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

School of Mathematics, Statistics and Computer Science
Te Kura Tatau

PO Box 600
Wellington
New Zealand

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

Object Detection using Neural Networks and Genetic Programming

Barret A. Chin

Supervisor: Dr. Mengjie Zhang

12 October 2007

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

Abstract

Object detection has become an interesting and important research topic within computer science. Everyday we are facing increasing volumes of data where one of these types of data that is of great interest are images. This has led to the need for computers to search, recognise and classify relevant images for all forms of applications from military to civil. The fields of neural networks and genetic programming have achieved these goals with varying success. This project will investigate the task of effective and efficient object detection using neural networks and genetic programming. This project will begin by examining a standard neural network system and a standard genetic programming object detection system to determine which system is faster and what approaches should be taken in order to develop a more effective and more efficient object detection system. In the end this project will produce an improved approach that is more effective and more efficient than most traditional object detection systems.

Acknowledgments

I would like to thank Dr. Mengjie Zhang for his support, advice and the use of his Neural Network package to act as the standard Neural Network system. I would also like to thank Will Smart for providing the use of the VGP system to act as the standard GP system. Finally I would also like to thank the GP discussion group for ideas and very helpful suggestions.

Contents

Figures

List of Tables

Chapter 1

Introduction

Object detection is the task of finding objects of interest in a large image. The process involves both sub-tasks of object classification and object localisation. Object classification refers to the task of distinguishing between images of different kinds of objects where each image is an image of only a single object known as ″image cutouts″; the idea is to categorise these images into classes or groups. Object localisation refers to the task of determining the positions of all of these objects of interest from a large image. Object detection can be used in many applications from detecting different kinds of coins or different kinds of shapes on a table to specialised medical applications of detecting the difference between micro aneurisms and haemorrhages in retina images.

Neural Networks (NN) is an artificial intelligence field that has shown a lot of recent promise. It is a connectionist approach and is inspired by a part of the human brain. Similar to the brain the NN structure consists of nodes or neurons and works together to produce an output function. It has been often criticised and regarded as a ″black box″ but past works [20] has shown that this is not the case and that it is possible to visualise the network behaviour by analysing the weights in the learned networks. This is evidence that we have yet to use and understand the full potential of NN.

Genetic Programming (GP) is a recent addition to artificial intelligence and is an evolutionary algorithm inspired by the idea of biological evolution. Instead of biological evolution, GP is used to find the most evolved optimal computer program to perform a user-defined task. The process involves initialising a population of computer programs and using a measure function called a fitness function to determine the next generation of computer programs. This process terminates when the optimal solution is found or the best program does not improve over a number of generations. GP has shown rapid progress with a great degree of success in past works [6, 8, 9, 14, 19, 21, 22, 23] in terms of achieving more precise programs that make less errors.

1.1 Motivation

Object detection is a very popular field of research as it can be used in many applications from military to civil applications. Heavy research has been conducted by the military and tertiary institutions to achieve the dream of computers being able to automatically recognise objects with great precision. The military could automatically spot and classify vehicles or missiles, companies could have stronger security with the addition of accurate and reliable facial recognition and we could solve vision identification problems such as detecting small medical diseases that the human eye has had difficulty in doing so even with powerful equipment. Most of the past work focuses on improving object classification or have tried new interesting approaches to refine the process in order to improve overall object detection performance. There are two fields that have been gradually improving object detection: GP being one and NN being the other. In the case of GP, it has been about how to better create the perfect object program detector while with NN it has been about how to train the NN architecture to better classify the objects of interest. This project will look into a few of these methods with the objective of how to best improve the efficiency and effectiveness of the object detection task.

1.2 Research Goals

The goal of this project is to achieve effective and efficient object detection performance. Effective object detection performance refers to object detection accuracy while the term efficient object detection performance refers object detection speed. The project will attempt to achieve this goal by first examining a base system and then advance further by applying new techniques to the object classification task and the object localisation task to improve the overall object detection process.
The specific goals of this project are to:

1. Develop a NN-based approach to object detection and investigate whether this approach can achieve good object detection performance on all data sets? If not, can we identify the limitations and devise a method or idea to address these limitations.

2. Introduce pixel statistics into the above approach and investigate whether it will address the limitations in the above approach and will the pixel statistics improve the object detection performance over the the use of raw pixels as features.

3. Develop a GP-based approach to object detection and investigate whether GP will address the limitations discovered in the above approaches and whether object detection has improved using a GP-based approach.

4. Develop an improved approach that addresses the limitations from the above approaches which will also achieve a much higher object detection performance on all data sets by using the information gained from the above specific goals.

1.3 Major Contributions

The project report makes the following three contributions:

1. This report explores and shows the object detection performance of a NN-based approach and a GP-based approach. Each of these approaches have a unique architecture which has their own strengths and weaknesses. In past works, these two approaches have rarely been explored together and it is unclear whether one approach is better than the other. This project explores both and shows that one is better than the other.

2. This report also shows how we apply each of the two approaches on some object detection problems in order to identify the limitations within each approach. The demonstration of these limitations is necessary to show how one approach is able to compensate the other. Using the idea of applying other approach's strengths to compensate another approach's weaknesses, we introduce an improved approach that will address the issue of objection performance in terms of efficiency.

3. Lastly the report shows our improved approach based on feature selection to addresses the weaknesses of both the above NN and GP approaches. The improved approach uses a new ″Region Refinement″ component that uses features to identify redundant regions to achieve object detection performance in terms of efficiency.

1.4 Structure of this Document

The report from this point onwards is organised as follows. Chapter 2 presents background information in Object Detection, Genetic Programming, Neural Networks, briefly about Probably Approximately Correct (PAC) Learning theory and Vapnik Chervonenkis (VC) Dimension. Chapter 3 will describe the data sets that the systems will be tested with. Chapter 4 will provide a detailed description of the object detection method and the first NN experiment with raw pixels as features; the results and limitation of the system are identified and discussed. Chapter 5 will then describe the second NN experiment but we use pixel statistics as the features instead. In this second NN experiment, we will describe the differences in the object detection method and discuss its results and limitations in comparison to the first NN experiment. Chapter 6 will describe the GP experiment and its object detection method differences, the experiment will be compared to the NN experiments and the chapter will summarise the GP's performance on the data sets. Chapter 7 describes the new approach and the experiment using the new approach. The chapter will also discuss the new approach's performance against the experiments with the NN and GP systems. Finally Chapter 8 will present the overall conclusions of the project in terms of whether the overall research goal was achieved and whether the specific goals were also answered. Chapter 8 will also specify indirect findings of the project and a list of future work.

Chapter 2

Background

This chapter is divided into five sections and is devoted in first providing a brief overview of the topic that describes the main components and structure of the approach. It is then followed by an overview of the GP and NN with a detail discussion on the two techniques. PAC learning theory and VC dimensions are briefly discussed and finally some problems and limitations in past works is presented.

2.1 Computer Vision

Computer vision is the area concerning with designing artificial systems to automatically interact with images in some way to achieve some goal. Computer vision can involve many different fields but the one that mostly relates to this project is known as object (or pattern) recognition which is also a sub-topic of machine learning.

2.1.1 Object Recognition

Object recognition is the task of acting on raw data with the objective of categorising the data. In this project the action conducted on the raw data is the extraction of patterns that is used to categorise the data. These patterns consist of features usually based on either prior knowledge, statistics or even plain raw data. The pattern extraction is used for the learning process where the artifical system is learning to discriminate different classes of data. The term data used in this project is referring to image data therefore pattern extraction involves the use of the pixel intensities of the images. In this project, we will therefore consider the use of raw pixels and pixel statistics as features. There are also three learning strategies which are known as supervised, unsupervised and hybrid.

Supervised learning is where a teacher is involved in the pattern extraction process. This type of strategy involves the teacher using two data sets known as a training set and a test set. The teacher manually identifies the correct classes that each image belongs to and the artifical system tries to create rules that correctly matches these designated classes. These rules are examined by the test set to determine the performance of the artifical system.

Unsupervised learning is where a teacher is not involved and there are no correct classes assigned to the data set. This type of strategy forces the system to discover some correlation between groups of data and uses these correlations to classify new data.

Hybrid learning is the type of learning that describes learning strategies that do not really completely fall in the above two learning types. Hybrid learning consist of both supervised and unsupervised learning characteristics.

2.1.2 Object detection

Object detection is the task of finding objects of interest in a large image. Object detection involves both object classification and object localisation. Object classification refers to the task of distinguishing between images of different kinds of objects where each image is an image of only a single object; the idea is to categorise these images into classes or groups. Object localisation refers to the task of determining the positions of all of these objects of interest from a large image.

The object detection method used in this project is a type known as the ”single pass approach” [7, 21, 22, 24] where only a single program sweeps through an image once. During one sweep, the program will attempt to gather all objects of interest. While the other type known as the ”segmentation based approach” [8, 9, 21, 22, 24] is a multi-stage process. Instead of a single program sweeping for the objects of interest, there are several programs responsible for different specific tasks. These tasks can be preprocessing, segmentation, feature extraction or classification, but this can vary depending on the system architecture or approach. For the purpose of a simple control or reference system to make comparisons with, it is my belief that the single pass type is sufficient for this project.

The system will be using a domain independent approach, where low level features such as raw pixels and pixel statistics will be used directly as inputs to the classifier or detector. The reason for using domain independent features rather than domain specific features is simply so that we can run the object detection without the need to manually select and prepare the features before hand and to create a solution that is not tailored for one type of problem.

Finally the object detection method is intended to be used on a ”one class object detection problem”; a one class object detection problem or also known as object versus background problem is used, as the processing of additional classes is beyond the scope of this project.

2.1.3 Performance Evaluation

Performance of the experiment will be measured by a detection rate (DR) and a false alarm rate (FAR). The DR is the number of objects correctly reported as a percentage of the total number of real objects in the image or images. While the FAR is the number of non-objects incorrectly reported as objects as a percentage to the number of real objects. Note that DR is number between 0 and 1 or 0 and 100% but the FAR can any positive number or percentage. For example assume the detection system is looking for shapes in figure 3.1 (a) on page 9 and it reports that there are 18 objects detected. If only 12 of these are real objects then the DR will be (12 / 15) * 100 = 80% while the FAR will be (6 / 15) * 100 = 40%.

The objective of the object detection system is to obtain a high DR and a low FAR but the objective of this project is to observe the standard DR and FAR from a standard base system for the image data sets that are discussed in chapter 3 and to achieve fast object detection.

2.2 Genetic Programming

Genetic programming is a method that searches for the best computer program that is able to achieve the goal of some task such as object detection. Koza is the pionner in using genetic programming to solve complex problems, most of his early work is presented in [5]. The search process involves the evolution of an initial population of computer programs until either the best program is found or the maximum number of generations have been reached. To determine whether a program is good or not, a measure function known as the fitness function is used to determine the fitness of a program. The fitness functions varies depending on the problem but usually involves a data set or fitness cases to determine the fitness rating. This therefore implies that the GP process uses a supervised learning strategy which is known as the evolutionary engine.

