The Role of Symmetry Features in Connectionist Pattern Recognition

A thesis submitted to the University of Manchester for the degree of Doctor of Philosophy in the Faculty of Engineering and Physical Sciences.

2012

Sam Holland

School of Computer Science
# Contents

1 Introduction

1.1 Research Context ........................................ 18
1.2 Research Problem ........................................ 19
1.3 Relevance of Symmetry ................................... 20
1.4 Overview .................................................. 20
1.5 Chapter Summary .......................................... 22

2 Symmetry

2.1 Introduction ............................................. 24
2.2 Reflectional Symmetry .................................... 25
2.3 Rotational Symmetry ...................................... 27
2.4 Symmetry in Nature ...................................... 27
2.5 Computational Use of Symmetry ......................... 29
2.6 Measuring Symmetry ..................................... 30
   2.6.1 Qualitative Symmetry Transform ................. 31
   2.6.2 Generalised Symmetry Transform ............... 40
   2.6.3 Discrete Symmetry Transform .................. 47
   2.6.4 Fast Radial Symmetry Transform ............... 53
2.7 Taxonomy ................................................ 58
2.8 Contribution ............................................. 61

3 Neural Networks

3.1 Introduction ............................................ 63
3.2 Usage .................................................... 64
3.3 Biological Neuron ....................................... 65
3.4 Artificial Neuron: Threshold Logic Unit .......... 66
3.5 The Perceptron ......................................... 67
3.6 Multi-Layer Perceptron ................................ 69
3.7 Popular Networks ....................................... 72
   3.7.1 Hopfield ........................................... 72
   3.7.2 Self-Organizing Map (SOM) / Kohonen ....... 76
3.7.3 Radial Basis Function (RBF) Network ....... 80
3.7.4 Probabilistic Neural Network ............... 83
3.8 Contribution ..................................... 88

4 Classification of Symmetry Features ....... 89
  4.1 Introduction .................................... 89
  4.2 Motivation .................................... 90
  4.3 Hypothesis .................................... 90
  4.4 Rationale .................................... 91
  4.5 Related Work .................................. 92
  4.6 Classification Problem ....................... 95
  4.7 Symmetry Features ............................ 95
  4.8 Experiment .................................... 97
    4.8.1 Symmetry .................................... 97
    4.8.2 Classifier .................................... 98
    4.8.3 Training and Test Patterns ............... 99
    4.8.4 Parameters .................................. 100
    4.8.5 Normalisation .............................. 101
  4.9 Results ....................................... 102
  4.10 Discussion .................................... 103
  4.11 Contribution .................................. 109

5 Position Invariance .......................... 110
  5.1 Introduction .................................. 110
  5.2 Hypothesis .................................... 110
  5.3 Motivation .................................... 111
  5.4 Rationale .................................... 111
  5.5 Position Invariance for Pattern Recognition ...... 114
  5.6 Transformation of Image ........................ 116
  5.7 Focal Point .................................... 118
  5.8 Experiment .................................... 119
    5.8.1 Patterns .................................... 120
    5.8.2 Symmetry Transform ....................... 120
    5.8.3 Position Normalisation Method .......... 121
    5.8.4 Intensity Normalisation ................. 121
    5.8.5 Classifier .................................. 121
    5.8.6 Performance Measurement .................. 122
  5.9 Result ......................................... 123
  5.10 Discussion ..................................... 123
  5.11 Contribution .................................. 127
8.5.2 Symmetry Orientation ........................................ 192
8.6 Visual Verification .................................................. 192
8.7 Experiment .......................................................... 196
  8.7.1 Hypothesis ...................................................... 196
  8.7.2 Classifier ........................................................ 196
  8.7.3 Patterns ........................................................ 197
  8.7.4 Symmetry Transform ........................................... 197
8.8 Results ............................................................. 198
8.9 Discussion ........................................................ 199
8.10 Contribution ....................................................... 207

