detecting the object of grey squares.

image sweeps resulted in a false alarm rate of 170%. This is not an accurate representation of the performance of this set of regions and is therefore only noted and not included in the averaging of the results.

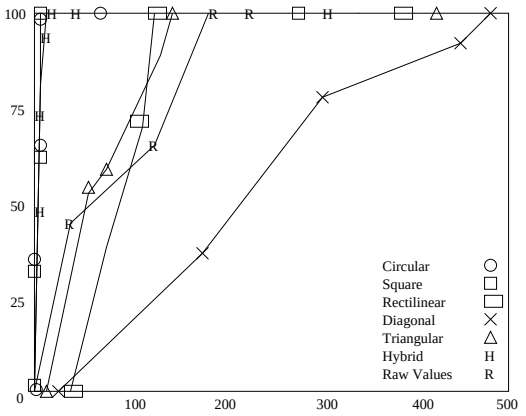


Figure 5.2: extended ROC Curve for Grey Squares Class

5.3 Database 1: 5 cent Coin Objects

This database poses a more difficult task in utilising real life objects as objects of interest.

5.3.1 Neural Network Architecture

Table 5.4: The neural network architecture for local region pixel statistics and raw pixel values used for database 1

Region	η	**CE**	**R**	**%**	**Input**	**Hidden**	**Output**
Circular	1.0	0.001	1.0	100	8	3	3
Square	1.0	0.001	1.0	100	8	3	3
Raw Pixels	1.0	0.001	1.0	101	4900	3	3

5.3.2 Classification

Table 5.5: Results from object classification on database 1 for local region pixel statistics and raw pixel values

Region Type	**Training**			**Testing**		
	Epoch	**MSE**	**Classification**	**MSE**	**Classification**	**Accuracy**
Circular	290.5	0.0027	$\frac{240}{240}$	0.0159	$\frac{233.6}{240}$	97.3%
Square	393.9	0.0029	$\frac{240}{240}$	0.0194	$\frac{237}{240}$	98.75%
Raw Pixels	91.67	0.001	$\frac{240}{240}$	0.0499	$\frac{229}{240}$	95.42%

The classification result is shown in Table 5.5, where analysis illustrates that these results have returned good performances where there was no classification accuracy under 95%. In noting that pixel statistics outperform the use of raw pixel values indicates that a simplification of data with the current data set can improve the performance of a neural network classifier.

5.3.3 Detection

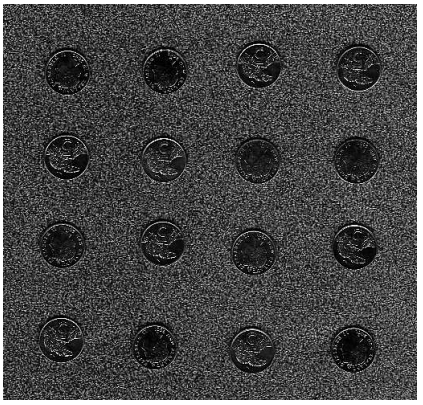


Figure 5.3: Example Image from Database 1 being Swept

Table 5.6: Results from object detection on database 1 for local region pixel statistics and raw pixel values

Database 1			**Object Classes**		
			Tails	**Heads**	**Overall**
Best Detection Rate (%)			100	100	**FAR(%)**
False Alarm Rate(%)	Circular Regions	Average	134.06 ± 18.12	9.38 ± 12.26	71.72
		Best	87.5	0	43.75
	Square Regions	Average	163.13 ± 18.98	5.63 ± 8.93	84.38
		Best	125	0	62.5
	Raw Pixels	Average	0	0	0
		Best	0	0	0

The object detection results are shown in Table 5.6 where unfortunately, these results show a poor performance of pixel statistics in comparison to perfect raw pixel values result. The reason that circular regions have outperformed square regions may be that the geometric circular shape captures information of the also circular coin objects more effectively. Note though that this difference is not large.

5.3.4 Analysis

The performance of pixel statistics for this database is not good. This is especially evident in comparison to raw pixel values which were able to obtain perfect results for this problem. This can be seen in the extended ROC curve in Figure 5.4.

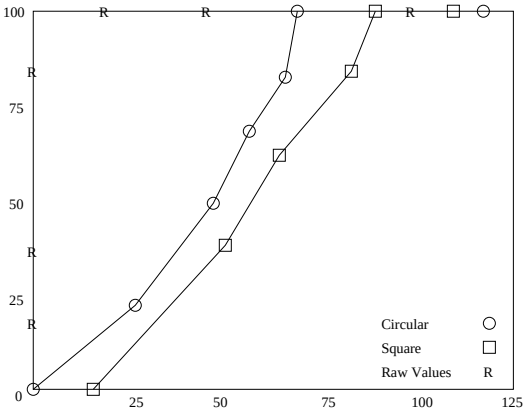


Figure 5.4: Extended ROC curve of the performance on Database 1

5.4 Database 2: 5 and 10 cent Coins

This database at first glance appears to be an easier task than that of database 1 and if this was a single class object detection problem, it would be trivial. What makes this database a more difficult problem is the larger number of classes, which should prove to be more of a difficult challenge for the neural network to learn to generalise between more classes.

5.4.1 Neural Network Architecture

Table 5.7: The neural network architecture for local region pixel statistics and raw pixel values used for database 2

Region	η	**CE**	**R**	**%**	**Input**	**Hidden**	**Output**
Circular	1.0	0.01	1.0	100	8	5	5
Square	1.0	0.01	1.0	100	8	5	5
Raw Pixels	1.0	0.001	1.0	101	8100	3	5

5.4.2 Classification

Table 5.8: Results from object classification on database 2 for local region pixel statistics and raw pixel values

Region Type	**Training**			**Testing**		
	Epoch	**MSE**	**Classification**	**MSE**	**Classification**	**Accuracy**
Circular	174.2	0.010	$\frac{194.5}{200}$	0.0136	$\frac{191}{200}$	95.5%
Square	195.8	0.010	$\frac{194}{200}$	0.0127	$\frac{191}{200}$	95.5%
Raw Pixels	295.67	0.001	$\frac{200}{200}$	0.0276	$\frac{181.33}{200}$	90.67%

As with the classification results shown in Table 5.8, pixel statistics were able to outperform raw pixel values in classification. The fact that the accuracy of all approaches is less than the

accuracy of classification in database 1 confirms that this is a more difficult database.

5.4.3 Detection

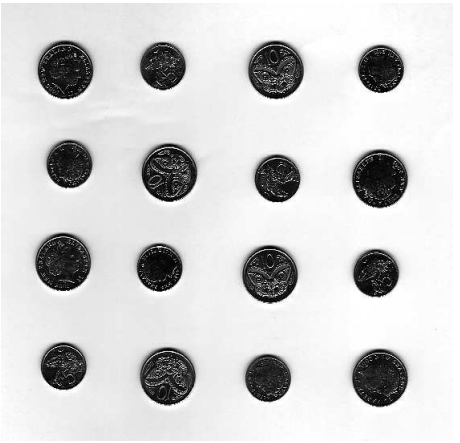


Figure 5.5: Example Image of Database 2 being Swept

Table 5.9: Results from object detection on database 2 for local region pixel statistics and raw pixel values

Database 2			**Object Classes**				
			5 Cents		**10 Cents**		**Overall**
			Tails	**Heads**	**Tails**	**Heads**	**FAR**
Best Detection Rate (%)			100	100	100	100	(%)
False Alarm Rate(%)	Circular Regions	Average	25 ± 19.61	55 ± 17.17	0	0	20
		Best	0	0	0	0	8.28
	Square Regions	Average	66.25 ± 35.60	0	0	0	16.56
		Best	0	0	0	0	7.24
	Raw Pixels	Average	61.11 ± 106.81	265.97 ± 29.97	11.11 ± 31.87	0	85.38
		Best	0	175	0	0	43.75

The results in Table 5.9 oppose the results of classification for this database and database 1, as well as the detection results of database 1. Pixel statistics were able to accurately detect and distinguish between the different coins in this database. The overall performance was also a considerable amount better than raw pixel values, indicating that pixel statistics are able to deal with a larger number of classes more effectively.

5.4.4 Analysis

This database has shown a problem which is advantageous for pixel statistics over raw pixel values. The effectiveness of the network is somewhat reflected in having a smaller neural network where pixel statistics were able to outperform raw pixel values with a false alarm rate over four times smaller. This measurement can be seen in the extended ROC curve in Figure 5.6.

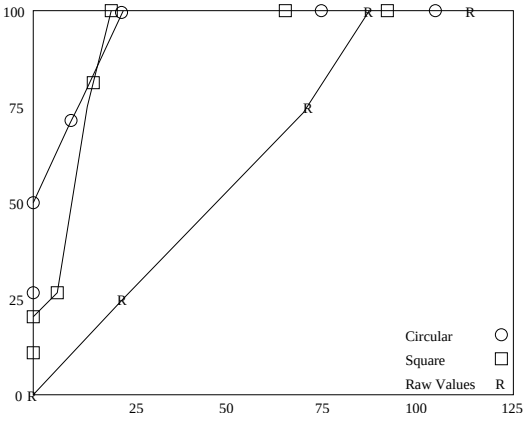


Figure 5.6: Extended ROC curve of the performance on Database 2

5.5 Database 3: Human Faces

This is the most difficult database of all and uses images of real people's faces. It is important to note that most work done so far for face detection uses high level, domain specific features and does not achieve perfect results [7]. The goal here is to investigate whether the neural networks with simple domain independent pixel statistics can do a reasonable job or not, rather than whether this approach can achieve perfect results.