2.2.1 Program Representation

Programs are represented in memory as a variable sized tree consisting of non-leaf and leaf nodes. Non-leaf nodes corresponds to a function belonging to the declared function set while leaf nodes correspond to a terminal belonging to the declared terminal set. A program or tree can be represented in LISP S-expression, for instance if the program was ((2x + y) * z) then as LISP it would be (* (+ (* 2 x) y) z). Figure 2.1 shows this program as a tree.

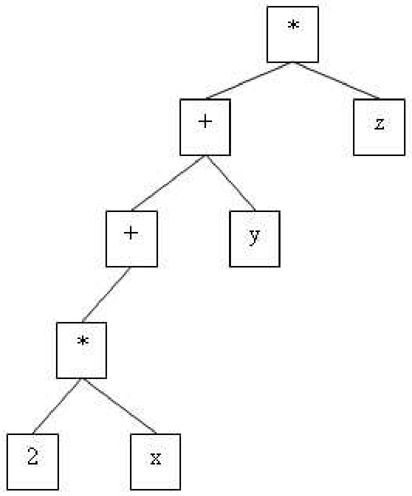


Figure 2.1: Program representation in GP

2.2.2 Operators

The evolutionary engine relies on genetic operators such as crossover, mutation and reproduction (also known as elitism) to evolve the individual programs in the population. GP uses these operators to alter the genes (functions or terminals) of the individual program. Each operator performs a special operation to create a successor or successors to continue to the next generation. Crossover selects two programs to be replaced by two new programs by swapping a sub-tree between the two programs. Mutation selects a program and creates a new program by randomly removing a sub-tree and replacing it with a random sub-tree. Finally reproduction (or elitism) is just the copying of the selected program to carry on in the next generation.

2.3 Neural Networks

A neural network consist of connected nodes representing the connected neurons of a human brain. These nodes are not fully connected and are usually grouped in layers. Each neuron contains a link to another neuron that contains a weight that is adjusted during training and a transfer function is applied to the output of every neuron. Traditionally there are two types of neural networks. One type is known as the single-layer feed forward network which consist of an input layer where the inputs are feed in and an output layer where the output value is produced, figure 2.2 (left) shows an example of a single-layer feed forward network. The other type is known as multi-layer feed forward network; as shown in figure 2.2 (right) these networks have the distinguishing feature of one or more hidden layers to contain additional information. A common algorithm to train neural networks is the back propagation algorithm [13], which this project will use for the NN system.

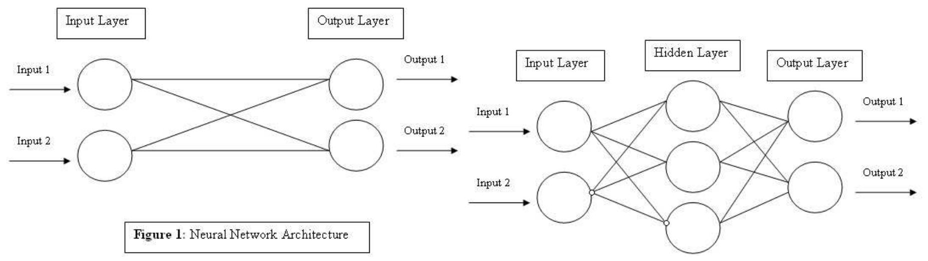


Figure 2.2: A single-layer feed forward neural network

2.4 Probably Approximately Correct Learning Theory and Vapnik Chervonenkis Dimension

PAC learning theory [15] is a machine learning framework where if a learner is given a training set, the learner is able to learn the concept regardless of the approximation ratio or probability of success as the learner is bounded approximately to a function with high probability.

VC dimension [16] is a set of instances that is shattered by a hypothesis space where if and only if for every dichotomy of the set of instances there exist some hypotheses in the hypothesis space that is consistent with this dichotomy.

PAC learning theory and VC dimension can be used together to determine the number of training examples required for given NN architecture. By caculating the required number of examples, we can avoid overtraining or training with insufficient examples.

2.5 Problems and Limitations

1. Past work [20] demonstrated that NN had difficulty in detecting objects with difficult problems.
2. Other past work [6] states that the NN architecture is slow as it relies on foreign or external manual aid in tasks such as preprocessing.
3. GP is likely to require a large search space for difficult problems and this can slow down object detection tremendously.

These problems and limitations need to be taken into account as modifying or creating a new variation of the existing system can face the above obstacles which pull the research away from achieving the research goals of a more efficient and effective object detection approach.

Chapter 3

Image Data Sets

In order to achieve the goals of this research project as specified in chapter 1, the following four image data sets have been selected. Each image data set is intended for one-class object detection and is of increasing difficulty in a key aspect. The first is relatively trival, the second set is slightly harder but still easy, the third set is harder by adding noise to the background and the last one is difficult and complex. All image data sets are greyscale.

Trivial Shapes: The easiest problem is a computer generated image containing only one type of shape that is a little noisy on a light uniform background. There are 10 images in this set consisting of a total of 300 cut out images where 150 are objects and 150 are background. There will be 150 cut out images for the training set and 150 cut out images for the test set. One full image will be generated randomly for the purpose of sweeping. It should be noted that there are only two classes in this image: shapes (class 1) and background (class 2). See figure 3.1 (a).

Easy Shapes: This problem is harder than the trivial set where this set also has three different types of shapes with varying degrees of brightness that can be considered as a single object class. See figure 3.1 (b).

Medium Shapes: This problem is even harder than the easy set as this set has a noisy background where it's degree of brightness is very similar to one of the shape types. See figure 3.1 (c).

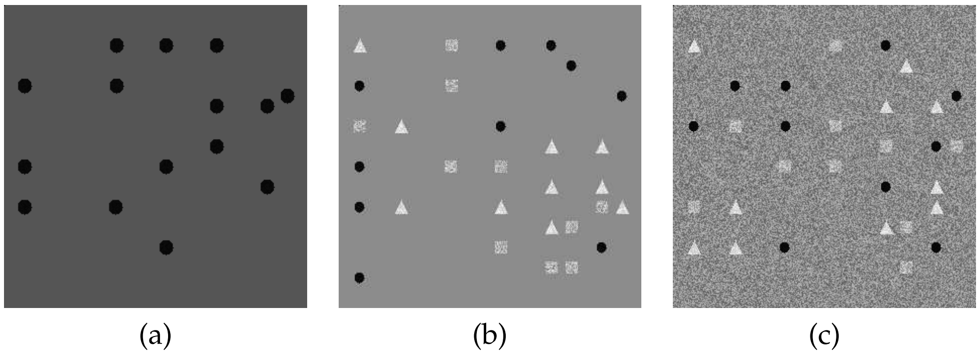
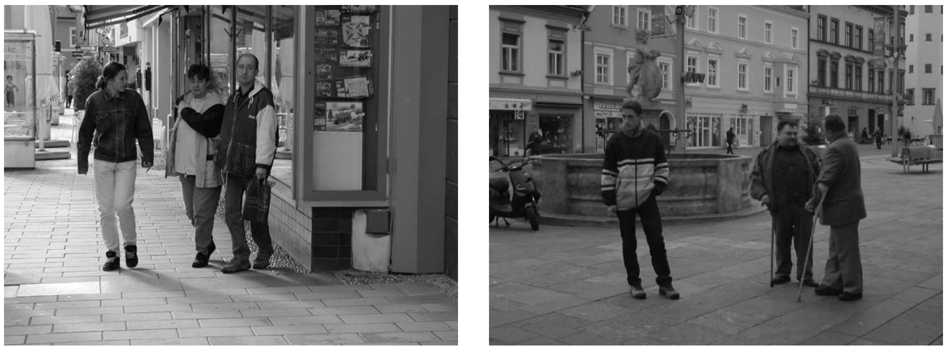


Figure 3.1: Example images from the first three data sets. (a) trivial; (b) easy; (c) medium.

Hard Faces: The most difficult problem is a photographic image containing at least two faces on a varied background. There are 22 images in this set consisting of a total of 106 cutout images where 53 are viewable faces and 53 are background. The images were collected from the PASCAL Object Recognition Database Collection[1]. There will be 53 cut out images for the training set and 53 cut out images for the test set. One full image will be selected randomly for the purpose of sweeping. It should be noted that there are only two classes in this image: viewable faces (class 1) and other (class 2). See figure 3.2 for two different examples and an explaination of what is counted as an object class. In figure 3.2 (a), all three faces will be counted as an object class as it is determined manually that the face on the left is viewable. In figure 3.2 (b), only the front two faces will be counted as an object class as it is determined manually that the face on the left and on the right in the distant background is not viewable.



(a) (b)

Figure 3.2: Example images in the hard data set. (a) hard example 1; (b) hard example 2.

Chapter 4

Neural Networks with Raw Pixels

This chapter outlines and describes the experiment conducted using NN with Raw Pixel features. The purpose of this experiment is to gain an insight and a reference to a base object detection system that utilises NN so that comparisons can be made with the later developed system's performance or the performance of other systems. The NN system that this chapter will use is one from past work [20]. We are only interested in object detection and therefore only a simple object detection system is required for the purpose of this experiment.

4.1 Object Detection Method

In chapter 2; we briefly described that this project employs the object detection method known as the "single pass approach" where a single program is used as a template to sweep through an image once. Upon this sweep, the program will attempt to gather all objects of interest. Finally we determine the performance results after the sweeping process is completed.

The NN system's object detection method is outlined as follows:

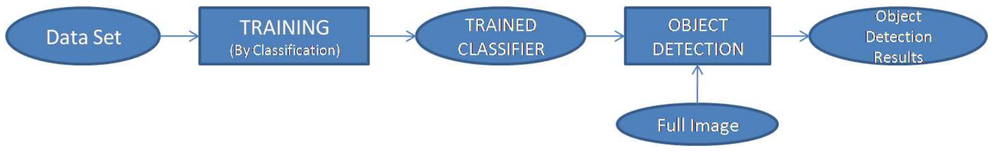


Figure 4.1: The Object Detection Method

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

2. Determine an appropriate size of n x n dimension which will cover a single object of interest and form the input field of the NN. A classification image data set is to be created by cutting out squares of size n x n dimension from the detection training set where each cut out image either contains a single object or a part of the background. The n by n dimension is determined to be 14 x 14 for Easy and Medium Shapes data set and 80 x 80 for Hard Face data set.

3. Determine the NN architecture. A three layer feed forward NN is used in this approach where the n x n pixel values from the input field from the inputs of the training

data and the classification class is the output.

4. NN is trained by the back propagation algorithm on the classification training data set. When the training is complete, the result is a trained NN which is then tested on the classification test data set to measure object classification performance. This step is important as it is a check to ensure that the NN is not overtrained.

5. The trained NN is then used for the object localisation task in a moving window template fashion on a full image to sweep for all objects of interest. An object is reported if the output of the NN for a class exceeds a given threshold. See figure 4.2.