9 Discussion .......................................................... 209
  9.1 Development of Ideas ............................................. 209
  9.2 Taxonomy of GST-based Symmetry Transforms ............... 211
  9.3 Contribution ....................................................... 213
    9.3.1 Published Work .............................................. 215
  9.4 Conclusions ....................................................... 215
    9.4.1 Research Aims and Objectives ............................. 216
    9.4.2 New Knowledge ............................................ 216
  9.5 Further Work .................................................... 218

References .......................................................... 220

A Published Literature ................................................ 234
List of Figures

2.1 Basic geometric shapes with their respective planes by which they are invariant to reflection. Specifically, these are discrete, global reflection planes. ........................................ 26
2.2 Huebner’s Qualitative Symmetry Transform showing the vertical symmetry of a 170 x 128 pixel image, with $m = 20$. Brightness corresponds to symmetry. .................. 33
2.3 Huebner’s Qualitative Symmetry Transform showing the fading border associated with values of $m$. .................. 37
2.4 Result of modifying the size of the operator mask, $m$, on the symmetries detected. .................. 38
2.5 GST: Pixels $p_i$ and $p_j$, each associated with a vector $(r, \theta)$, contribute to the symmetry of a pixel $p$ mid-way between them. 41
2.6 Normalised isomorphic symmetry map (magnitude) of transforms with different distance-weight settings. .................. 45
2.7 An example kernel at a point $(i, j)$ and the relationship of two points, $(p, q)$ and $(l, m)$ meeting the radius criteria. ............ 48
2.8 The Discrete Symmetry Transform: for an input image, $I$, the DST is computed from an non-uniformity measure, $\hat{E}$ and distance from perfect rotational symmetry, $T$. .................. 51
2.9 The output of the Discrete Symmetry Transform for varying radii, $r$, and number of axial moments, $n$. .................. 52
2.10 The pixel $p$ affects two others: one positively, one negatively at a distance $n$ that the gradient points to and away from respectively. .................. 55
2.11 The output of the Fast Radial Symmetry Transform on the image from Figure 2.2a for various radius $n$ in the set of radii $N$. In each case, the radial strictness parameter, $\alpha = 3$. .... 57
2.12 The output of the Fast Radial Symmetry Transform on the image from Figure 2.2a for various radius \( n \) in the set of radii \( N \). In each case, the radial strictness parameter, \( \alpha = 1 \). At this setting, the FRST performs the role of a bilateral symmetry transform.

3.1 Biological Neuron.

3.2 Basic artificial neuron (the McCulloch and Pitts model - Threshold Logic Unit (TLU)). The sum of the weighted (w) inputs (x) is passed through an activation function yielding the output (y).

3.3 The Perceptron.

3.4 A Multi-Layer Perceptron.

3.5 Three node (i, j and k) recurrent associative Hopfield network.

3.6 States of a Hopfield Network performing pattern recognition for the letter “L”.

3.7 Self-Organizing Layer and Neighbourhood Schemes with Distances of 1, 2 and 3 nodes displayed from the neighbourhood’s central node (shown in black) from Figure 8.13 of [25].

3.8 The lateral connections depicted as inhibitory (-) and excitatory (+) described as “On Centre off surround” by Gurney in Figure 8.2 of [25]. Kohonen SOFMs do not have these lateral connections.

3.9 The structure of the Radial Basis Function Neural Network. ‘g’ denotes that the activation-output functions of the hidden layer are Gaussian based. The nodes of the output layer are linear.

3.10 Structure of a Probabilistic Neural Network.

4.1 Overview of the proposed pattern recognition system structure used in the experiment.

4.2 Graph showing the relationship between the PNN smoothing parameter and the comparative generalisation performances the traditional feature recognition, and symmetry feature recognition using magnitude and orientation features separately; and both classifying both symmetry feature types. Generalisation performance is measured as the recognition rate on the test set of patterns. The traditional method outperforms the classification of symmetry features. The magnitude features outperform the orientation features, but performs worse than using both magnitude and orientation features.
4.3 Comparison of pattern recognition using normalised versus non-normalised symmetry magnitude features. The peak performances are identical for both the training and the test sets. Normalising the inputs of the patterns reduces the distances between the multi-variate Gaussian distributions in pattern space.