5.5.1 Neural Network Architecture

Table 5.10: The neural network architecture for local region pixel statistics and raw pixel values used for database 3

Region	η	**CE**	**R**	**%**	**Input**	**Hidden**	**Output**
Circular	1.0	0.001	1.0	100	8	3	5
Square	1.0	0.001	1.0	100	8	3	5
Raw Pixels	1.0	0.01	1.0	101	12544	3	5

5.5.2 Classification

Table 5.11: Results from object classification on database 3 for local region pixel statistics and raw pixel values

Region Type	**Training**			**Testing**		
	Epoch	**MSE**	**Classification**	**MSE**	**Classification**	**Accuracy**
Circular	785.13	0.01731	$\frac{45}{45}$	0.02455	$\frac{4.79}{5}$	95.8%
Square	3233.67	0.00526	$\frac{45}{45}$	0.00612	$\frac{4.95}{5}$	99%
Raw Pixels	30.3	0.00876	$\frac{45}{45}$	0.00754	$\frac{4.99}{5}$	99.8%

Table 5.8 illustrates the results of object classification in this database of faces. Raw pixel values had an excellent classification performance only having a minor misclassification. Pixel statistics were also able to perform quite well and though this is the first time it was outperformed by raw pixel values, the extremely complex nature of the faces indicate that pixel statistics are still able to give an accurate classification performance. This is quite promising and much better than expected.

5.5.3 Detection

Table 5.12: Results from object detection on database 3 for local region pixel statistics and raw pixel values

Database 3			**Object Classes**				
			Face 1	**Face 2**	**Face 3**	**Face 4**	**Overall**
Best Detection Rate (%)			100	100	100	100	**FAR(%)**
False Alarm Rate(%)	Circular Regions	Average	20 ± 16.33	5000 ± 5177.31	555 ± 559.14	95 ± 54.16	1417.5
		Best	0	40	80	60	45
	Square Regions	Average	18 ± 11.59	9260 ± 512.64	953.33 ± 463.17	340 ± 197.994	2642
		Best	0	8720	440	200	2340
	Raw Pixels	Average	0	156 ± 137.53	106 ± 114.72	32 ± 70.05	73.5
		Best	0	0	0	0	0

The detection results are shown in Table 5.9. These illustrate the pixel statistics detection results have performed abysmally, with an overall false alarm rate that indicates pixel statistics are not able to handle the more difficult task of object detection. Though raw pixel values still had a number of false alarms, this was surprisingly slightly less than that in database 2. Given the difficulty of this task, it has illustrated that neural networks are an effective approach to difficult object detection problem.

5.5.4 Analysis

As expected, this database had extreme difficulty in detecting the objects of interest. Surprisingly, raw pixel values performed very well. Given the degree of difficulty and has a reasonably low false alarm rate, as well as a 100% detection rate. The reason for the poor performance of pixel statistics maybe due to the fact that it is extremely difficult to capture such a complex image in such a small number of features. The poor performance can be compared against raw pixel values in the extended ROC curve that can be seen in Figure 5.7.

For this very difficult task, while not good in detection, pixel statistics did very well in face classification, which suggests that neural networks with these pixel statistics can be applied to a lot of situation where only classification is needed.

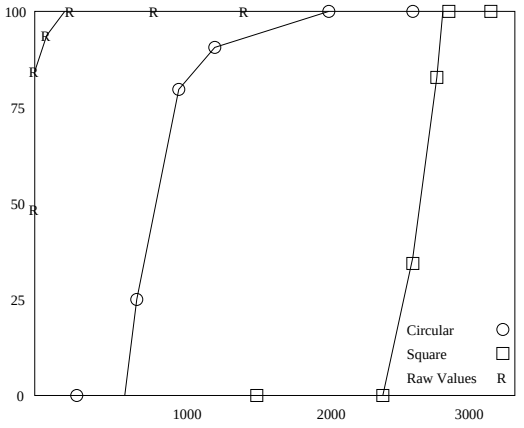


Figure 5.7: Extended ROC curve of the performance on Database 3

5.6 Chapter Summary

This chapter has identified the best set of local regions pixel statistics to be concentric circular and concentric square regions. Hybrid regions also managed to do quite well but not perfect enough to be carried on for use. These areas manage to do brilliant jobs in detecting the very easy database of artificially generated shapes on a noisy background, whereas the remaining local regions had a high false alarm rate. It is therefore considered that these local regions will not perform well on harder databases and is superfluous to continue using them.

The common theme in the well performing regions is that they capture the entire object, where all other regions break up information on the object. This is important as it captures the rotational dependency of an object. An cutout with a 5 cent coin directly on centre will have the same pattern features as another coin directly on centre but rotated to any degree (not taking into consideration the contrasts of the cutouts). This will not be the same with the other local regions, explaining their poorer performance on the first database.

The reason that hybrid regions did not also perform perfectly and kept for continued use was the lack of features that could be extracted on the object, due to the smaller number of regions. This small but significant difference will mean that less information can be captured by the pixel statistics, accounting for the decrease in performance.

In object classification, the set of local region pixel statistics used achieved a very good performance. This was not the situation for object detection where the performance was not good in more difficult databases. This proves that object detection is a more difficult task than the corresponding classification.

Raw pixel values did not really have a consistent result, with the performance being better than pixel statistics in 2 out of the 4 databases. The fact that pixel statistics performed quite well for half of the testing shows that this is a promising method. Through post-analysis, a clear distinction is evident between potential objects that are true objects and potential object that are false alarms. An example of this can be seen in Figure 5.8 where there is a clear distinction between the potential targets that are true objects of that class and the false alarms. The target class is the tail side of a 5 cent coin, try to guess the object locations that are a 5 cent coin tail side up and a 5 cent coin tail side down, the answer can be found checking the original image in Chapter 6, Figure 6.1.

Another reason that the pixel statistics did not achieve good results in detection (but achieved very good results for classifications) might be because the number and range of training

examples is not sufficient for detection - which needs to process a very large set of different kinds of background. This project will also investigate whether the object detection performance can be improved by increasing the number and diversity of the training examples of background. This will be discussed in Chapter 7.

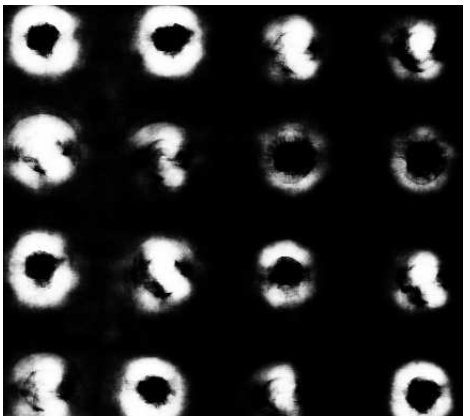


Figure 5.8: A sweep for the tails side of 5 cent coins shows a clear distinction between potential objects

This has identified that the current false alarm filter, the Centre Finding algorithm is not accurate and could be improved to reduce the false alarm rate. Currently the Centre Finding algorithm is based on the assumption that true objects will have a higher activation (brighter pixel intensity) than false alarms. Ideally an algorithm that can distinguish between the two different shapes that occur from potential locations sweeps could reduce the false alarm rate.

Chapter 6

The Donut False Alarm Filter

6.1 Chapter Goals

The goal of this chapter is aimed at reducing the false alarm rate by applying a more powerful algorithm as the false alarm filter. Where the previously used centre Finding Algorithm is simple in the fact that it was not taking a proper consideration of the potential objects, the tentatively dubbed "Donut" algorithm is designed and modified in post analysis of the resulting object detection sweeps where true objects found result in a filled circle and false alarms form rings or donuts, hence the name. Figure 6.1 shows an example where this algorithm can produce a perfect distinction between the false alarms and true locations.

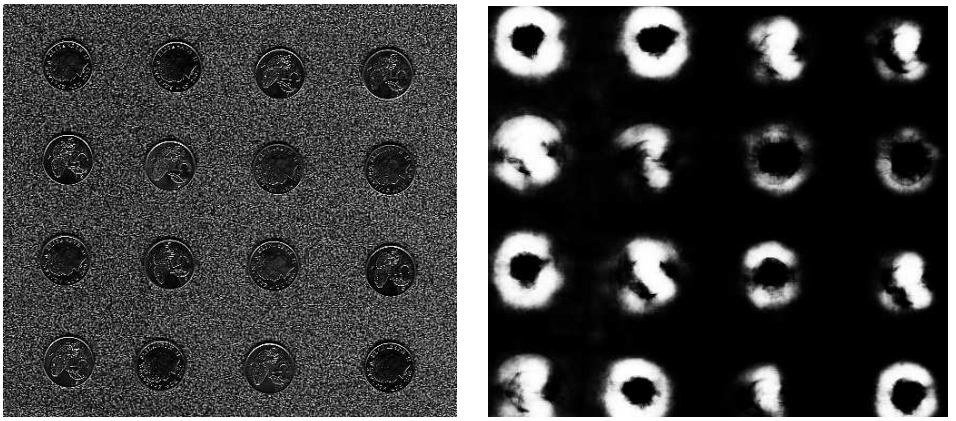


Figure 6.1: Example sweep for the tails side of 5 cent coins

6.1.1 The Donut Algorithm

This algorithm is much more effective that the centre Finding Algorithm, as it will take a greater consideration from the activations of the network's sweeping results. It also utilises a threshold by removing any weak activations, but this is the only common characteristic between the algorithms.