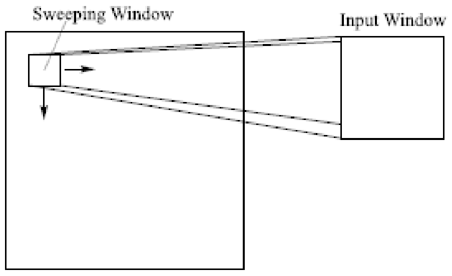


Figure 4.2: The Sweeping Process

6. Finally the object detection performance of the NN is evaluated by calculating the DR and FAR.

4.2 Chapter Goal

The goal of this chapter is to accomplish the first specific goal which was to develop a NN-based approach to object detection and investigate whether this approach can achieve good object detection performance on all data sets? If not, can we identify the limitations and devise a method or idea to address these limitations.

4.3 Results

4.3.1 Experiment Setup

The NN architecture uses the following parameters:

- **Learning Rate:** 0.001
- **Momentum Value:** 0
- **Critical Error:** 0.01 for all data sets but 0.08 for Easy Shapes data set
- **Random Range:** [-1,1]
- **Number of Layers:** 3
- **Units in Input Layer:** 196 for the shapes set or 6400 for the faces set
- **Units in Hidden Layer:** 3 for Trivial Shapes, 8 for both Easy and Medium Shapes and 10 for Hard Faces.
- **Units in Output Layer:** 2

During NN training, the NN parameters were experimented to obtain an optimal NN. The primary parameters that were altered were the learning rate, momentum and the number of hidden units or nodes. Many learning rates were tested but a learning rate of 0.001 seem to work well for all data sets. Hidden nodes of 3, 8, 10 and 12 were tested and the aim was to keep it as low as possible to avoid high chance of overfitting.

4.3.2 Object Classification Results

The NN training and testing results for object classification are presented in table 4.1. In all cases the the network training procedure and testing was repeated 10 times and the average from the 10 trials were used as the object classification result. However for the Easy Shapes data set, there seemed to be a problem for the NN to converge to a critical error of 0.01 when the critical error reached to 0.06. In response the critical error was specifically re-set to 0.06 and as seen in table 4.1, training still took longer than any of the other data sets. However the NN system did achieve near ideal performance for the Trivial and Easy Shapes, but this is expected as from past work [20], we see ideal results from a similar case. But for Medium Shapes and Hard Faces there seems to be some overfitting. Further examination of the results the reason for the problem with the Easy Shapes data set is because the background is a dark background with similar pixel intensity to the pixel intensity of the black circles, indicating that the NN system has problems in the detection objects on a background with similar intensity. The medium data set can reinforce this assumption as it introduces a distorted background, we see an indication of the same problem where the test accuracy is of about 15% difference compared to the training accuracy. Lastly for the Hard Faces data set; we find that it had difficulties in identifying faces from the background where it was only able to classify faces 56.79% of the time.

ImageDatabases	*NetworkArchitecture*	*TrainingEpochs*	*TrainingAccuracy*	*TestAccuracy*
Trivial Shapes	196-3-2	1671.7	100.00%	99.40%
Easy Shapes	196-8-2	4065.8	90.67%	100.00%
Medium Shapes	196-8-2	1472.8	100.00%	85.53%
Hard Faces	6400-10-2	53.4	100.00%	56.79%

Table 4.1: Object classification results for the three databases (NN with Raw Pixels)

4.3.3 Object Detection Results

The object detection results are shown in table 4.2. Theses results are calculated with a tolerance of half the input window's size. Therefore with the shapes data sets the tolerance is 7 pixels which means anything within 7 pixels of the correct centre point is considered a match while for the face data set it is a tolerance of 40 pixels. One random full image is selected for the sweeping and the DR and FAR are calculated. So far it seems that the Trivial obtained ideal results but the Easy Shapes data set seem to had difficulties and detected about half the objects but maintained a low FAR. However looking at the object sweeping maps below, the Trivial Shapes show perfect performance. But for Easy Shapes, Medium Shapes and Hard Faces; the NN is not very clear whether an object or background exists in its input window. Specificially for the Hard Faces we find that it had difficulties in identifying faces from the background where it was only able to classify faces 50% of the time its lowest FAR of 900%.

ImageDatabases	*Threshold*	*DetectionRate*(%)	*FalseAlarmRate*(%)
Trivial Shapes	0.50	100%	0%
Easy Shapes	0.57	53.33%	13.33%
Medium Shapes	0.58	73.33%	0%
Hard Faces	0.92	50%	900%

Table 4.2: Object detection results for the three databases (NN with Raw Pixels)

4.3.4 Analysis of Results — Object Sweeping Maps

After an NN sweep or the object localisation task, an object sweeping map for each object class is produced. This object sweeping map shows the object centres of all objects that were detected during the sweep. The images in figures 4.3, 4.4, 4.5, 4.6 are examples of an image and their object sweeping maps. With a sweeping map we can try to understand what the NN is doing during the sweeping process. The meaning of a sweeping map is as follows: black area means no match, white areas mean a match and when there is a partial match the colour grey is used for the centre instead. Therefore the closer the shades of grey are to the colour white, the greater the confidence that this particular area is likely to be a match. Ideally a sweeping map would only show white dots of all objects belonging to the object class but this is almost impossible due to noise. In this experiment we do see the ideal case for the trivial data set. But for the easy and medium data sets, we see uncertainty of grey patches varying from dark ones to light ones. These two different patches are perfect examples of two different uncertain spots where the light one is in favour of classifying that patch or spot as the object class while the darker patch is in favour of classifying the area as the background.

Looking at its sweeping map in figure 4.5 we find that the grey squares which resembles most closely to the background are the 'least certain' object that was detected, as we see they fade into the background. Comparing this to figure 4.4 (the sweeping map for the Easy Shapes data set) we see similar maps with similar problems where in the case the Easy data set, we have the black circles instead fading into the background. At this point a reason I believe for the odd long training session for the Easy data set was due to the luck of the weight initialisation. However in consideration of PAC learning theory and VC dimension, it is most likely the large NN architecture lacked sufficient training examples. Lastly for the Hard Faces data set; we see the NN is uncertain of the exact positions of the faces where parts of the image are classified as objects. I believe the main problem is caused by the over-training as shown by the differences in the training accuracy and test accuracy. Another problem that is likely to be large amounts of extremely different backgrounds that the NN is exposed to that the NN had not seen during training. Once again PAC learning theory and VC dimension questions whether there were sufficient training examples for such a such NN architecture.

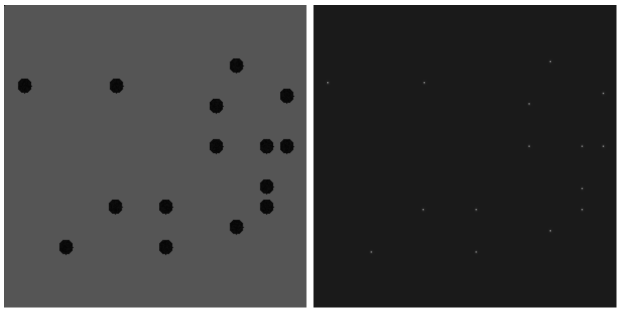


Figure 4.3: Trivial Shapes Original Image (raw pixel features), Object-sweeping-map (raw pixel features)

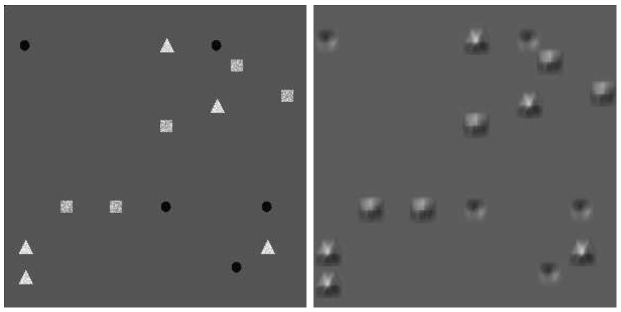


Figure 4.4: Easy Shapes Original Image (raw pixel features), Object-sweeping-map (raw pixel features)

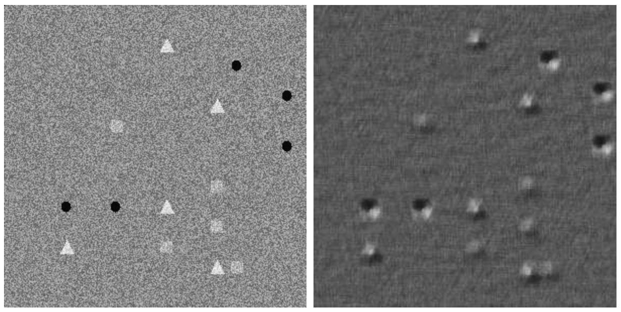


Figure 4.5: Medium Shapes Original Image (raw pixel features), Object-sweeping-map (raw pixel features)

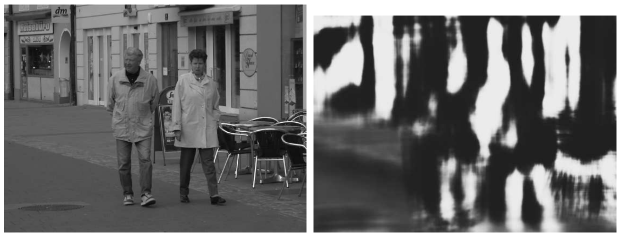


Figure 4.6: Hard Faces Original Image (raw pixel features), Object-sweeping-map (raw pixel features)

4.4 Summary

This chapter presented an experiment of a standard three layer feed forward neural network object detection system. The experiment showed that the system could perform the tasks of both object classification and object localisation on all data sets. However the experiment showed only good performance for object classification with the exception of the Hard Faces data set where the performance was not good when the NN system was applied to the sweeping process. The outcome of the results lead me to believe that the poor performance was due to the confusion between the objects and the background such as similar pixel intensities but PAC learning theory and VC dimension dictates that the experiment is using too few training examples for the large NN architecture. Note that the experiment is using 106 - 300 cut out images for a NN architecture that consist of a total of 1176 - 128000 nodes. PAC learning theory suggests a forumla: input nodes x hidden nodes + hidden nodes x output nodes to determine the required training. Therefore applying this forumla, the NN architecture demands 3,528 training examples. The issue here is that we have too many features for the amount of examples we are providing and we will need to address this problem by reducing the features so that the amount of examples we are providing to the NN system is sufficient or at least closer to the sufficient amount. The solution considered is to use fewer high level features rather than low level raw pixel intensities.

This experiment has hightlighted the following key factors that must be taken note when developing a more efficient object detection system:

1. The NN system achieved poorer performance as the data set used became harder.

2. Major reason for the poorer performance was that the objects were misclassified as the background or background as the object even though object classification was quite high.

3. PAC learning theory and VC dimension dictates that the architecture is too large for such a small amount of training examples, implying we either have to increase the amount of training examples or decrease the size of the architecture.