5.1 Output of a common method for spatial normalisation prior to classification for four characters sharing two common features or subcomponents. Each character can be distinguished by the relative positions of the sub components.

5.2 Potential alignment of characters by normalising their position based on resizing the object to fill the retina.

5.3 Potential alignment of characters by normalising their position based on the location of the point with the greatest symmetry magnitude.

5.4 Overview of the proposed pattern recognition system structure to achieve position invariance using symmetry, as used in the experiment.

5.5 Graph showing the effect of shifting the object position with respect to the recognition rate. As the position offset distance increases, the object moves diagonally across (and away from) the centre of the pattern. Beyond 8 pixels, the object is occluded by the boundary of the pattern.

6.1 Results of the orientation normalisation applied to a digit representing the number one at four different input orientations. Although the four outputs look similar, Figure 6.1h differs by 180 degrees.

6.2 Results of the orientation normalisation applied to a digit representing the number five at four different input orientations. There are almost two distinct output orientations but they are not identical.

6.3 Results of the orientation normalisation applied to a digit representing the number seven at four different input orientations. The transform has produced two distinct output orientations. Figures 6.3b and 6.3f share a common orientation, as do Figures 6.3d and 6.3h.
6.4 Results of the orientation normalisation (using the alternative GST definition) applied to a digit representing the number one at four different input orientations. Although there appears to be two distinct output orientations, closer visual inspection of the digit’s curve demonstrates that there are four different output orientations. .................................................. 139

6.5 Results of the orientation normalisation applied (using the alternative GST definition) to a digit representing the number five at four different input orientations. There is no common output orientation in this case. ................................. 140

6.6 Results of the orientation normalisation (using the alternative GST definition) applied to a digit representing the number seven at four different input orientations. There are no common output orientations. ................................. 141

6.7 Visual depiction of calculating the average of two angles using the continuous variate arithmetic mean. Two different input angle combinations are shown to share the same average. This behaviour is not desired. ................................. 142

6.8 Visual depiction of calculating the average of two angles using the circular variate mean. .................................................. 143

6.9 Results of the orientation normalisation (using circular variate mean definition) applied to a digit representing the number one at four different input orientations. There is a single, common output orientation. ................................. 144

6.10 Results of the orientation normalisation (using circular variate mean definition) applied to a digit representing the number five at four different input orientations. There is a single, common output orientation. ................................. 145

6.11 Results of the orientation normalisation (using circular variate mean definition) applied to a digit representing the number seven at four different input orientations. There is a single, common output orientation. ................................. 146

6.12 Overview of the proposed pattern recognition system structure to achieve position invariance using symmetry, as used in the experiment. .................................................. 147

6.13 The use of the added padding. The squares with the dotted lines represent the original pattern before and after rotation by 45 degrees. The padding prevents useful information in the pattern from exceeding the boundaries. Without this, pattern information may be lost. .................................................. 148
6.14 One valid pattern may become another valid pattern under rotation, such as the digits six and nine demonstrated here. In such scenarios, normalising orientation will not aid classifiers.

7.1 The Generalised Symmetry Transform (GST) Phase Function, $P(i, j)$. The two horizontal axes represent the orientations of points $i$ and $j$ respectively. Asymmetrical combinations yield zero. Symmetrical combinations yield positive output.

7.2 The Noise Tolerant Generalised Symmetry Transform (NT-GST) Phase Function, $P(i, j)$. The two horizontal axes represent the orientations of points $i$ and $j$ respectively. Like the GST, asymmetrical combinations yield zero. Two different symmetry combination types are defined: negative and positive function values when the difference between orientations $i$ and $j$ approaches positive $\pi$ or negative $\pi$ respectively.