Two buffers are used, one for the row being processed and one for the row that has just been processed, with each value in the buffer corresponding to the activation level for each pixel from the swept image. At each position, it can be in the state of 'on', where it is above the threshold, or 'off', where it is below the threshold. If the pixel is on, it is considered part of an object and can contain either information on the object, such as the currently evaluated centre,

the number of pixels in the object and minimum and maximum positions, or a reference to the position which contains information on the same object.

The buffer is then swept from left to right, top to bottom, and at each position it is checked to see whether the pixel is on or off. When a position encountered is found to be on, it is considered that this position is part of an object and will check to see whether this is an object that has already been found or if this is a new object. This is done through checking the pixel immediately above to see whether it is on or not. If it is on and is referencing a position somewhere in the previous row, the information on this object is copied into the current position. If it is on and references to information stored somewhere in the current row, the current position will also reference itself that position. The position to the immediate left is then checked and if it is on and the position above is off, the current position will reference itself to same reference as the of the pixel to the left. If the pixel to the left is on and the pixel above is on, but they reference different objects, it indicates that what has been considered as two objects are actually one and the information is then integrated. If neither the object above or to the left are on, it is a realisation that a new object is encountered and this position will reference itself as the position holding the information on the object.

In the situation where the position is off, a scan will be done of an area, 10 by 10 pixels, centred on the current position, to see if any other positions within this range area also on. If so the position will turn itself on and perform the same functions as if it was initially on. This is an essential feature of this algorithm in it creating a small level of tolerance for discontinuities in objects, therefore an object that has been segregated into two, will be repaired into one.

This is done for all positions in the row, where once the end is reached, a check is done through the previous buffer to see if there is any information that references itself. If so, this is indication that the end of an object has been reached and the centre is computed. This is done through taking the average of the row and column positions contained in the information on the object. A small area in the centre is then checked to see whether it is activated or not and if so, it is regarded that a true object has been found, if not, the information is discarded as a false alarm.

Though this algorithm is more effective at considering potential locations than the Centre Finding algorithm, it will also have the limitation in finding it difficult to handle false alarms on the background that extended to form a "bridge" between objects. This will make the algorithm consider two objects as one. Rings that are also heavily weighted on one side will also skew the centre of the object and though it has been considered to take the average of the minimum and maximum coordinates of the activations in an object, this is more open to corruption where false alarms extending out of the object will also skew the object centre. Though this algorithm is more effective at removing the false alarms, it also has a higher probability in not being able to achieve an ideal detection rate, regardless of the false alarm rate.

As the Donut algorithm is based on improving the performance of the previously trained performance, the network architecture and classification results are not shown in this section but can be referred back to in Chapter 5.

Note that the elimination database is no longer used. In achieving an ideal classification and detection rate, this problem has been solved and is no longer needed.

6.2 Database 1: 5 cent Coin Objects

6.2.1 Detection

Table 6.1: Results from object detection on database 1 for local region pixel statistics and raw pixel values using the Donut algorithm false alarm filter

Database 1			**Object Classes**		
			Tails	**Heads**	**Overall**
Best Detection Rate (%)			100	100	**FAR(%)**
False Alarm Rate(%)	Circular Regions	Average	0	0	0
		Best	0	0	0
	Square Regions	Average	0	0	0
		Best	0	0	0
	Raw Pixels	Average	0	61 ± 138.55	30.5
		Best	0	0	0

The results using the approach with the Donut algorithm replacing the original Centre Finding algorithm are shown in Table 6.1. Both approaches of pixel statistics have managed to achieve perfect detection results, while raw pixel values had difficulty in having a number of false classifications with the head side of the 5 cent coins.

6.2.2 Analysis

This database proved the ideal testing ground in refining the false alarm filter with the Donut algorithm. In this case, the algorithm managed to interpret the result into a perfect detection rate with 0% false alarms. Unfortunately, this performance was not reflected with raw pixel values, where there was a reduction in the performance. This can be seen in Figure 6.2.

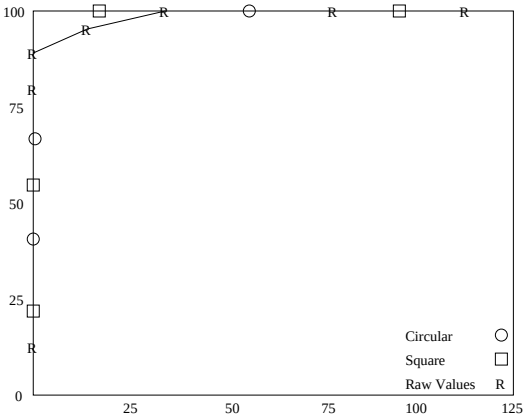


Figure 6.2: Extended ROC curve of the Donut algorithm on Database 1

6.3 Database 2: 5 and 10 cent Coin Objects

6.3.1 Detection

Table 6.2: Results from object detection on database 2 for local region pixel statistics and raw pixel values using the Donut algorithm false alarm filter

Database 2			**Object Classes**				
			5 Cents		**10 Cents**		**Overall FAR (%)**
			Tails	**Heads**	**Tails**	**Heads**	
Best Detection Rate (%)			100	100	100	100	
False Alarm Rate(%)	Circular Regions	Average	0	33.1 ± 26.2	0	0	8.3
		Best	0	0	0	0	0
	Square Regions	Average	7.7 ± 24.4	21.3 ± 23.7	0	0	7.2
		Best	0	0	0	0	0
	Raw Pixels	Average	202.8 ± 156.6	308.3 ± 34.6	16.7 ± 32.7	18.5 ± 44.1	136.3
		Best	0	275	0	0	68.75

The new results for database 2 are shown in Table 6.2. Pixel statistics managed to have a very good performance with only a small number of false alarms in detecting the head side of a 5 cent coin. Raw pixel values also had a great deal of difficulty in dealing with the classes of 5 cent coins where the best performance would still have a large false alarm rate with the head side of a 5 cent coin.

6.3.2 Analysis

Like the results from database 1, the Donut algorithm was able to reduce the false alarm rate. Though this was not perfectly reduced to 0%,the best trained neural network detectors based on both pixel statistics sets achieved ideal results. Also as in the previous database, performance of the filter was reduced where raw pixel values were used, indicating that the results of using a network based on raw pixels are too noisy for the algorithm to deal with. The result can be seen in the extended ROC curve in Figure 6.3

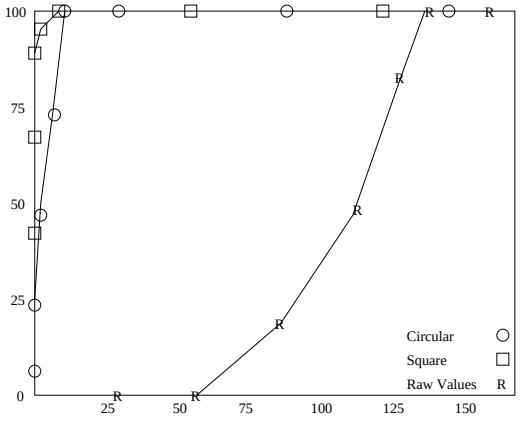


Figure 6.3: Extended ROC curve of the Donut algorithm on Database 2

6.4 Database 3: Human Faces

6.4.1 Detection

Table 6.3: Results from object detection on database 3 for local region pixel statistics and raw pixel values using the Donut algorithm false alarm filter

Database 3			**Object Classes**				
			Face 1	**Face 2**	**Face 3**	**Face 4**	**Overall**
Best Detection Rate (%)			100	100	100	100	**FAR(%)**
False Alarm Rate(%)	Circular Regions	Average	8 ± 10.95	$\frac{60 \pm 54.77}{388 \pm 554.54}$	205 ± 55.08	120 ± 23.09	$\frac{90}{180}$
		Best	0	40	140	80	60
	Square Regions	Average	18 ± 11.59	$\frac{56 \pm 35.78}{116 \pm 102.37}$	240 ± 87.18	132 ± 100.60	$\frac{89}{140}$
		Best	0	60	140	40	60
	Raw Pixels	Average	0	172 ± 141.80	128 ± 105.49	312 ± 272.80	153
		Best	0	0	0	0	0

The new results for the face detection database is sown in Table 6.3. It is important to note that a perfect detection result was not obtainable for the second face when using pixel statistics. This is display in the table as a fraction with the best detection rate over the false alarm rate. This has also affected the overall false alarm rate and this is also displayed as a fraction, with the best detection rate over the false alarm rate.

6.4.2 Analysis

Compared with the previous two databases, the results for this database are disappointing. However, the detection performances for the two pixel statistics sets were markedly improved due to the use of the Donut algorithm.

The Donut algorithm also performed poorly when used with raw pixel values, confirming that this false alarm filter is not appropriate for use with this approach, as the results are too noisy to interpret as seen in Figure 6.4.

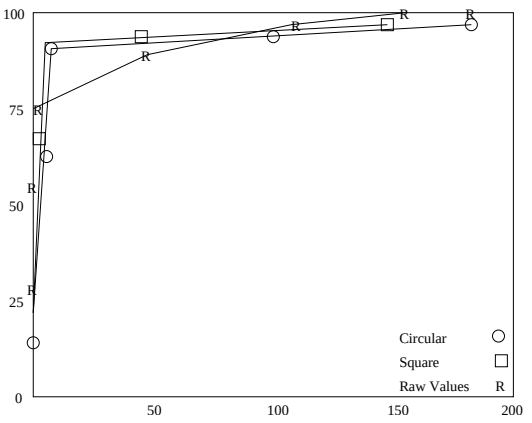


Figure 6.4: Extended ROC curve of the Donut algorithm on Database 3

6.5 Chapter Summary

The implementation of the Donut algorithm as the false alarm filter produced some promising results. For databases 1 and 2, where the Centre Finding algorithm false alarm filter produced average results with a lot of false alarms, the Donut algorithm was able to process the potential target locations to easily distinguish between false alarms and true object locations and achieved the ideal performance. Database 3 was an extremely difficult task for the networks to classify. The Donut algorithm did an excellent job in bringing the false alarm rate when using the Centre Finding algorithm down from a unacceptable level to a reasonable rate. These results suggest that the Donut algorithm can be used as an effective false alarm filter if pixel statistics are used as inputs to neural networks for object detection.