Addressing the NN architecture size was decided as the next step rather than increasing the training size. This decision was made because a large NN architecture would only inevidently become efficient and would only run into the likely risk of overtraining. Therefore the decision to reduce the NN architecture size from inputting 196 raw pixel features to 12 pixel statistic features would more sensible in developing a more efficient object detection system. The new experiment is described in chapter 5.

Chapter 5

Neural Networks with Pixel Statistics

This chapter outlines and describes another experiment conducted with NN. The purpose of this experiment is to address the problems of the previous NN experiment by reducing the size of the NN architecture. The NN system used in this chapter is the same as the one used in chapter 4 but with the use of pixel statistics rather than low level pixels as the input features. This will effectively reduce the amount of input nodes within the NN architecture which will then reduce the size of the NN architecture so that the required amount of training examples would not be so high as dictated by PAC learning theory and VC dimension theory.

5.1 Object Detection Method

The NN system's object detection method is the same as the one used in the previous chapter but instead the experiment will use pixel statistics as features and regions to specify where we gather the pixel statistics. These differences are discussed in detail as follows:

5.1.1 Features

There are many different features that could be used for an object detection system where some are simple and some are relatively complex, but exploring the effectiveness of these features is out of the scope of this project and we focus on using the simple ones that are used most commonly in past works [2,6,7,9,13,15,16,19]. In this experiment we use pixel intensity of an image to form two kinds of features; they are the mean of a region and the standard deviation of a region. The mean and standard deviation are calculated according to the following formula:

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

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

5.1.2 Regions

The project uses a number of different regions sets in its experiments. Regions are only used when pixel statistics are used therefore when an experiment only uses raw pixel intensities as features, we do not use regions. We use regions to specify the different sets of pixels used to calculate the mean and standard deviations. Therefore for each region, we can extract two features and depending on the number of regions we can extract twice the amount of features as there are regions from an image cutout. The region we use is called Rectilinear Regions as it has shown very promising performance in past works [21, 22, 25]. Rectilinear Regions are illustrated in figure 5.1.

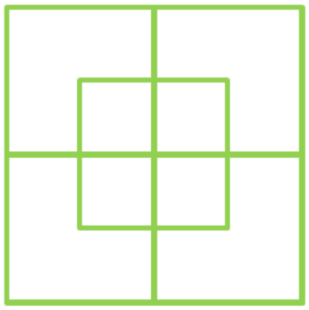


Figure 5.1: Rectilinear Regions

Rectilinear Regions: consist of the following 6 regions; the full image window, central window and each quarter window of the image cutout is a region.

5.2 Chapter Goal

The goal of this chapter is to accomplish the second specific goal which was introduce pixel statistics into the first approach and investigate whether it will address the limitations in the first approach and will the pixel statistics improve the object detection performance over the the use of raw pixels as features.

5.3 Results

5.3.1 Experiment Setup

The NN architecture uses the following parameters:

- **Learning Rate:** 0.01 for Trivial Shapes and Hard Faces, 0.001 for Easy Shapes and Medium Shapes.
- **Momentum Value:** 0.00
- **Critical Error:** 0.01
- **Random Range:** [0.001, -0.999] for Trivial shapes while all other data sets used [0.01, -0.99]
- **Number of Layers:** 3
- **Units in Input Layer:** 12
- **Units in Hidden Layer:** 8 for Trivial Shapes and Medium Shapes, 3 for Easy Shapes and 12 for Hard Faces.
- **Units in Output Layer:** 2

During NN training the NN parameters were experimented to obtain an optimal NN. Different learning rates were tested but it was discovered that very small learning rates of 0.01 and 0.001 worked best to avoid a local minimum point problem as well as the use of a lot of hidden nodes. Specifically Trivial Shapes and Hard Faces used a learning rate of 0.01 while Easy and Medium Shapes used 0.001. It was also discovered that a better NN had used a random range value of 0.01 to -0.99 or 0.001 to -0.999, therefore the data sets seem to favour negative weights as a starting point. Specially only Trivial Shapes used a range of 0.001 to -0.999 while all other data sets used a range of 0.01 to -0.99.

5.3.2 Object Classification Results

The NN training and testing results for object classification are presented in table 5.1. In all cases the network training procedure and testing was repeated 10 times and the average from the 10 trials were used as the object classification result. As we compare this to the table 4.1, we see the classification results have overall improved with the exception of the trivial data set where it has gotten slightly worse. However it is only by 0.67% training and 0.60% testing difference compared the result it got in table 4.1, this is only a slight variation to be consider a significant difference. The key difference is that the Medium Shapes data set is no longer overtrained implying pixel statistics have accomplished more than just slight overall improvements. For the Hard Faces data set, the object classification performance was reasonable but with a test result that deviates from the training result by about 33% suggests overfitting, however there is still an improvement as we compare it to the result from table 4.1.

ImageDatabases	*NetworkArchitecture*	*TrainingEpochs*	*TrainingAccuracy*	*TestAccuracy*
Trivial Shapes	12-8-2	275.9	99.33%	99.33%
Easy Shapes	12-3-2	136.8	100%	100%
Medium Shapes	12-8-2	6429.8	99.40%	96.20%
Hard Faces	12-12-2	30336.8	100%	66.42%

Table 5.1: Object classification results for the three databases (NN with Pixel Statistics)

5.3.3 Object Detection Results

The object detection results are shown in table 5.2. Once again we apply the same tolerance idea to sweeping process. Table 5.2 shows all data sets obtained a 100% DR while maintaining a reasonable FAR with the exception of the Trivial data set where it obtained an ideal result of 100% DR and 0% FAR. For the Easy Shapes we have an FAR of 66.66% and the Medium Shapes data set followed this trend of a higher FAR of 253.33% and finally the Hard Faces data set with a FAR of 3,300%. This is to be expected as each data set are of increasing difficulty. I believe even with pixel statistics, NN still has problems in classifying objects with noisy backgrounds especially if the objects are of similar intensity. This is evident as you examine the object sweeping maps figures 5.2, 5.3 and 5.4 we find the NN is able to visualise the objects in the Trivial and Easy Shapes data sets pretty clearly. But for the Medium Shapes, it begins to lose clarity and this is evident with the triangles which are the closest intensity to the background to receive the weaker weighting. This indicates that the distorted background is hindering the ability of the object detection in terms of object localisation. For the Hard faces data set, it is much worse as from our human eyes the object sweep map looks completely black, acccording to the threshold there are few object weightings. I imagine the major reason for the problem in the classification task is because of the many different types of backgrounds that the NN was not exposed to during the training process. However it should be noted that unlike the previous experiment with raw pixel features, every system is able to detect all objects of interest, except with a higher FAR.

ImageDatabases	*Threshold*	*DetectionRate*(%)	*FalseAlarmRate*(%)
Trivial Shapes	0.74	100%	0%
Easy Shapes	0.74	100%	66.66%
Medium Shapes	0.35	100%	253.33%
Hard Faces	0.0273	100%	3300.00%

Table 5.2: Object detection results for the three databases (NN with Pixel Statistics)

5.3.4 Analysis of Results — Object Sweeping Maps

For the Trivial Shapes data set, figure 5.2 shows the object detection was not perfect as we should only see the object centre as the white pixel and not the whole object. This is not a significant problem as we would prefer the object detection system to detect objects before reaching to the object's actual centre, however the system is designed to put a heavy weight on the input window's centre. But according to the sweeping map, it tells us that it did consider an object detection when the input window was partially over half an object of interest. It is interesting to note that raw pixel features did better than pixel statistics for the trivial data set.

For the Easy Shapes data set, figure 5.3 shows that it has no trouble in detecting the position of the objects but its ability is not as good as the Trivial Shapes data set as we see bigger white spots than the ones shown in the Trivial's sweeping map (figure **??**) and we also learn that the NN is not able to identify the shape differences but this is expected as the system is only interested in the object centre.

The Medium Shapes data set, figure 5.4 shows that there was difficulty in detecting objects on a distorted background. This is especially a problem with the square and triangle shapes where they have similar pixel intensities to the background's pixel intensity. However this does definitely tell us that distorted background does hinder the object detection process in terms of object localisation creating uncertainty as evident with the sweeping map.

For the Hard Faces data set, figure 5.5 show very poor performance implying NN is not suited for object localisation tasks. We see a sweeping map in of mostly black pixels where looking closely you can see slight white patches indicating objects are faintly detected. The poor performance is likely to be caused by the complex backgrounds that the NN was not exposed to during the training process.

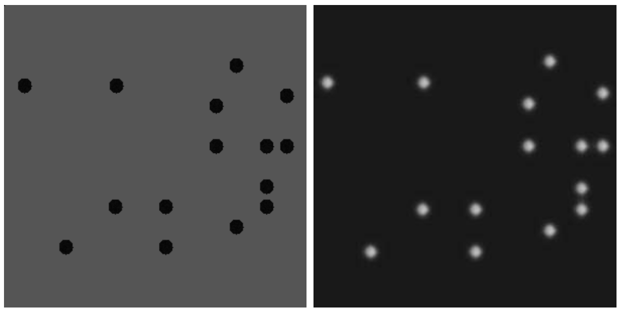


Figure 5.2: Trivial Shapes Original Image (pixel statistics), Objects-sweeping-map (pixel statistics)

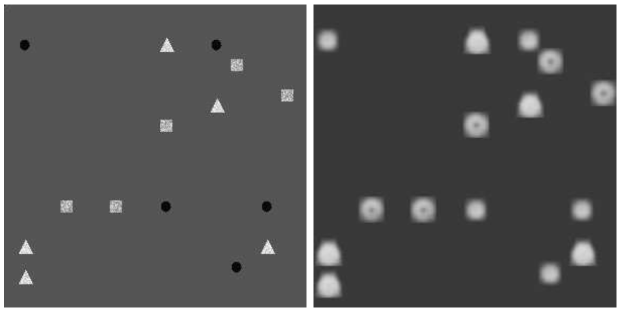


Figure 5.3: Easy Shapes Original Image (pixel statistics), Objects-sweeping-map (pixel statistics)

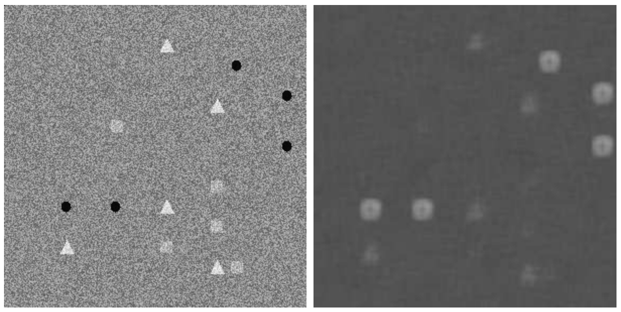


Figure 5.4: Medium Shapes Original Image (pixel statistics), Objects-sweeping-map (pixel statistics)

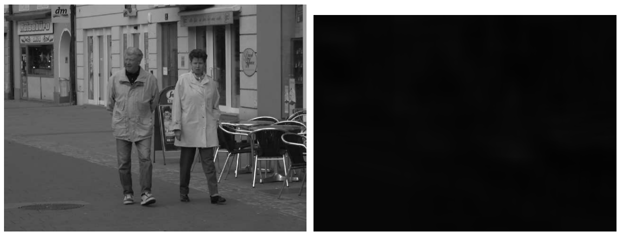


Figure 5.5: Hard Faces Original Image (pixel statistics), Objects-sweeping-map (pixel statistics)

5.4 Summary

This chapter presented an experiment of a standard three layer feed forward neural network object detection system that uses fewer higher level features rather than the numerous lower level features like in the previous experiment. The change of features did improve the system's performance in terms of the most difficult problems. We find that distorted backgrounds close to an object's intensity do cause significant uncertainty with the NN still, this also occurs with the Hard Faces set as well. Although it is not exactly clear with the performance results, the application of pixel statistics has improved the object detection process when compared to the previous NN system. The reason for this is that the experiment achieved performance results of 100% DR which was not obtainable by the previous NN system. But the NN system did not achieve this ideal DR without the cost its FAR as it is a lot worse than the NN system with raw pixel features. But it is important to realise that the overall performance has increased and we now have a system that can solve more of the data sets than the first NN system. This shows progress has been made and the project will proceed by further addressing the low object detection performance by using a non-NN system in an attempt to avoid the limits of the architecture.