7.3 The Generalised Asymmetry and Symmetry Transform (GAST) Phase Function, $P(i, j)$. The two horizontal axes represent the orientations of points $i$ and $j$ respectively. Asymmetrical pair combinations yield negative output. Symmetrical pair combinations yield positive output.

7.4 The Generalised Asymmetry Transform (GAT) Phase Function, $P(i, j)$. The two horizontal axes represent the orientations of points $i$ and $j$ respectively. Asymmetrical pair combinations yield positive output. Symmetrical pair combinations yield zero.

7.5 Overview of the proposed pattern recognition system structure used in the experiment.

7.6 The results of the feature test. The network input values were non-normalised. The recognition rate is provided per symmetry transform (GST, GAT and GAST) and by data set (training or test set).

7.7 The results of the feature test. The network input values were normalised. The recognition rate is provided per symmetry transform (GST, GAT and GAST) and by data set (training or test set).

8.1 The input and resulting output symmetry magnitude map after processing by the GST. Two reflectional symmetry planes are present, but there should be four.
8.2 The three relevant components associated with the two points used to determine the magnitude and direction of a potential reflection plane. These are colour intensity, \( I(p) \); distance to the reflection plane, \( d \); and internal angle, \( \psi \). Only points on non-uniform areas are considered. This example demonstrates two points on the edge of a shape.  

8.3 A comparison of the GST’s phase weight function with the RST’s symmetric phase function. In both sub-figures, \( \alpha_{ij} = 0 \). In the case of the RST, the phase tolerance parameter, \( \sigma_P = \frac{\pi}{8} \).  

8.4 The symmetric intensity function of the Reflectional Symmetry Transform. The function peaks when the two intensities are equal. As the two intensities differ, the function’s output drops at a rate controlled by \( \sigma_I \). In this example, \( \sigma_I = 0.1 \).  

8.5 Three pairs of points sharing a single reflection plane. In each case, the distance between point \( i \) and the reflection plane; and the distance between the reflection plane and point \( j \) are equal. In the case of the Generalised Symmetry Transform, the top pair are more symmetrical than the bottom pair because in the top case, the distance between points \( i \) and \( j \) is less than the that in the bottom case. This is not consistent with the functional definition of reflectional symmetry.  

8.6 The Reflectional Symmetry Transform’s symmetric distance function. When the distance between point \( i \) and the reflection plane is equal to the distance between point \( j \) and the reflection plane, the output is maximum. As the two distances differ, the functions output drops at a rate controlled by \( \sigma_D \). In this example, \( \sigma_D = 1 \).  

8.7 Alternative functions to the GST’s distance weight function. These are only appropriate where suppression of point pairs’ contributions is desired based on the distance between each point.  

8.8 Comparison of the output of the GST and RST.  

8.9 Comparison of the output of the GST and RST.  

8.10 Comparison of pattern recognition on the training data set using the Generalised Symmetry Transform (GST); the Noise Tolerant GST (NTGST); the Generalised Asymmetry and Symmetry Transform (GAST); and two configurations of the Reflectional Symmetry Transform: low and high tolerances to deviations from the ideal symmetric case.
8.11 Comparison of pattern recognition generalisation on the test data set using the Generalised Symmetry Transform (GST); the Noise Tolerant GST (NTGST); the Generalised Asymmetry and Symmetry Transform (GAST); and two configurations of the Reflectional Symmetry Transform: low and high tolerances to deviations from the ideal symmetric case.

8.12 Graph to show the effect of increasing the tolerance to deviations from the ideal symmetric case when performing pattern recognition using the Reflectional Symmetry Transform.