In contrast to the performance of the Donut algorithm false alarm filter with pixel statistics, these results are contradicted with raw pixel values. For every single database, the results from the raw pixel values approach degraded from the results with the Centre Finding algorithm. This is due to neural network being too large which meant that more false alarms have occurred around the objects. Where the Centre Finding algorithm was ideal because these activations were not high enough to be considered, the Donut algorithm had difficulty. If the threshold was reduced to eliminate these activations, the structure from the activations of an object begins to corrode, meaning the algorithm could not detect them as true objects.

A major limitation was identified where a 100% detection rate was not able to be achieved. This is due to having to reduce the threshold down to such a low level that the structure of the potential target locations are destroyed. The algorithm cannot process these objects and considers them to be false alarms.

It would be very interesting to further investigate whether the reduction of the detection rate can be avoided in the Donut algorithm and whether this algorithm can be successfully applied to the neural network approach with raw pixel inputs in the future.

Chapter 7

Off-Centre Training

7.1 Chapter Goals

Chapter 6 has shown an attempt to improve the object detection performance by attempting to improve the false alarm rate filter. This may be ineffective where a network has not been able to perform an accurate conceptual detection sweep.

As the training set does not contain many of the sub-images (cutouts) that will be encountered in detection, training the neural network may be improved through the inclusion of a more descriptive range of exemplars.

This chapter will analyse the performance of neural networks trained with the inclusion of a range of off-centre object cutouts. This includes quarters and halves of objects as shown with the tail side of a 5 cent coin in Figure 7.1. These results will be evaluated with the use of the centre Finding Algorithm as the false alarm filter and be compared with the preliminary methodology.

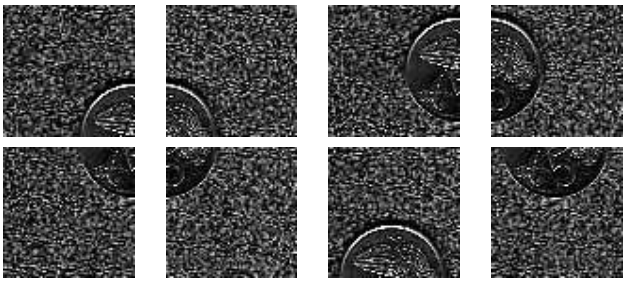


Figure 7.1: Examples of partial cutouts of the tail side of a 5 cent coin

7.2 Database 1: 5 cent Coin Objects

In this database, the four central coins are used for cutouts. These coins were selected as taking off-centre cutouts from the boundary would often go out of bounds from the image. Therefore, a quarter of the objects in this database were used to create 8 extra cutouts each.

7.2.1 Neural Network Architecture

Table 7.1: The neural network architecture for local region pixel statistics and raw pixel values trained with off-centre object exemplars used for database 1

Region	η	**CE**	**R**	**%**	**Input**	**Hidden**	**Output**
Circular	1.0	0.001	1.0	100	8	3	3
Square	1.0	0.001	1.0	100	8	3	3
Raw Values	1.0	0.001	1.0	101.0	4900	3	3

7.2.2 Classification

Table 7.2: Results from object classification on database 1 for local region pixel statistics and raw pixel values trained with off-centre object exemplars

Region Type	**Training**			**Testing**		
	Epoch	**MSE**	**Classification**	**MSE**	**Classification**	**Accuracy**
Circular	482.9	0.001	$\frac{528}{528}$	0.0063	$\frac{521.5}{528}$	98.8%
Square	860	0.001	$\frac{528}{528}$	0.0045	$\frac{521.9}{528}$	98.8%
Raw Pixels	64.4	0.001	$\frac{528}{528}$	0.0000	$\frac{528}{528}$	100%

The new results of the off-centre training approach for database 1 are shown in Table 7.2. The inclusion of training using off-centre cutouts has produced good results for pixel statistics and ideal results for raw pixel values. This shows that off-centre cutouts were helpful for the object classification of this dataset. While the classification accuracies of the circular and square regions are similar, the network with circular regions was trained and converged much fast.

7.2.3 Detection

Table 7.3: Results from object detection on database 1 for local region pixel statistics and raw pixel values trained with off-centre object exemplars

Database 1			**Object Classes**		
			Tails	**Heads**	**Overall**
Best Detection Rate (%)			100	100	**FAR(%)**
False Alarm Rate(%)	Circular Regions	Average	135.94 ± 19.24	12.19 ± 12.50	74.07%
		Best	112.5	0	56.25
	Square Regions	Average	155 ± 44.48	32 ± 16.01	93.5
		Best	25	12.5	18.75
	Raw Pixels	Average	0	0	0
		Best	0	0	0

In reflection of the perfect classification accuracy by raw pixel values, this approach has also achieved perfect results for detection, outperforming pixel statistics by a significant amount.

7.2.4 Analysis

In considering the results for the networks using pixel statistics, these results surprisingly had no resulting improvement in the performance, it actually reduced performance by having a larger false alarm rate with the current network architecture and the current set of parameters. The performance of the network with raw pixel values with the previous training achieved perfect results and therefore could not be improved upon. The results with a larger set of training examples replicated these perfect results and may show more promising results in more difficult databases. The extended ROC curve of these results can be seen in Figure 7.2.

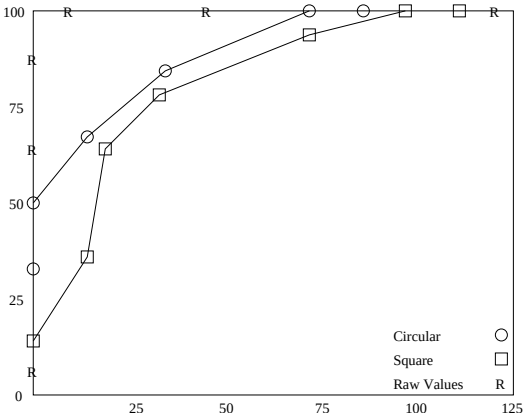


Figure 7.2: Extended ROC curve of the network trained with off-centre cutouts on Database 1

7.3 Database 2: 5 and 10 cent Coin Objects

7.3.1 Neural Network Architecture

Table 7.4: The neural network architecture for local region pixel statistics and raw pixel values trained with off-centre object exemplars used for database 2

Region	η	**CE**	**R**	**%**	**Input**	**Hidden**	**Output**
Circular	1.0	0.005	1.0	100	8	3	5
Square	1.0	0.005	1.0	100	8	3	5
Raw Values	1.0	0.005	1.0	101.0	8100	3	5

7.3.2 Classification

Table 7.5: Results from object classification on database 2 for local region pixel statistics and raw pixel values trained with off-centre object exemplars

Region Type	Training			Testing		
	Epoch	MSE	Classification	MSE	Classification	Accuracy
Circular	123	0.005	$\frac{513.7}{520}$	0.0078	$\frac{507.2}{520}$	97.5%
Square	150.4	0.005	$\frac{513.7}{520}$	0.0068	$\frac{508.9}{520}$	97.9%
Raw Pixels	163.4	0.005	$\frac{514.1}{520}$	0.0000	$\frac{520}{520}$	100%

The results shown in Table 7.5 are very similar to the results of database 1 where raw pixel values achieved a perfect classification accuracy while pixel statistics achieved very similar ones. Overall the results are excellent though and is justification for off-centre training for object classification.

7.3.3 Detection

Table 7.6: Results from object detection on database 2 for local region pixel statistics and raw pixel values trained with off-centre object exemplars

Database 2			Object Classes				
			5 Cents		10 Cents		Overall FAR (%)
			Tails	Heads	Tails	Heads	
Best Detection Rate (%)			100	100	100	100	
False Alarm Rate(%)	Circular Regions	Average	55 ± 70.98	8.13 ± 19.93	33.75 ± 106.9	0	24.22
		Best	0	0	0	0	0
	Square Regions	Average	62.5 ± 103.47	1.88 ± 6.69	1.88 ± 6.69	0	17.32
		Best	0	0	0	0	0
	Raw Pixels	Average	18.75 ± 15.09	235.42 ± 60.76	20.83 ± 45.32	0	68.75
		Best	0	125	0	0	31.25

Overall the results shown in Table 7.6 are quite good, though unfortunately the perfect results of the previous database could not be repeated. Surprisingly raw pixel values had a poor detection performance in comparison to pixel statistics, indicating that the increase in classes causes a significant problem, despite a better representation of the database in training.

7.3.4 Analysis

These results exhibit a trend from the results of database 1 where training with off-centre cutouts reduced the performance of the neural networks using pixel statistics training with on-centre cutouts. Interestingly though, where the network used raw pixel values as input, the performance showed a small but significant increase in performance by reducing the false alarm rate. This analysis can be seen in Figure 7.3.