This experiment has hightlighted and reinforced the following key factors that must be taken note when developing a more efficient object detection system:

1. Pixel Statistics or higher level features can improve object detection as evident with the clearer object sweeping maps of the objects detected.

2. There is still great dependence for sufficient good quality examples in order for the object detection system to be able to recognise an object from background. This is likely to be a NN architecture problem still.

3. Pixel Statistics and regions seem to provide a lot more expressive power than low level raw pixels as we see from the results that we achieved better object detection performance with less features.

To address the constraints of the NN architecture, the decision for the next step is to experiment with a non-NN architecture based system. Therefore we will experiment on a GP object detection system with the use of pixel statistics to see if the reason for the low performance is due to the NN architecture.

Chapter 6

Genetic Programming with Pixel Statistics

This chapter outlines and describes the experiment conducted with GP. The purpose of this experiment is to address the limitations and issues of the NN approach. The GP system that this chapter will use is a standard GP system.

6.1 Object Detection Method

The object detection method used is the same as the one used in the previous chapter where we use pixel statistics as features and we use Rectilinear Regions to specify where we gather the pixel statistics. The differences in this experiment will be the use of a GP system rather than an NN system to train the classifier template for sweeping. The differences are discussed in detail as follows:

6.1.1 The Evolutionary Process

The GP system uses the idea of evolution to achieve its goal of finding a good classifier. The GP system does this by first initialising a population consisting of individual programs with random characteristics. Using this initial population the GP system will ″mate″ the individual programs by applying the genetic operators: crossover, mutation and reproduction as discussed in chapter 2. The GP system will create a new population consisting of the new individual programs and will repeat the process until a termination criteria is met. See figure 6.1.

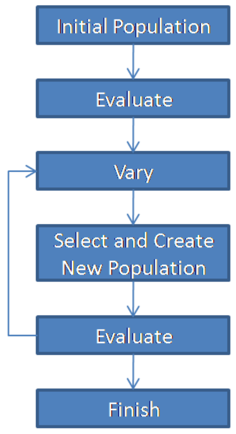


Figure 6.1: The Evolutionary Process

6.1.2 The Terminal and Function Set

The terminals are the leaf nodes of the individual program tree as discussed in chapter 2 about "Program Representation". In this experiment the terminal set consists of the 12 pixel statistic features from the Rectilinear Regions and a random value between [-1,1]. The 12 pixel statistic features are the same ones we used in chapter 5 where it consist of six means and 6 standard deviations.

The function set are the non-left nodes of the individual program tree as discussed in chapter 2 about "Program Representation". In this experiment the function set consist of an add, subtract, multiply, protected division where if divided by 0 it will return 0 and the conditional statement known as "if".

The terminal and function set are summarised as follows:

- **Function Set:** {+, -, *, % (protected division), if}
- **Terminal Set:** The 12 pixel statistics and a constant value that is randomly generated between the range of [-1, 1] upon initialisation of the program.

6.1.3 Fitness Function

The experiment uses a fitness function that measures the accuracy of the program in its ability to correctly classify each cut out image in the training set. This implies that a fitness function of 1 is the fitness where the program has correctly classify all images cut outs in the training set and 0 if the programs fails to correctly classify any. When the optimal program is found then it is briefly tested with the test set to evaluate its performance. This is to indicate whether there is overtraining.

6.1.4 Parameters and Termination Criteria

A GP system requires specific parameters before it can begin so that the evolutionary process is aware of the evolutionary conditions. The parameters used for the evolutionary process are as follows:

- **Elite Rate:** 10%
- **Crossover Rate:** 60%
- **Mutation Rate:** 30%

- **Population Size:** 500 for Trivial and Medium, 3330 for Easy and 3000 for Hard
- **Initial Minimum Tree Depth:** 3
- **Initial Maximum Tree Depth:** 5
- **Minimum Tree Depth:** 3
- **Maximum Tree Depth:** 6
- **Maximum Generations:** 200

During the evolutionary process, each individual program produces an output value that determines whether the program believes the image cutout belongs to the object class or background class. The GP system will assume any output greater than 0 is an object class while anything below or equal to 0 is a background, this is known as the "Standard Binary Classification" strategy.

Finally the termination criteria is either when the GP system finds the best program with the ideal fitness where in this experiment is 1, or if the GP process reaches the maximum specified generations where it will report the best program with the best fitness at that point.

6.2 Chapter Goal

The goal of this chapter is to accomplish the third specific goal which was to develop a GP-based approach to object detection and investigate whether GP will address the limitations discovered in the previous approaches and whether object detection has improved using a GP-based approach.

6.3 Results

6.3.1 Object Classification Results

During GP process, the usual 500 population was large enough for a perfect solution to be found with the Trivial and Medium Shapes data sets. But for the Easy and Hard Faces data set, a population of about 3000 (3330 for Easy Shapes and 3000 for Hard Faces) was required to find the best program.

The GP results for object classification are presented in table 6.1. The tables shows ideal results for the first three data sets but there seems to be some evidence of possible overfitting for the hard data set with the need for the full 200 generations to find an optimal program compared to the shapes where 22 generations was the highest. This can indicate a huge difference in difficulty. Since the Trivial, Easy Shapes data and Medium set obtained ideal training and test classification accuracy, it is certain that GP does not have difficulty in obtaining ideal results on these problems where varying shapes do not hinder the performance of the classification. In the case of the Hard Faces data set, we did not get ideal results but we have near ideal results for the object classification as both training and test performances shows a result quite close to 100% but since the test result deviated from the training result of about 15%, this would indicate overtraining which could cause the performance during the sweeping process to degrade.

ImageDatabases	*Generations*	*TrainingAccuracy*	*TestAccuracy*
Trivial Shapes	0	100%	100%
Easy Shapes	1	100%	100%
Medium Shapes	22	100%	100%
Hard Faces	200	92.45%	77.36%

Table 6.1: Object classification results for the four databases (GP with Pixel Statistics)

6.3.2 Object Detection Results

The object detection results are shown in table 6.2. Once again these results are calculated with a tolerance of half the input window's size. Table 6.2 shows that the Trivial, Easy and Medium data sets obtain an ideal measure of 100% DR and 0% FAR. But for the Hard Face data set, the GP system interestingly enough obtained an ideal DR of 100% with a FAR of 250%. Considering the training and testing was not 100%, I expected a much higher FAR.

ImageDatabases	*Threshold*	*DetectionRate*(%)	*FalseAlarmRate*(%)
Trivial Shapes	76	100%	0%
Easy Shapes	165	100%	0%
Medium Shapes	0	100%	0%
Hard Faces	2300	100%	250%

Table 6.2: Object detection results for the four databases (GP with Pixel Statistics)

6.3.3 Analysis of Results — Object Sweeping Maps

The object sweeping-maps used in this experiment varies slightly from the ones that were used in the NN experiment. In the NN experiment each pixel in the sweeping-map represented the weight of the NN considering how likely that pixel was an object of interest. GP does not have such weighting and relies on the threshold of the GP program's output and therefore the sweeping-maps here have been altered to display white if the program generates an output that exceeds the user-defined threshold and black if the program generates an output below the threshold. This is just implementing the standard binary classification strategy.

According to the sweeping maps, we find that that the Trivial, Easy Shapes data and Medium set there were no problems for the GP to work out if a position in the image contain an object or not. By examining the object sweeping maps in figures 6.2, 6.3 and 6.4 we found the GP program is able to detect the objects in the Trivial, Easy Shapes and Medium data sets pretty clearly.

In the case of the Hard Faces data set, As we look at the object sweeping map in figure 6.5, it is not easy to determine whether the objects are detected by the sweeping process or not but it is clear that quite a number of the background were considered objects of interest during the sweep. This may indicate that the overtraining is possibly referring to the objects and there is still insufficient background examples used for the training as the object detection program was exposed to the varying backgrounds that it had not seen before.

Another interesting outcome was that the Easy Shapes data set required a much larger intial population than the Medium Shapes data set. It would seem the more objects that closely resemble the intensity level of the background the more harder the problem therefore a distorted or noisy background is actually considered an easier problem. This was also experienced during the NN experiments where local minimas were more likely with the Easy Shapes data set rather than Medium Shapes data set.