8.13 Comparison of symmetry transforms performing pattern recognition with increasing Gaussian noise. The control results are based on a Probabilistic Neural Network performing recognition using the traditional features (with noise).
List of Tables

2.1 A tabulated taxonomy of the symmetry transforms. .............. 60

4.1 Experimental results comparing generalisation performances across the four experimental networks: the traditional, control network; symmetry magnitude features network; symmetry orientation features network; and both symmetry magnitude and orientation features network. Peak generalisation performances are shown in bold. Generalisation performance is measured as the recognition rate on the test set of patterns. The control data used was the original USPS patterns without the application of any symmetry transforms, as is common in the application of neural networks in character or digit recognition. (Tabulation of data in Figure 4.2) ............ 103

4.2 Data to show the effect of classifying normalised versus non-normalised symmetry magnitude features. Peak recognition performances are highlighted in bold, where the smoothing parameter maximises both test and training set performances. (Tabulation of data in Figure 4.3) ......................... 106

5.1 Recognition rates for the two networks, listed by data sets, against the Position Offset Distance. ....................... 125

6.1 The results of the orientation-invariant pattern classification experiment for right angular input (prior to normalisation) orientations. ............................................. 151

7.1 Combinations of point orientations defined as representing perfect symmetry by the Generalised Symmetry Transform’s Phase Function, $P(i,j)$. ................................. 160
7.2 The results of the feature classification tests using non-normalised input. The classification and generalisation performance provided by the recognition rates of the training set and test set respectively. Comparison is made between the three GST based symmetry transforms. Performance is measured for a range of Probabilistic Neural Network smoothing parameters, $\sigma$.

7.3 The results of the feature classification tests using intensity normalised input. The classification and generalisation performance provided by the recognition rates of the training set and test set respectively. Comparison is made between the three GST based symmetry transforms. Performance is measured for a range of Probabilistic Neural Network smoothing parameters, $\sigma$.

8.1 Data to show the ability of the classifier to recognise the training set using symmetry features for the Generalised Symmetry Transform (GST); the Noise Tolerant GST (NTGST); the Generalised Asymmetry and Symmetry Transform (GAST); and two configurations of the Reflectional Symmetry Transform: low and high tolerances to deviations from the ideal symmetric case.

8.2 Data to show the ability of the classifier to generalise by performing recognition on the test set using symmetry features for the Generalised Symmetry Transform (GST); the Noise Tolerant GST (NTGST); the Generalised Asymmetry and Symmetry Transform (GAST); and two configurations of the Reflectional Symmetry Transform: low and high tolerances to deviations from the ideal symmetric case.

8.3 Data to show the effect of noise on the performance of a Probabilistic Neural Network performing digit recognition using traditional features (the control) and symmetry features produced by the Generalised Symmetry Transform (GST); the Noise Tolerant GST (NTGST); the Generalised Asymmetry and Symmetry Transform (GAST); and two configurations of the Reflectional Symmetry Transform: low and high tolerances to deviations from the ideal symmetric case.

9.1 A tabulated taxonomy of the GST-based transforms.
List of Algorithms

2.1 Huebner’s Qualitative Symmetry Transform .................................. 36
2.2 Generalized Symmetry Transform .............................................. 46
2.3 Discrete Symmetry Transform .................................................. 50
2.4 Fast Radial Symmetry Transform .............................................. 58
3.1 Perceptron Learning Algorithm ................................................ 68
3.2 Backpropagation Algorithm .................................................... 71
3.3 Kohonen Self-Organizing Feature Map Algorithm ....................... 78
4.1 Intensity Normalisation Algorithm ........................................... 102
Abstract

An investigation has been made into symmetry features of patterns as a means by which the patterns are described, or with which they are transformed prior to classification in order to assist a pattern recognition system.

There are two main points of departure from existing symmetry use in the pattern recognition domain. The first is the adoption of the theory that patterns can be categorised solely using a map of the symmetry features that exist within the static pattern. The second is the application of symmetry transforms to aid non-trivial recognition in patterns which are not intended to be perfectly symmetrical.