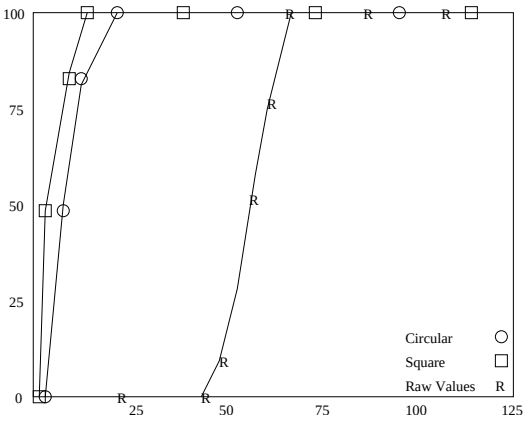


Figure 7.3: Extended ROC curve of the network trained with off-centre cutouts on Database 2

7.4 Database 3: Human Faces

7.4.1 Neural Network Architecture

Table 7.7: The neural network architecture for local region pixel statistics and raw pixel values trained with off-centre object exemplars used for database 3

Region	η	**CE**	**R**	**%**	**Input**	**Hidden**	**Output**
Circular	1.0	0.001	1.0	100	8	3	5
Square	1.0	0.001	1.0	100	8	3	5
Raw Pixels	1.0	0.01	1.0	101	12544	3	5

7.4.2 Classification

Table 7.8: Results from object classification on database 3 for local region pixel statistics and raw pixel values trained with off-centre object exemplars

Region Type	**Training**			**Testing**		
	Epoch	**MSE**	**Classification**	**MSE**	**Classification**	**Accuracy**
Circular	1852.12	0.00271	$\frac{188.99}{189}$	0.01264	$\frac{20.25}{21}$	96.43%
Square	3339.61	0.00111	$\frac{188.92}{189}$	0.00186	$\frac{20.88}{21}$	99.43%
Raw Pixels	22.81	0.00962	$\frac{186.24}{189}$	0.0000	$\frac{21}{21}$	100%

As with the previous databases, the results in Table 7.8 show that raw pixel values achieved a 100% accuracy in classification in this database, indicating that off-centre cutouts training is highly advantageous for classification with raw pixel values based neural networks. Pixel statistics also managed to achieve a very high accuracy for classification though overshadowed by the perfect performances of raw values.

7.4.3 Detection

Table 7.9: Results from object detection on database 3 for local region pixel statistics and raw pixel values trained with off-centre object exemplars

Database 3			**Object Classes**				
			Face 1	**Face 2**	**Face 3**	**Face 4**	**Overall**
Best Detection Rate (%)			100	100	100	100	**FAR(%)**
False Alarm Rate(%)	Circular Regions	Average	8.57 ± 22.68	9650 ± 1117.23	940 ± 276.27	160 ± 40	2689.64
		Best	40	8780	160	120	2275
	Square Regions	Average	220 ± 164.32	9815 ± 979.98	1384 ± 1780.30	180 ± 78.74	2899.75
		Best	0	8880	200	100	2295
	Raw Pixels	Average	0	12 ± 31.55	112 ± 77.86	36 ± 58.73	40
		Best	0	0	0	0	0

From the results shown in Table 7.9, off-centre training was no good for object detection using pixel statistics where the false alarm rate is much too high. Good results were able to be achieved by raw pixel values though where an average overall false alarm rate of 40% can be considered to be quite good.

7.4.4 Analysis

These results continue on the trend of previous databases where the difficulty of the problem was reflected in the performance. As before networks trained with pixel statistics had a slight reduction in the performance by having an unacceptably larger false alarm rate.

The better set of training examples seems to improve the performance of neural networks trained with raw pixel values as the false alarm rate was able to be reduced to almost half of that without the larger training set. A comparison of pixel statistics against raw pixel values can be seen in Figure 7.4.

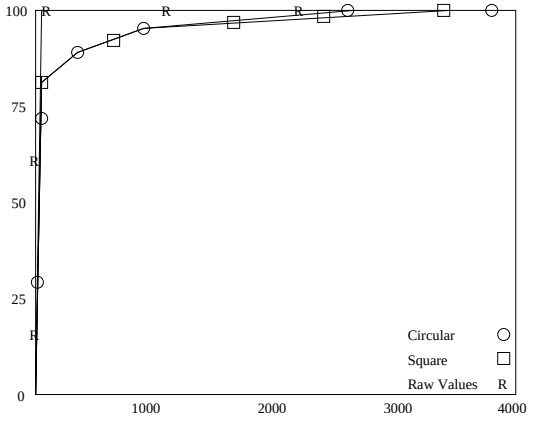


Figure 7.4: Extended ROC curve of the network trained with off-centre cutouts on Database 3

7.5 Chapter Summary

This chapter has introduced an interesting result into training the neural network. It seems logical to think that the network will perform better if given more information about the world. In evaluating the results, it appears that when used with pixel statistics, the results produced are similar, but slightly worse for object detection. This is unfortunate especially where the complexity of database 3 has caused the performance to be considerably worse than the slight changes like with the easier databases. This maybe due to the pixel statistics based neural network having difficulty when processing a portion of an object, as this can skew the statistics considerably. A coin that exhibits heads seems to contain on average, a darker number of pixel intensities than a coin exhibiting tails. Thus in training with a cutout that has half the head side of a coin in the cutout and a background that contains a considerable number of lighter pixel intensities such as those in database 1 and 3, the average can quite easily sum up to be close to the values of a pattern based on the tail side of the coin. This maybe the primary cause in the confusion of the network and thus a degradation in performance, when using off-centre training with pixel statistics.

There are two promising results that stand out. One is that the classification results of the neural networks all improved by using off-centre cutouts. This is good for a task that is aimed only at object classification, but the classification improvement of neural networks using pixel based statistics is contradicted in the poorer performance in object detection, indicating a network's performance in object classification will not necessarily be reflective of a network's performance in object detection.

Another encouraging result is the improvement shown when using off-centre training for a raw pixel value based neural network. These results give reinforcement into the fact that a raw pixel value based neural network has difficulty in dealing with rotational variance. A coin that has been rotated a mere 5° will have a completely different input pattern to the original coin. By introducing off-centre cutouts into network training, this gives the network a better representation of the database, improving its performance to consider it to be an important method to use with raw pixel values.

Chapter 8

Conclusions

This chapter first summarises the results for the object classification and detection tasks, then presents direct conclusion related to the research goals. After giving some additional findings in this project, the we briefly identify the future work directions reserved from this work.

8.1 Summary of the Results

8.1.1 Object Classification

The results of object classification are summarised in Table 8.1. The first line in this table shows that for Database 1, the circular region pixel statistics achieved 97.3% of accuracy when only on-centre training examples are used, and 98.8% accuracy when both on-centre and off-centre exemplars are used to train the network.

Table 8.1: Overall Classification Results

Database	Region Type	On Centre	Off-Centre
Database 1	Circular	97.3	98.8
	Square	98.75	98.8
	Raw Values	95.42	100
Database 2	Circular	95.5	97.5
	Square	95.5	97.5
	Raw Values	90.67	100
Database 3	Circular	95.8	96.43
	Square	99	96.43
	Raw Values	99.8	100

Note that for the preliminary database, all the methods achieved 100% classification accuracy and the results were not included in Table 8.1.

8.1.2 Object Detection

The object detection results achieved by different approaches in this project are summarised in Figures 8.1, 8.2 and 8.3 for Databases 1, 2 and 3 respectively. In these figures the legend corresponds to:

- **CFA** means the preliminary method with the Centre Finding algorithm false alarm filter and on-centre cutouts as training examples.
- **Donut** means the preliminary method with the Donut algorithm replacing the Centre Finding algorithm as the false alarm filter and on-centre cutouts as training examples.
- **Off-Centre** stands for the preliminary method plus the inclusion of off-centre cutouts as training examples.

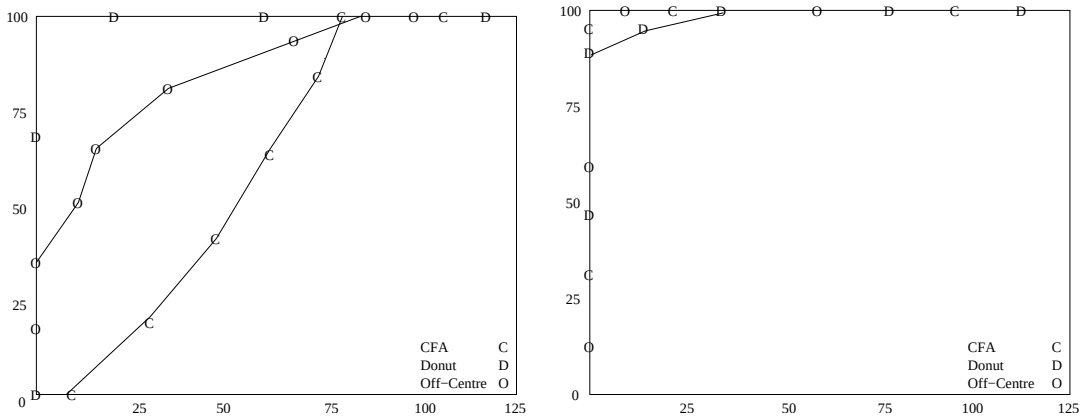


Figure 8.1: Extended ROC comparing the CFA, Donut and Off-Centre Training methods for (a) pixel statistics and (b) raw pixel values on Database 1