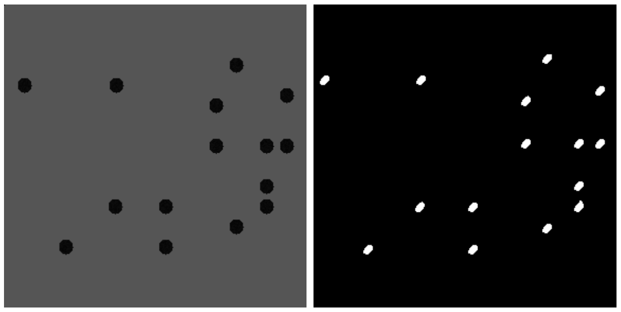


Figure 6.2: Trivial Shapes Original Image (pixel statistics), Objects-sweeping-map (pixel statistics)

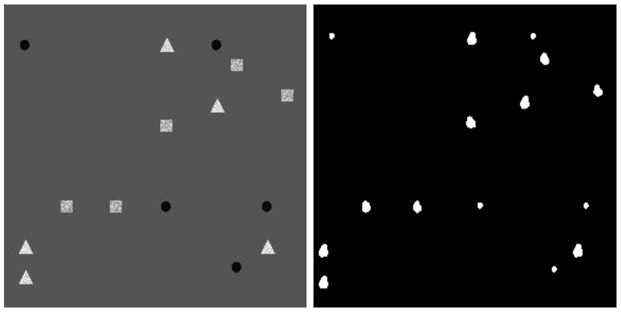


Figure 6.3: Easy Shapes Original Image (pixel statistics), Objects-sweeping-map (pixel statistics)

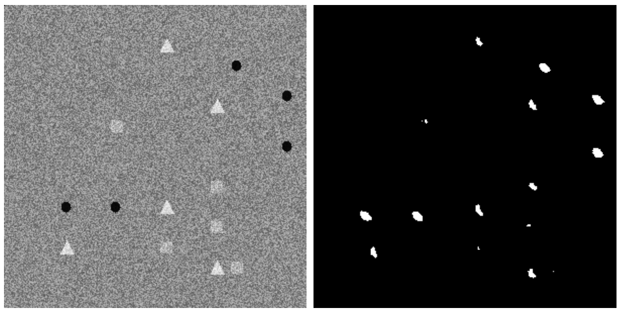


Figure 6.4: Medium Shapes Original Image (pixel statistics), Objects-sweeping-map (pixel statistics)

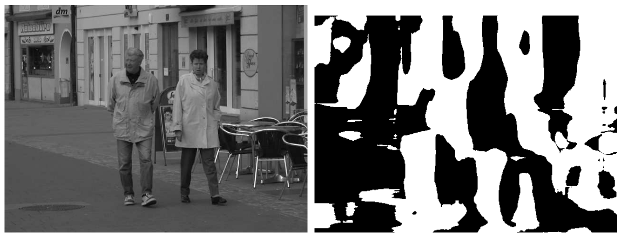


Figure 6.5: Hard Faces Original Image (pixel statistics), Objects-sweeping-map (pixel statistics)

6.4 Summary

This chapter presented an experiment of a standard GP object detection system that uses pixel statistics. Performance measurements were retrieved and results showed that the shapes data sets were much better than any of the NN approaches. Even moderately distorted backgrounds did not hinder the object classification or object localisation task. But as expected the face detection results on real photo images were worse than shapes but still much better than the NN approaches, this implies that the NN architecture was another limitation causing the poor performance. However the GP system is able to achieve these performance results because it takes a tremendous about of time and effort to find the best individual program and it still takes the same amount of time to actually sweep an image for the object detection process.

This experiment has hightlighted the following key factors that must be taken note when developing a more efficent object detection system:

1. GP seems to be superior in terms of finding a template with rules suited for the object detection than NN.

2. Difficult problems are likely to excessively require more background examples than objects.

3. Finding the right parameters seem to take longer than the time required for the GP process to find an optimal program.

4. Distorted objects or objects that nearly blend into the background will hinder the object detection process. This also seemed to be the case with the previous NN experiment with raw pixels. With higher level pixels the situation seemed to improved therefore perhaps more pixels statistics, specific features or varying regions can improve the situation?

To address the problem of where the GP system is still not able to perform well on the Hard Faces data set and the issue of slow sweeping. The decision to design an approach based on feature and region control is the best domain independent strategy in improving object detection in terms of effectiveness and efficiency. This feature and region control based approach will determine if better performance can be achieved with even fewer higher level information.

Chapter 7

GP-RegionRefiner with Pixel Statistics

This chapter outlines and describes the experiment conducted with the improved system using pixel statistic features which is named GP-RegionRefiner. The purpose of this experiment is to address the limitations and weaknesses of the GP system that was used in the previous chapter in terms of the inefficient object detection. Due to the superior performance of the GP system over the NN system, GP-RegionRefiner is based off the GP system.

7.1 Object Detection Method

The object detection method used in this experiment is the same as the one used in the previous experiment. GP-RegionRefiner is based of the GP system and therefore it also uses the evolutionary process. Therefore this approach is most similiar to the approach used in chapter 6 except we introduce a window that uses nine regions and another window that uses sixteen regions rather than the previously used Rectilinear Regions. The idea behind the new approach was inspired by the NN system where certain features that belong to certain regions that were not being used when the NN weight for that feature was 0; and the idea that a human being does not necessarily take into account of every little information to identify an object of interest. For example if we were to identify a face compared to a background then obviously a background does not contain eyes and hair where the hair is above the eyes. The differences are further discussed in detail as follows:

7.1.1 Features

The features used in this experiment are the same pixel statistics of the mean and standard deviation that were used in both chapters 5 and 6. However GP-RegionRefiner does not use all the features from every region and only uses the most important features discovered by the newly implemented ″Region Refinement″ component which will be discussed later.

7.1.2 Regions

The strength of GP-RegionRefiner is its ability to find and use only the useful regions and not waste precious computational resources or time on working with features that do not help. Therefore the original Rectilinear Regions where regions are overlapping is not necessarily a good example in testing this new approach due to overlapping information. In order to demonstrate that the idea of redundant regions, we introduce a ″Nine Regions″ window where there exist nine regions in a window where each region is approximately of the same size and a ″Sixteen Regions″ window where it is a window with sixteen approximately equal sized regions. It is noted that the full window itself is not a region to prevent overlapping information that will confuse whether a particular region of pixel information is useful or not. The new ”Nine Regions” is shown in figure 7.1 and the ”Sixteen Regions” is shown in figure 7.2.

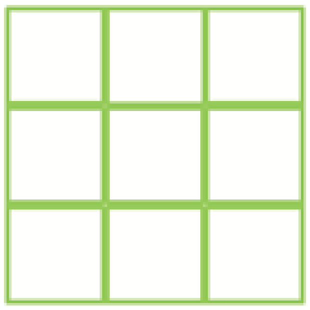


Figure 7.1: Nine Regions

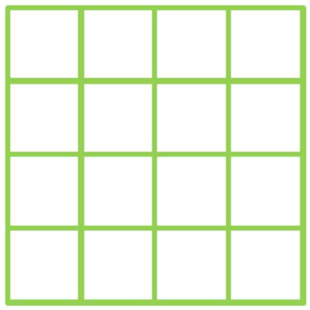


Figure 7.2: Sixteen Regions

7.1.3 Region Refinement and Feature Selection

In order for the system to use this idea, GP-RegionRefiner needs to collect information during the training process of what features are being used more than others when determining an object of interest so that GP-RegionRefiner can eliminate these regions from the sweeping process as well. This therefore will improve the efficiency of the actual object detection within a full image by a considerable amount as we eliminate the need to calculate unnecessary features or even regions for each point where we test against the template. Since we are moving pixel by pixel, this is a tremendous amount of work removed just by identifying the useful features or regions. GP-RegionRefiner accepts a ”maximum features” argument instead of a ”maximum regions” argument because wish to also identify whether the mean is more useful or if the standard deviation is useful. Using these useful features such as the top 10 features, we can determine what regions they belong to and this is enough to assume that these regions are more important.

The ”Region Refinement” component is designed to receive the parameter of the amount features allowed and to collect the statistics from the top individual programs on all features that can be used of each generation. Therefore this means all eighteen features are used initially in the training process. GP-RegionRefiner will continue to collect statistics until it meets its termination criteria where it will then restart the training process with only the top set of features. Therefore the result is a classifier template that is refined to only use important features in determining an object of interest. For example using figure 7.3, we see a window capture a face. To identify a face compared to a background then obviously a background does not contain eyes and hair where the hair is above the eyes. This therefore implies we only need the highlighted four regions which then effectively eliminates the 12 other windows consisting of 24 features to identify a face.

The sweeping process is then modified to only use the useful regions for object detection rather than using all regions like in the previous experiment. See figure 7.4 to see how the new "Region Refinement" component fits into the existing GP system.

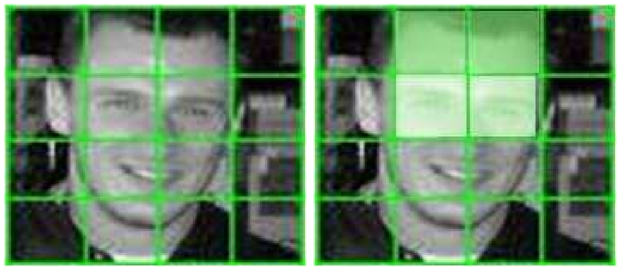


Figure 7.3: The face, The face with the necessary windows filled

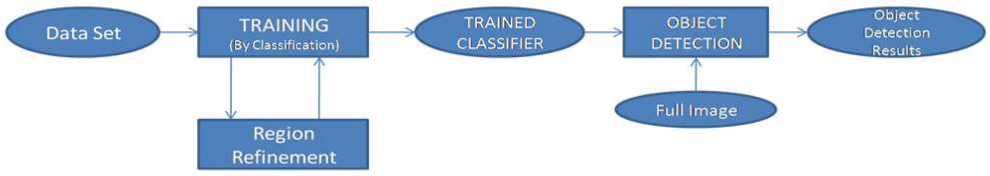


Figure 7.4: The New Object Detection Method

7.2 Chapter Goal

The goal of this chapter is to accomplish the fourth specific goal which was to develop an improved approach that addresses the limitations from the previous approaches which will also achieve a much higher object detection performance on all data sets by using the information gained from the previous approaches.

7.3 Experimental Setup

The expermental setup in terms of the parameters, fitness function and function set are much the same as the GP approach. But for the terminal set there is a slight variation where all 18 features (Nine Regions) or 32 features (Sixteen Regions) are used but only 10 features will be used in the actual object detection. These 10 features will determine the regions to be used. For the maximum generations, we have a maximum generations for the initial training where the statistics are collected and then a maximum generations for the final training where we train a classifier or find a genetic program that can classifier the objects from the training data set well. Therefore besides the two phase training process, the termination process is still the same where either if the best program is found or the maximum amount of generations have been reached. The following is a list of the parameters used for the experiment.

- **Elite Rate:** 10%
- **Crossover Rate:** 60%
- **Mutation Rate:** 30%
- **Population Size:** To be determined
- **Initial Minimum Tree Depth:** 3
- **Initial Maximum Tree Depth:** 5
- **Minimum Tree Depth:** 3
- **Maximum Tree Depth:** 6
- **Maximum Generations:** 50 for initial training and 50 for final training for Nine Regions, 50 for initial training and 200 for final training for Sixteen Regions
- **Function Set:** {+, -, *, % (protected division), if}
- **Terminal Set:** 18 pixel statistics for Nine Regions and 32 for Sixteen Regions but the template will only use the top 10 pixel statistics. A constant value that is randomly generated between the range of [-1, 1] upon initialisation of the program is also used.

7.4 Results

7.4.1 Object Classification Results

During GP process, the usual 500 population was enough for a better solution to be found for all data sets, it seems increasing the population would either be not necessary as the problem was too easy or it would encourage more overfitting if the problem was too difficult. However this was an exception for the Hard Faces data set when applying the sixteen regions window where we used a population of 3300.