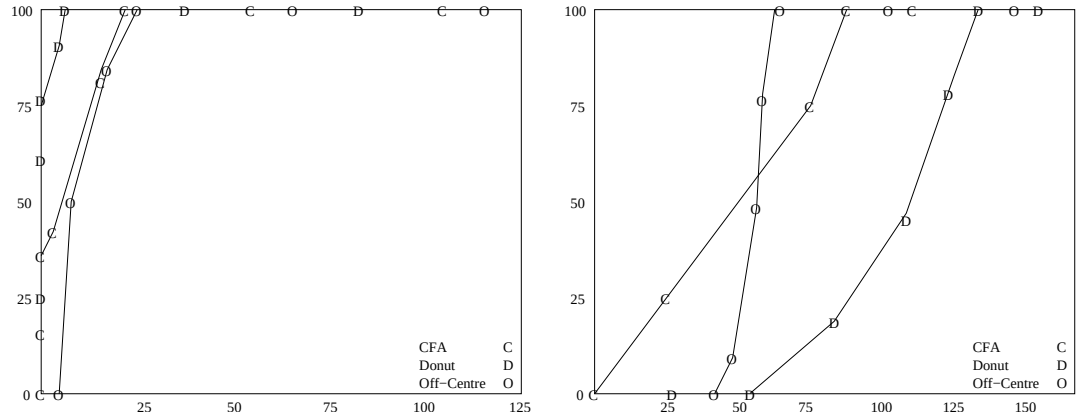


Figure 8.2: Extended ROC comparing the CFA, Donut and Off-Centre Training methods for (a) pixel statistics and (b) raw pixel values on Database 2

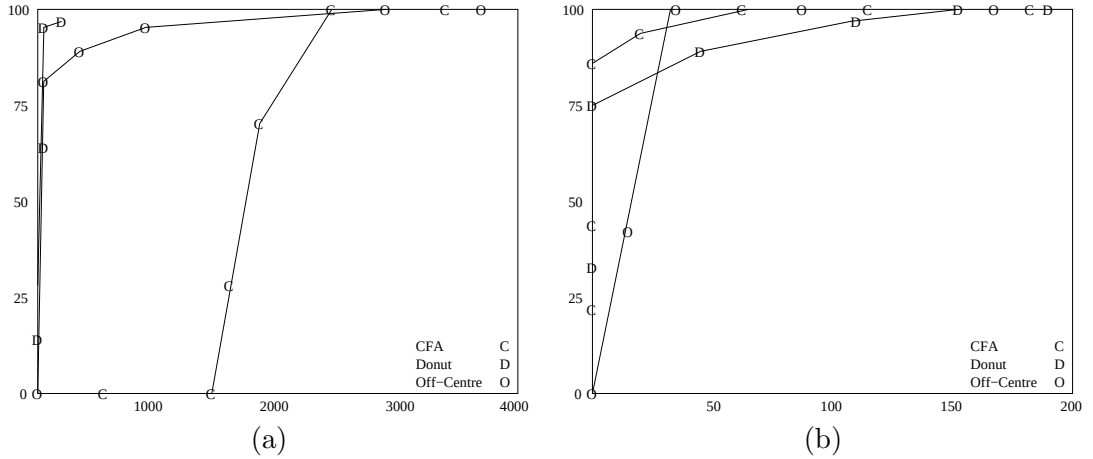


Figure 8.3: Extended ROC comparing the CFA, Donut and Off-Centre Training methods for (a) pixel statistics and (b) raw pixel values on Database 3

Also note that the Centre Finding algorithm achieved the perfect results for the elimination database, and the other two methods were not applied to this database.

8.2 Conclusions

Some significant findings were derived from the analysis of this project. In answering the research questions identified at the start of this project in Section 1.2:

"1. *Which pixel statistics set is good for object classification and/or detection?*"

The selection of the pixel statistics set of local regions is important. From one perspective, in terms of classification, the performance of different local regions slightly vary from each other, the best sets being circular, square and hybrid regions. These regions are common in the characteristics that the all capture the object of interest in their inner regions without destroying the rotational invariance.

The same result was achieved in the detection of objects in the elimination database. Circular, square and hybrid regions were all able to achieve good results. It is no surprise that out of the local regions eliminated, the best performing set of regions also have one region dedicated to capturing the rotational invariance of the object but just as with all other sets the remaining regions break up the object of interest.

In further investigation of the local region pixel statistics in the other databases, generally the difference between the circular and square sets was insignificant. This is enough to disregard the thought that the geometric shape of a region can have a considerable influence in its performance. The discrepancy can be summarised down to the same discrepancy that would occur if a similar region set was implemented. For example, a set of concentric hexagonal local regions would most likely have a result very similar to concentric squares and concentric circles, maybe in between the discrepancy of the two, as a hexagonal shape seems to be a combination of the two.

What maybe of greater significance, is the number of regions in the set. Where hybrid regions achieved a similar but poorer result to concentric circles and squares gives indication that maybe this is due to it having less features than either the squares or circles and therefore less information is captured on the cutout, as it is summarised into less numbers. This maybe consistent with the relatively poor performance of all the approaches on the most difficult

database of face detection. The objects in these images are very complex with the elements in the same class being much more different to each other than with between elements in the same coin class in database 1 or 2. Therefore it is extremely difficult to capture such a complex set of information in such a few numbers and this maybe improved through use of a more extensive set of local regions, such as 5 or 6 concentric square or circular local regions.

"2.*Can pixel statistics perform better for object classification and object detection than raw image pixels?*"

In the comparison of pixel statistics to raw pixel values for object classification, pixel statistics seems to produce a much better result when dealing with coin objects. This can be seen in the second database where raw pixel values produced the lowest accuracy for any classification result. This is most likely due to the larger number of classes, making it difficult for the network to generalise. Though this result was not reflected in database 3 where pixel statistics were able to perform well, but raw values were able to classify almost perfectly. Consideration must be taken into the difficulty of the databases where with database 2, the cutout area to accommodate 10 cent coins must be reproduced with the 5 cent coins, meaning there is a large amount of background in the picture. A lot of information is lost about the object making it harder to classify, as it must take into consideration more background.

The reason why face classification did not have such a high performance with pixel statistics as with raw pixel values may be due to the nature of the objects. It seems difficult to capture such a complex object such as faces, with only a small number of features.

This analysis of the classification results is reflected in the performance of detection, where with the coin databases, pixel statistics were able to perform much better overall than raw pixel values. In the face database, pixel statistics were not able to achieve good results with two of the approaches (preliminary and off-centre) and the Donut algorithm could not even achieve a 100% detection result. When put in comparison, the raw pixel values based neural network were able to do very well considering the extremely difficult nature of this task.

This analysis indicates that pixel statistics is an effective approach for classifying and detecting objects in large images where often, it can achieve a better performance than raw pixel values applied to same problem. This is most likely due to the smaller structured network being able to generalise more effectively. These results also identify the difficulty in capturing complex information with such a small number of statistics. An improvement for the performance of pixel statistics for face detection may lie in using a larger number of regions of statistics to assist in capturing data on the object.

"3.*Can the false alarm rates for relatively difficult object detection be improved by a false alarm filter? If so, can a new false alarm filter improve the object detection performance?*"

One of the key components of object detection is the false alarm filter. This interprets the data of conceptual object locations and recognises true object locations from false alarms by the activation levels of the neural network sweep. The Centre Finding algorithm, used as the standard approach in this project, is very basic and will consider the highest activations as true objects. The lower activations are considered to be false alarms and are removed through the use of a threshold function.

Post analysis of conceptual location results illustrate that the raw activation levels are not the best indication of true and false objects. The pattern of activations generally form a filled circle around the centre of a true object and a ring around the centre of a false alarm. An algorithm called the Donut algorithm was introduced as a false alarm filter to distinguish between the filled circles and rings (donuts), identifying the true object centres from false alarms.

In comparison of the detection results between the Centre Finding algorithm and the Donut algorithm false alarm filter for pixel statistics, every result produced a noticeable reduction in the false alarm rate. This donut false alarm filter worked perfectly for the first database in which it managed to improve results from an averaged 78.05% to 0% false alarms. An error was encountered in database 3 where 100% detection was not able to be achieved. This is not the entire fault of the false alarm filter but rather the activation levels of the conceptual results are too low for the algorithm to interpret. In taking a comparison of the false alarm rate in database 3, at the same level of detection when the Centre Finding algorithm was in use, the Donut algorithm managed to reduce this by over 20 times!

This result was conversely reflected when the Donut algorithm was applied on results obtained through a raw pixel values based neural network. For all results, there was a consistent degrade in the performance against the Centre Finding algorithm. The conceptual results of a raw pixel based neural network was much "noisier" than that of the pixel based approach where there were many specks of activations on the background amongst objects. The Donut algorithm is designed to distinguish between objects and it considers these activations as a connection between objects, therefore considering two objects as one. When the threshold is reduced to remove these bridging activations, too much data is lost about the actual objects, also making the results interpretable. This indicates that the Donut algorithm is a suitable false alarm filter when in use with pixel statistics, but in its current form, is not suitable with the results of raw pixel values based neural networks. One of the advantages of this algorithm though is that it can easily be "tweaked" to interpret the results more effectively, but this requires more extensive research and begins to suggest a loss of domain independence.

"4. *Can an off-centre training method improve the object classification and object detection performance?*"

The segregation between the tasks of object classification and object detection is in the data that they work on. Though the trained neural network will be effectively performing the same task, the test set in object classification will generally have each object on centre in the cutout. In object detection, a sweeping window is applied, taking cutouts that will contain the object in a cutout in many different positions. It therefore seems unfair that the network be expected to be able to distinguish between cutouts that it has never seen before.

This has prompted the investigation into a more in-depth training set, with the inclusion of off-centre object cutout exemplars. Effectively this will make object classification a harder task, increasing the number of examples to classify between. Surprisingly, in eight out of nine cases with the standard approach and the off-centre training method, the averaged classification performance was improved. Interestingly though with both circular and square local regions the averaged performance was the same, indicating that off-centre training can remove the indifference between similar region sets.

The detection results for a network trained with off-centre cutouts shows promise, where with all detection results for raw pixel values based neural networks, there was an improvement in the detection results, except for database 1 which already had optimal results.

The results were not concurrent with pixel statistics. In fact, in every detection result, there was a slight decrease in the performance. Due to the summarising nature of pixel statistics, this indicates that the increase in training data brought the generalisation between classes closer together. This maybe due to a cutout containing half an object of one class will be in the same class as half an object from another class.

It can be considered that the inclusion of off-centre training examples is an important method that can improve classification results when in use with pixel statistics or raw pixel values based neural networks. This approach can improve the performance of detection with a network based

on raw values but it is not recommended for inclusion when training a pixel statistics based neural network.

"5. *Will the object classification and detection performance deteriorate as the degree of difficulty for classification and detection problems increases?*"

The correlation between the considered degree of difficulty of a database and the actually detection performance of the classification and detection process are not necessarily dependent. The classification results indicate that using the preliminary methodology, both pixel statistics and raw pixel values, find it the most difficult to distinguish between the 5 and 10 cent coins in database 2. In the case of additional off-centre cutouts training, the performance of pixel statistics based neural networks found the increasing difficulty of the databases more challenging, but raw pixel values used the enhanced training set to achieve perfect results in all databases.

The detection results are interesting because the raw pixel values based neural network found it easier to distinguish between the faces in database 3 than the 5 and 10 cent coins in database 2, with database 1 being the best. Pixel statistics based neural networks on the other hand, found it easier to distinguish between the coins in database 2 than the coins in database 1. In the face detection database, it performs abysmally with a low false alarm rate for one approach without an ideal detection rate.

These results suggest that there is no direct correlation between the considered degree of difficulty of the databases, and more so, there is no real correlation between the classification results and the detection results.

8.3 Additional Findings

These results have discovered that different approaches are advantageous when used with certain networks but not with others. In specifics, this project has additionally discovered:

- *The Donut algorithm is an effective false alarm filter when used with pixel statistics based neural networks.* The Donut algorithm was initially implemented and modified to work with the results derived from the pattern conceptualised locations of pixel statistics based neural networks. It is therefore no surprise that it can perform well with the remaining databases. In the situation where it did not perform perfectly for the detection rate, this is due to the reliance on the detection from the neural network, a better training method may provide more effective results here.

- *The inclusion of off-centre cutouts exemplars can improve the classification performance of neural networks.* For classification, the cutouts were obtained through quite a controlled fashion. It is blatantly obvious when an object is on or off-centre and this inclusion of exemplars improved training to allow a better generalisation between classes.

- *The inclusion of off-centre cutouts exemplars can improve the detection performance when used to train raw pixel values based neural networks.* As with object classification the inclusion of off-centre cutouts describes a better representation of the dataset. This is especially important where the sweeping procedure will regularly encounter positions with only portions of objects. A raw pixel value based neural network captures a significant amount of information in a cutout and in having been trained with cutouts of this situation, it is able to reduce the false alarm rate in this situation.

- *The simple pixel statistics and raw pixels achieved a surprisingly good performance for face classification.* Out of the 9 results, only 1 result had an accuracy of less than 95%, and

even then it was still a good result being over 90%. This indicates that using either the pixel statistics or raw pixel values based neural networks approach, they can be applied to reasonably difficult object classification tasks with a very good accuracy.

These additional findings in combination with the conclusion from the results of this project have identified ways in which neural networks can be used in object classification efficiently and effectively. Both methods can generally be trained in a relatively small number of epochs and the advantage of this is that the network can be applied repeatedly for different tasks without the need for retraining.

8.4 Future Work

This project is open to a large degree of future work. It has been illustrated that an effective false alarm filter can improve the performance by reducing the false alarm rate, but it is entirely dependent on the performance of the neural network in achieving a perfect detection accuracy. Improving training by using the off-centre cutouts training approach and then applying the Donut algorithm false alarm filter may be able to achieve better results.

The Donut algorithm false alarm filter cannot currently deal with noisy conceptualised object locations. This algorithm is easily extendable and further investigation into creating a more generic version may be able to improve the use of this approach when used with a neural network based on raw pixel values.

It would also be interesting to see the detection performance of applying the Donut algorithm false alarm filter to the result of a network trained with off-centre cutouts. This will most likely not perform well, as the results of this project have concluded that the Donut algorithm false alarm filter, will generally only improve the results when applied to a pixel statistics neural network. Off-Centre training was shown to improve only the detection results of raw pixel values based neural networks. Potentially, the combination of the two will not be very successful but this will require further investigation.

The best set of local region pixel statistics tested in this project are based on concentric shapes that capture the rotational invariance of an object. The discrepancy between the shapes of these objects is relatively insignificant. Further research into whether the number of local regions used plays a significant factor is required. A set of local regions with 5 or 6 concentric circles or squares maybe able to capture more information on complex objects, such as faces. This result indicates that local region pixel statistics are not as domain independent as it was originally perceived.

In concluding this project, neural networks have shown promising performance in object classification and object detection tasks. Where training has not been extensive, excellent results have been achieved in both classification and detection performance for reasonably difficult real life problems. The use of pixel statistics can produce results just as effective as using raw pixel values and this is advantageous as the network is smaller and thus generally more efficient. This system would be excellent resourced as a quick and simple system for monotonous real world classification and detection tasks, such as in the quality control in product lines. This would free up human resources labour for more human specific tasks and be just as effective.

Bibliography

[1] The oxford english dictionary. http://dictionary.oed.com, 2003.

[2] Bhowan, U. A domain independant approach to multi-class object detection using genetic programming. BSc Honours Report, 2003.

[3] Chow, R. Multiple class object detection using pixel statistics in neural networks. BSc Honours Report, VUW, 2002.

[4] Cun, Y. L., Jackel, L., Boser, B., Denker, J. S., Graf, H. P., Guyon, I., Henderson, D., Howard, R. E., and Hubbard, W. Handwritten digit recognition: Applications of neural network chips and automatic learning. *IEEE Communications Magazine* (1989).

[5] Haykin, S. *Neural Networks.* Macmillan College Publishing Company, Inc, 1994, ch. Introduction.

[6] Hecht-Nielson, R. *Neural networks for vision and image processing.* The MIT Press, 1992, ch. Neural Networks for Image Analysis.

[7] Isaksson, M. Face detection and pose estimation using triplet invariants. Tech. Rep. LiTH-ISY-EX-3223-2002, Avdelning, Institution, 2002.

[8] Koza, J. R. *Genetic Programming.* The MIT Press, 1993.

[9] Loveard, T. *Genetic Programming for Classification Learning Problems.* PhD thesis, RMIT, 2002.

[10] Martin T. Hagan, H. B. D., and Beale, M. *Neural Network Design.* PWS Publishing Company, 1996.

[11] Mengjie Zhang, P. A., and Chow, R. Pixel statistics based neural networks for domain independent multiclass object detection. Tech. Rep. CS-TR-02-15, School of Math. and Comp. Science, VUW, 2002.

[12] Metz, C. E. Roc methodology in radiologic imaging. *Investigate Radiology* (1986).

[13] Moran, M. Friendly fire is all too common. Microsoft NBC News, http://www.msnbc.com/, March 2003.

[14] Pritchard, M. Genetic programming for multi-class object detection. BSc Honours Report, 2002.

[15] Rai, N. Pixel statistics in neural networks for domain independent object detection. Master's thesis, RMIT University, Melbourne, Victoria, Australia, 2001.

[16] Reeves, T. E., Welch, O. J., and Welch, S. T. 10-fold cross-validation for discriminant analysis.

[17] Rumelhart, D., Hinton, G., and McClenlland, J. L. *Parallel Distributed Processing, Explorations in the Microstructure of Cognition, Volume 1: Foundations,*. The MIT Press, 1986, ch. A general framework for parallel distributed processing.

[18] Smart, W. Genetic programming for multi-class object classification. BSc Honours Report, 2003.

[19] Zhang, M. *A Domain Independent Approach to 2D Object Detection Based on the Neural and Genetic Paradigms*. PhD thesis, RMIT University, 2000.

[20] Zhang, M. *Neural Networks for Multiple Class Object Detection Package*. Dept. of Comp. Sci. RMIT, February 2000.

[21] Zhang, M. Pixel based neural networks for multiclass object detection. Tech. Rep. CS-TR-01-13, School of Math. and Comp. Science, VUW, 2001.

[22] Zhang, M., and Andreae, P. Pixel statistics in object detection. Tech. rep., School of Math. and Comp. Science, VUW.

Appendix A

Neural Networks for Multiple Class Object Classification and Detection Package

This is a draft instruction of using the package of neural networks for multiple class object detection [20]. The neural networks used here are those with the three layer feed forward network architecture trained by the backpropagation algorithm. There are two parts of this package. The first was not modified for use in this project and is designed to use the raw image pixel data directly to form the input patterns. There are two versions of this where one will work in binary and the other in ASCII format. The binary program will be faster to run than the ASCII program, but the ASCII program has the benefit of producing human readable data.

The second part of this package has been worked on by several developers. The last implementation was done by Roy Chow in his project 'Multiple Class Object Detection using Pixel Statistics in Neural Networks' [3]. The implementation by Chow had configured the sweeping procedure to work with a rectilinear set of local regions. This project has expanded upon that in having the program handle the local regions used in this project. The program has also been written for easy scalability and ease of use in selecting the region that will be used in the sweep.