The GP-RegionRefiner's results for object classification are presented in table 7.1. The tables shows ideal results for the Trivial Shapes data set. The GP-RegionRefiner seem to obtain almost ideal results for the Easy Shapes data set and Medium Shapes data set where the Easy Shape's training was off by 10% but it did achieve 100% for testing so this is not an issue. In the case of the Hard Faces data set, we still have overfitting. Therefore performance was not very much different from the GP with Pixel Statistics experiment which implies that the GP-RegionRefiner can perform just as well as the previous GP approach in classifying the objects of interest from the background. Although it is worth mentioning that when the GP-RegionRefiner was in its training step; it was found that a maximum generation setting of 200 provided worse object detection results than a maximum generation setting of 50. This leads to the likihood that the GP-RegionRefiner is able to achieve results of the GP base system at 200 generations at its eqivalent of 50 generations. Implying GP-RegionRefiner is in fact better as well as fast at training. We further investigate these claims with the "Sixteen Regions" window by referring to table 7.2, and found that the results were closely the same, except for the Hard Faces data set where there was better results with 200 generations.

ImageDatabases	*Generations*	*TrainingAccuracy*	*TestAccuracy*
Trivial Shapes	0	100%	100%
Easy Shapes	50	90.67%	100%
Medium Shapes	39	100%	99.33%
Hard Faces	50	83.02%	56.60%

Table 7.1: Object classification results for the four databases using Nine Regions (GP-RegionRefiner with Pixel Statistics)

ImageDatabases	*Generations*	*TrainingAccuracy*	*TestAccuracy*
Trivial Shapes	0	100%	100%
Easy Shapes	50	90.67%	100%
Medium Shapes	39	100%	98%
Hard Faces	200	100%	52.83%

Table 7.2: Object classification results for the four databases using Sixteen Regions (GP-RegionRefiner with Pixel Statistics)

7.4.2 Object Detection Results

The object detection results are shown in table 7.3. These results are once again calculated with a tolerance of half the input window's size. Therefore with the shapes data sets the tolerance is 7 pixels which means anything within 7 pixels of the correct centre point is considered a match while for the face data set it is a tolerance of 40 pixels. Table 7.3 shows that only Trivial achieved ideal results with the Easy data set being close to ideal. However the results for the Medium data sets did not seem to achieve good results. Finally for the Hard data set, the GP-RegionRefiner achieved quite a high FAR in order to have a 100% DR. The object detection results therefore shows all but the Trivial data set showed poorer performance than the GP base system. This would first indicate that the GP-RegionRefiner failed to maintain the performance of the GP base system when achieving more efficient object detection; or that the simple counting of the features being used during training is not enough for detecting redundant regions. If we were to examine the results for the sixteen regions, we find that the performance have improved tremendously and it seems that with more regions GP-RegionRefiner can be perform better than even the original GP system. Importantly, we find GP-RegionRefiner's performance for the Hard Face data set is superior compared to the original GP system's performance in table **??** in terms of DR and FAR.

ImageDatabases	*Threshold*	*DetectionRate*(%)	*FalseAlarmRate*(%)
Trivial Shapes	0.05	100%	0%
Easy Shapes	1.30	100%	33.33%
Medium Shapes	0.02	46.67%	53.33%
Hard Faces	0.59	100%	700%

Table 7.3: Object detection results for the four databases using Nine Regions (GP-RegionRefiner with Pixel Statistics)

ImageDatabases	*Threshold*	*DetectionRate*(%)	*FalseAlarmRate*(%)
Trivial Shapes	15	100%	0%
Easy Shapes	0.04	100%	0%
Medium Shapes	7000	93.33%	86.67%
Hard Faces	230	100%	50%

Table 7.4: Object detection results for the four databases using Sixteen Regions (GP-RegionRefiner with Pixel Statistics)

7.4.3 Analysis of Results — Object Sweeping Maps

By examining the object sweeping maps in figures 7.5, 7.6 and 7.7 it was found the GP-RegionRefiner is able to detect the objects in the Trivial and Easy Shapes quite clearly where Easy Shape's 33% FAR is due to the noise within the shapes. For the Medium data sets, the sweeping map shows that the system cannot differentiate the noisy shapes from a noisy background. But for the Hard Face data set we see clear and precise points of object positions rather than large parts being white. It seems the GP-RegionRefiner has the ability to deal with a complex situation in terms of varying backgrounds. Further examination with the "Sixteen Regions" sweeping maps, we found the performance to be relatively the same with the interesting exceptions of Medium Shapes and Hard Faces. Looking at figure 7.11, we see with more regions, GP-RegionRefiner is able to produce a better classifier that is not as easily confused in determining objects of interest from a noisy background for the Medium Shapes data set. For the Hard Faces data set, it can seen in figure 7.12 that GP-RegionRefiner is able to produce an even more precised classifier than even the original GP system.

If compared to the sweeping maps in chapter 6, the sweeping maps in here shows that the two systems have about the same degree of certainty when detecting objects in a full image except that GP-RegionRefiner is more certain with the Hard data set than the GP base system. Therefore this really does imply that GP-RegionRefiner is actually superior in terms of actual object detection on a full image.

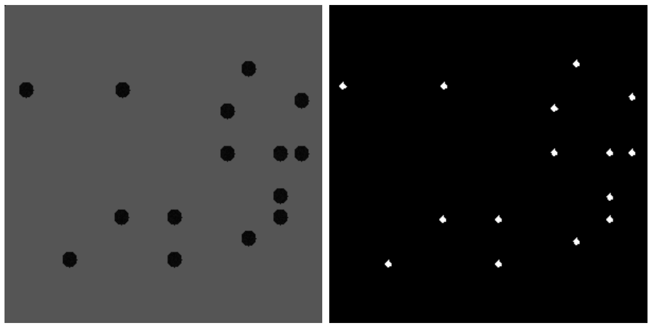


Figure 7.5: Nine Regions: Trivial Shapes Original Image (pixel statistics), Objects-sweeping-map (pixel statistics)

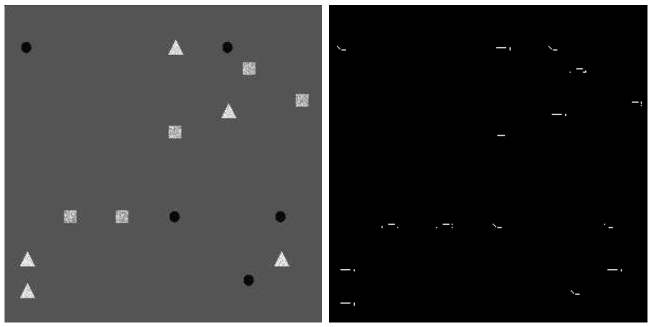


Figure 7.6: Nine Regions: Easy Shapes Original Image (pixel statistics), Objects-sweeping-map (pixel statistics)

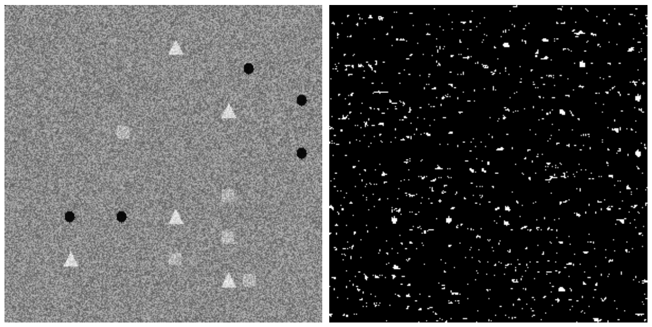


Figure 7.7: Nine Regions: Medium Shapes Original Image (pixel statistics), Objects-sweeping-map (pixel statistics)

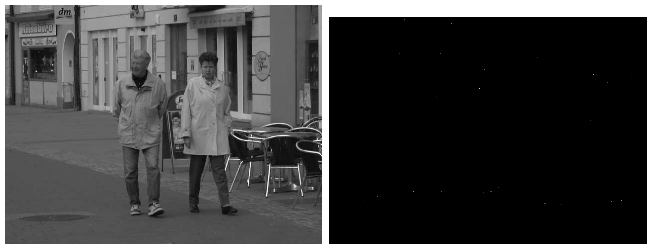


Figure 7.8: Nine Regions: Hard Faces Original Image (pixel statistics), Objects-sweeping-map (pixel statistics)

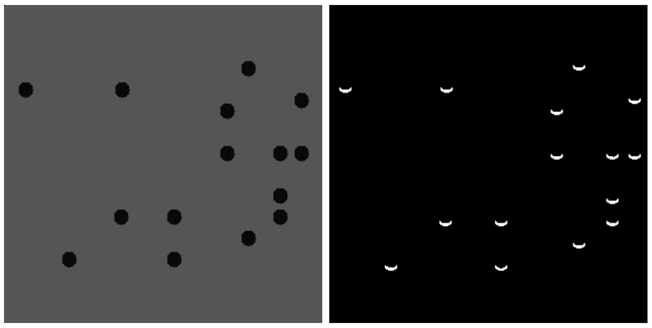


Figure 7.9: Sixteen Regions: Trivial Shapes Original Image (pixel statistics), Objects-sweeping-map (pixel statistics)

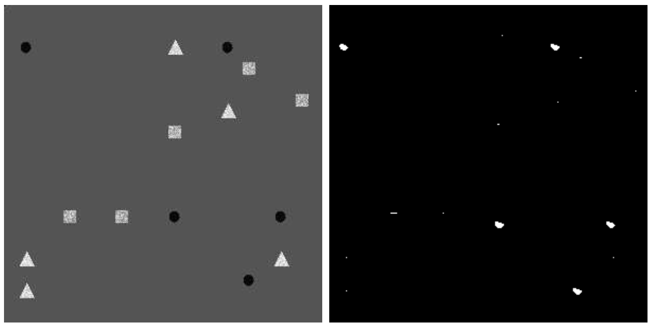


Figure 7.10: Sixteen Regions: Easy Shapes Original Image (pixel statistics), Objects-sweeping-map (pixel statistics)

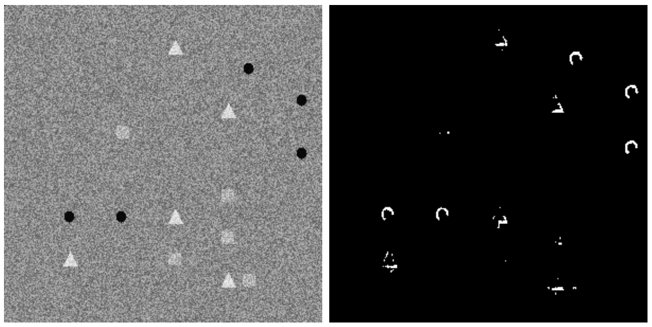


Figure 7.11: Sixteen Regions: Medium Shapes Original Image (pixel statistics), Objects-sweeping-map (pixel statistics)

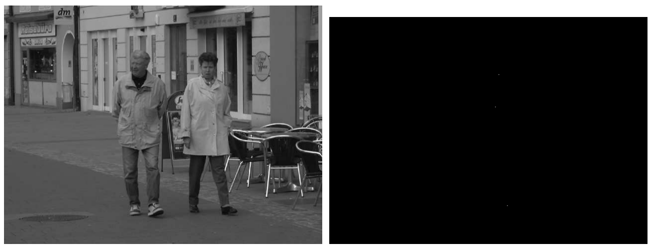


Figure 7.12: Sixteen Regions: Hard Faces Original Image (pixel statistics), Objects-sweeping-map (pixel statistics)

7.5 Summary

This chapter presented an experiment of the GP-RegionRefiner which has a more efficient object detection method than it's GP base counterpart. The GP-RegionRefiner utilised a domain independent based approach to a one object class object detection problem. Performance measurements were retrieved where the classification results were about the same as the GP base system and better than the NN base systems; object detection results initially seemed poorer than the GP base system but the sweeping maps showed that the GP-RegionRefiner could actually be better at the actual object detection on a full image. This claim was further reinforced when we examined the results from the "Sixteen Regions" which showed more promising results, implying that if there were more regions available for GP-RegionRefiner to choose from then, it would be able to produce even better results. However I still believe it is possible to improve the statistics collection system to further refine the sweeping process. I believe this is true as it seems overfitting occurs much earlier with the GP-RegionRefiner than the GP base system in terms of the Hard data set, indicating that the learning is more effective and should perform either the same or better than the GP base system. This is evident with the results from the Sixteen Regions where for the Hard data set, we found a even better object detection performance at 200 generations compared to the GP base system.