A.1 Overview of the Approach

To give a clear view of the approach, the format is as follows [20]:

1. Assemble a database of pictures in which the locations and classes of all the objects of interest are manually determined. Divide these full images into three sets: a *training set* for the network training procedure, a *testing set* for the network classification testing procedure and a *detection set* for the network sweeping procedure.

2. Determine an appropriate size (n) of a square which will cover all objects of interest and form the input field of the networks. Note that the algorithms are currently only defined for an n of an even size and therefore n must be even. Build training and test sets of the object (sub-image) example by cutting out squares of size n from the full training and testing image sets, which are used to form the network patterns. Each of the squares (sub-images) only contain a single object and/or a piece of the background. Note that the background is also considered as a class, but not the class of interest.

3. Use the pattern generation program to create a pattern file of all the cutouts from the training and testing sets in that order. This is dependant upon the approach used. The

programs are separate for the various local regions of pixel statistics or raw pixel values. Manually modify the top line of the pattern file with the number of cutouts that are dedicated to training and the number dedicated to testing.

4. Determine the network architecture. A three layer feed forward neural network is used in this approach. For pixel statistics, the number of features of the set form the inputs of a training pattern. For raw pixel values, it is the $n \times n$ number of pixel values. The classification (number of classes) is the output. The number of hidden nodes is empirically determined based on the experiments.

5. Train the pre-defined network by the backward error propagation algorithm. The network training will be stopped according to the termination strategy/strategies selected.

6. Use the trained network for object classification. This will use the trained network to evaluate the classes of the patterns allocated for testing. If the class evaluated by the network, corresponds to the actual class of the object, it is considered a correct classification. If the class evaluated by the network, does not correspond to the actual class of the object, it is considered a misclassification. This can be used to compute the *accuracy* of the network.

7. Use the trained network as a moving window template to detect the multiple class objects of interest in large pictures in the full detection image set. This will produce data on the potential object locations for each class.

8. Use the selected false alarm filter to screen the target locations identified by the potential locations so that the false alarms can be reduced. It is important to note that the thresholds for different classes are different. This component will also evaluate the performance by object matching and calculating the *detection rate* and *false alarm rate*.

Appendix B

Sweeping Images

The figures in this appendix illustrates the discrepancy between approaches through the imaging of the potential target locations for database 1. The intensity of the pixels corresponds to the activation from the network for the particular class, therefore a high activation (e.g. 99%) will have a white pixel at that location, a medium activation (e.g. 50%) will have a grey pixel and a low activation (e.g. 1%) will have a black pixel. These images assist in illustrating the discrepancy for the detection performance between the different local regions and raw pixels values.

B.1 Database 1

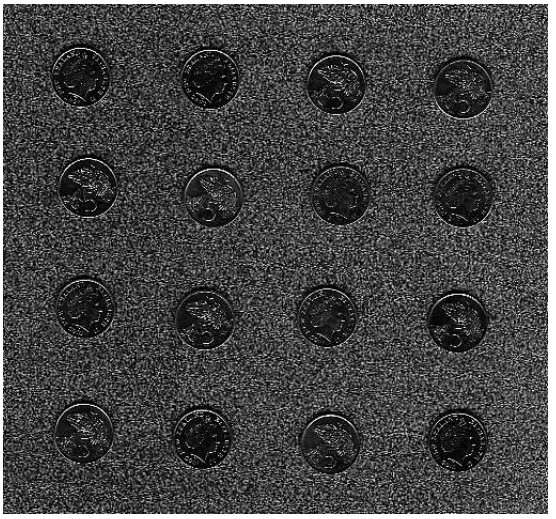
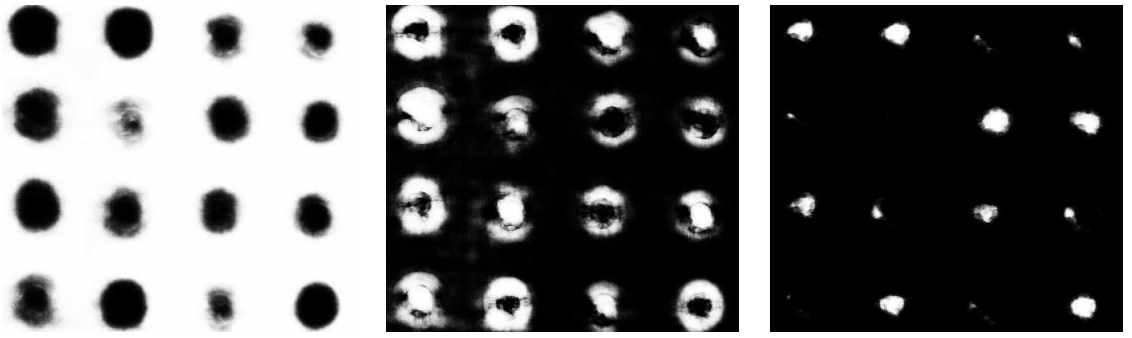
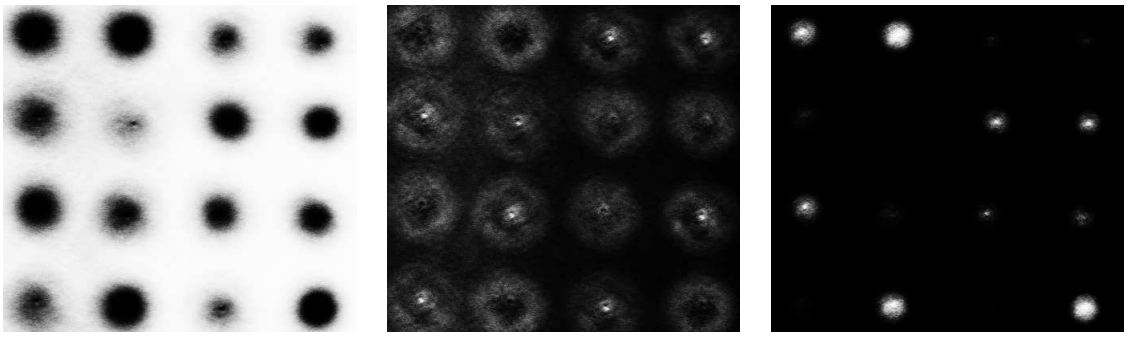


Figure B.1: Example Image from Database 1 being Swept



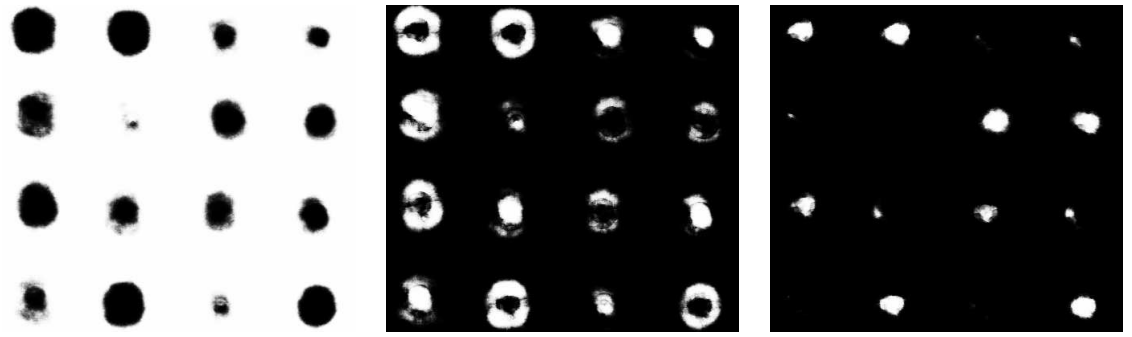
Background Sweep Tails Sweep Heads Sweep

Figure B.2: Concentric Circular Regions sweep of B.1



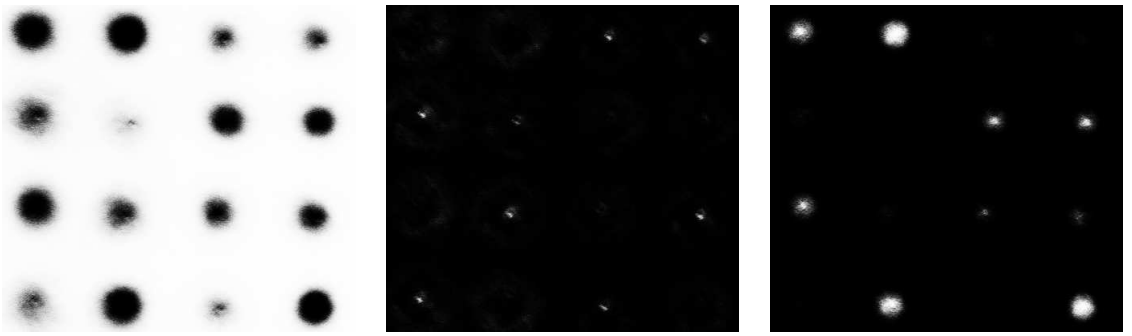
Background Sweep Tails Sweep Heads Sweep

Figure B.3: Raw Pixel Values sweep of B.1



Background Sweep Tails Sweep Heads Sweep

Figure B.4: Concentric Circular Regions sweep of B.1 with the inclusion of Off-Centre Cutouts in Training



Background Sweep Tails Sweep Heads Sweep

Figure B.5: Raw Pixel Values sweep of B.1 with the inclusion of Off-Centre Cutouts in Training