This experiment has hightlighted the following key factors:

1. GP-RegionRefiner is superior to the NN base system but marginally worse than the GP base system when given limited regions to refine from.

2. Sweeping maps show that the GP-RegionRefiner can handle varying complex backgrounds such as the ones in the Hard Faces data set but cannot handle random noisy backgrounds such as the ones in the Medium Shapes data set.

3. Identifying redundant information in terms of regions or features and applying the information to the sweeping process can definitely make the object detection process more efficient.

4. Introducing windows with many regions can improve the GP-RegionRefiner's performance quite a considerable amount.

Chapter 8

Conclusions

This chapter presents the main conclusion that will answer the specified research goals, then it will specify indirect findings and the chapter will end by discussing the possible future work that this project will lead to.

8.1 Main Conclusions

The goal of this project is to develop a new approach that can achieve better object detection performance in terms of effectiveness and efficiency to single-class object detection problems of increasing difficulty that is domain independent. The project has explored and examined current standard or base systems and determined their strengths and weaknesses in order to develop a new approach. During the development and experiments of this project, the following were discovered:

1. The initial NN-based approach to object detection established poor performance due a high requirement of training examples. This was confirmed with the concept of PAC Learning theory and VC dimensions where the required examples were at least 3,528 training examples for the trivial problem. This large requirement was due to the large NN architecture consisting of at least 1176 nodes for the trivial problem which indicated our limitation was in the features we were using as it constrained the performance to require far more examples in order to obtain an improvement. However although it is quite possible that databases are likely to contain many examples for training, there are such database that only have few quantities of data and we run the risk of overtraining with all possible training data. The aim here was clearly to address the problem with the use of raw pixel features.

2. Introducing pixel statistics solved the feature problem in the initial NN-based approach as it reduced the NN architecture size tremendously which in accordance to PAC learning theory and VC dimensions would reduce the amount of training examples required. However performance was not satisfactory especially in terms of the hard data set and it is at this point it was determined that the NN architecture itself was a limitation for high performance in terms of effectiveness. The decision to replace the NN architecture with GP was considered.

3. The GP system showed very effective performance results with very high DR results and low FAR results. The experiment proved that GP is superior to NN in terms of object detection with our data sets. But there were still problems with the hard data set and its ability to efficiently perform the object localisation task. Although GP did achieve better results, it did not improve in terms of efficiency. The experiment pointed

out that the genetic programs only used a subset of the total available features which implied that resources were wasted on calculating useless features or training using useless regions during the object detection process.

4. The new GP-RegionRefiner showed very promising results in its more efficient design. GP-RegionRefiner was able to identify the regions that were useful by collecting feature statistics. Then it was able to use these useful regions from the feature statistics to train a classifier that could achieve performance that was slightly more effective. Although the DR and FAR were quite even, GP-RegionRefiner did perform better than the GP system on the hard data set with a more precised sweeping map.

Throughout the many experiments, the development of GP-RegionRefiner was quite successful but it did not perform well on the medium data set where there was a lot of noise involved with the background. I believe this was caused by a limitation of time rather than a limitation of the system where time was not invested in determining how best to discover what regions are considered useful. However this was out of the scope of this project and the goal was to make the object detection process more efficient and effective which was achieved and reinforced by the use of two different types of region windows in the final experiment.

8.2 Other Findings

A few additional indirect findings were discovered during the development of this project. They are described as follows:

- Mini experiments were conducted in addressing the lack of background examples problem as mentioned in the summary in some of the experiments. To examine this problem, variations of the hard data set were made consisting of inbalance data sets. It was found that peformance did not increase unless excessive amounts of background examples were included. See table A.1 for the results. It is noted that the term "new" means the new fitness function where the experiment required a new fitness function to handle inbalance data sets. The variations are as follows, hard(75) consist of 75 background examples to 53 object examples, hard(106) consist of 106 background examples to 53 object examples and finally hard(159) consist of 159 background to 53 object examples.

- The other indirect findings which could not be include due to colour code diagrams was that it was found standard deviation features were more popular than mean features when the difficulty of the data set increased. This questions whether the introduction of more features could have improved performance or region refinement.

8.3 Future Work

This section briefly discusses a few interesting research questions that could not be found in this project due to time constraints. However the following are a list of interesting areas to expand from this project:

1. GP-RegionRefiner uses basic statistics to count the features used in determine what regions are useful and what are not. Is this the best way to do this?

2. It was found that the standard deviation feature was used more often than the mean as the problem got harder. What will happen to the performance of the GP-RegionRefiner system if we include other features such as the median?

3. Function sets and terminal sets were not modified and only the default set that VGP provied were used? What would happen if additional functions or terminals were used with GP-RegionRefiner?

4. The use of more background examples was not explored in relation to the GP-RegionRefinement system. Would including more background examples improve the RegionRefinement system?

Bibliography

[1] The pascal object recognition database collection. http://www.pascal-network.org/challenges/VOC/databases.html, April 2007.

[2] BEN-YACOUB, S., FASEL, B., AND LUTTIN, J. Fast face detection using mlp and fft. Tech. rep., IDIAP, 1999.

[3] GARCIA, C., AND DELAKIS, M. A neural architecture for fast and robust face detection. Tech. rep., University of Crete, Department of Computer Science, 2002.

[4] HOWARD, D., ROBERTS, S., AND BRANKIN, R. Target detection in imagery by genetic programming. *Advances in Engineering Software 30* (1999), 303–311.

[5] KOZA, J. R. *Genetic Programming: On the Programming of Computers by Means of Natural Selection (Complex Adaptive Systems).* The MIT Press, December 1992.

[6] LIN, Y.-Y., LIU, T.-L., AND FUH, C.-S. Fast object detection with occlusions. Tech. rep., Institute of Information Science, Academia Sinica and National Taiwan University, Department of CSIE, 2004.

[7] NAZEER, ., OMAR, ., JUMARI, ., AND KHALID, . Face detecting using artificial neural network approach. *ams 00* (2007), 394–399.

[8] ROBERTS, M. E., AND CLARIDGE, E. Cooperative coevolution of image feature construction and object detection. Tech. rep., University of Birmingham, School of Computer Science, 2004.

[9] ROBERTS, M. E., AND CLARIDGE, E. A multistage approach to cooperatively coevolving feature construction and object detection. Tech. rep., University of Birmingham, School of Computer Science, 2005.

[10] ROWLEY, H. A., BALUJA, S., AND KANADE, T. Human face detection in visual scenes. Tech. rep., Carnegie Mellon University, 1995.

[11] ROWLEY, H. A., BALUJA, S., AND KANADE, T. Rotation invariant neural network-based face detection. Tech. rep., Carnegie Mellon University, School of Computer Science, 1997.

[12] ROWLEY, H. A., BALUJA, S., AND KANADE, T. Neural network-based face detection. *IEEE Transaction PAMI, 20(1)* (1998), 23–28.

[13] RUMELHART, D. E., AND MCCLELLAND, J. L. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition : Foundations (Parallel Distributed Processing).* MIT Press, August 1986.

[14] SONG, A., AND CIESIELSKI, V. Fast texture segmentation using genetic programming. Tech. rep., RMIT University, School of Computer Science and Information Technology, 2003.

[15] VALIANT, L. G. A theory of the learnable. *Communications of the ACM* (1984), 1134–1142.

[16] VAPNIK, V. N. *The Nature of Statistical Learning Theory (Information Science and Statistics).* Springer, November 1999.

[17] VIOLA, P., AND JONES, M. Rapid object detection using a boosted cascade of simple features. Tech. rep., Mitsubishi Electric Research Labs and Compaq CRL, 2001.

[18] VIOLA, P., AND JONES, M. J. Robust real-time face detection. *Int. J. Comput. Vision 57*, 2 (2004), 137–154.

[19] WINKELER, J. F., AND B.S.MANJUNATH. Genetic programming for object detection. Tech. rep., University of California at Santa Barbara, Electrical and Computer Engineering Department, 1997.

[20] ZHANG, M. Mining small objects in large images using neural networks. Tech. rep., Victoria University of Wellington, School of Mathematical and Computing Sciences, 2005.

[21] ZHANG, M., ANDREAE, P., AND BHOWAN. A two phase genetic programming approach to object detection. Tech. rep., Victoria University of Wellington, School of Mathematical and Computing Sciences, 2004.

[22] ZHANG, M., ANDREAE, P., AND PRITCHARD, M. Pixel statistics and false alarm area in genetic programming for object detection. Tech. rep., Victoria University of Wellington, School of Mathematical and Computing Sciences, 2002.

[23] ZHANG, M., AND CIESIELSKI, V. Neural networks and genetic algorithms for domain independent multiclass object detection. Tech. rep., Victoria University of Wellington, School of Mathematical and Computing Sciences, 2004.

[24] ZHANG, M., CIESIELSKI, V. B., AND ANDREAE, P. A domain-independent window approach to multiclass object detection using genetic programming. Tech. rep., Victoria University of Wellington, School of Mathematical and Computing Sciences, 2003.

[25] ZHANG, M., AND NY, B. False alarm filters in neural networks for multiclass object detection. Tech. rep., Victoria University of Wellington, School of Mathematical and Computing Sciences, 2004.

Appendix A

Other Findings

A.1 Additional Background Example Results

ImageDatabases	*TrainingAccuracy*	*TestingAccuracy*	*DR*	*FAR*
Hard (53) (old)	92.45 %	77.36%	100%	250%
Hard (53) (new)	94.44 %	66.10%	100%	550%
Hard (75) (new)	93.59%	72.17%	100%	400%
Hard (106) (new)	99.04%	69.45%	100%	400%
Hard (159) (new)	92.95%	61.48%	100%	150%

Table A.1: GP standard system mini-experiment of using additional background examples)