An experiment is conducted to classify the reflectional symmetry features of digits; using the Generalised Symmetry Transform to produce the features and Probabilistic Neural Networks to perform the classification. Symmetry feature information is also used to define parameters of affine transformations to achieve improved performance in tolerance to variances in position and orientation.

The results lead to an investigation of the role of asymmetry. The Generalised Symmetry Transform is modified to produce two related transforms: the Generalised Asymmetry Transform and the Generalised Asymmetry and Symmetry Transform.

Finally, a new symmetry transform is proposed which separates the factors affecting the degree of symmetry in an image into three non-linear functions of corresponding pairs of pixels. These factors are: the colour intensity values; the pixel orientations; and the respective distance between point and potential reflection plane. The strictness of symmetry, or tolerance to non-symmetrical artifacts, is defined in variable parameters which are set to suit the desired application. This new transform is called the Reflectional Symmetry Transform. The structure of its input and output match those of the Generalised Symmetry Transform, which it is intended to replace.
Declaration

I declare that no portion of the work referred to in the thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Copyright Statement

The author of this thesis (including any appendices and/or schedules to this thesis) owns certain copyright or related rights in it (the “Copyright”) and he has given The University of Manchester certain rights to use such Copyright, including for administrative purposes.

Copies of this thesis, either in full or in extracts and whether in hard or electronic copy, may be made only in accordance with the Copyright, Designs and Patents Act 1988 (as amended) and regulations issued under it or, where appropriate, in accordance with licensing agreements which the University has from time to time. This page must form part of any such copies made.

The ownership of certain Copyright, patents, designs, trade marks and other intellectual property (the “Intellectual Property”) and any reproductions of copyright works in the thesis, for example graphs and tables (“Reproductions”), which may be described in this thesis, may not be owned by the author and may be owned by third parties. Such Intellectual Property and Reproductions cannot and must not be made available for use without the prior written permission of the owner(s) of the relevant Intellectual Property and/or Reproductions.

Further information on the conditions under which disclosure, publication and commercialisation of this thesis, the Copyright and any Intellectual Property and/or Reproductions described in it may take place is available in the University IP Policy (see http://www.campus.manchester.ac.uk/medialibrary/policies/intellectualproperty.pdf), in any relevant Thesis restriction declarations deposited in the University Library, The University Library’s regulations (see http://www.manchester.ac.uk/library/aboutus/regulations) and in The University’s policy on presentation of Theses.
Acknowledgements

I would like to thank my mother, sister and family for their love, support and encouragement whilst conducting research I hope my father would have been proud of; and Dr Richard Neville, my supervisor, for his wisdom and guidance.
Chapter 1

Introduction

1.1 Research Context

Pattern recognition and classification systems capture real world objects in a digital representation form; perform feature processing (such as object segmentation and noise removal); feature extraction; and make a decision as to which category of patterns that object belongs. Typically, the decision leads to an action or further abstraction of the object.

An example is Optical Character Recognition (OCR.) A transducer (such as a camera) captures the page of text and produces a digital image representing it. A segmentation algorithm splits the image into smaller images, each representing a single character. Noise reduction algorithms are applied. For each character, a classifier (such as an artificial neural network) takes the features and decides which character the feature represents. In this way, the system could create an electronic document representing the page; or query a character string captured from a video of traffic against a database of ve-
hicle registrations, as in the case of Automatic Number Plate Recognition (ANPR) systems.

1.2 Research Problem

Although considerable progress has been made in improving the performance of pattern recognition and classification systems, they have not yet reached the performance of humans. These systems work well under ideal conditions: when the patterns have little variation amongst members of a class; the differences between patterns across class boundaries are significant; or there is little or noise interference. However, when there are many classes of patterns; or variances in position, orientation, size, skew and lighting; when noise is introduced to the patterns; or variances in the form of pattern (as in the case of hand produced versus machine produced patterns); or when an object in the pattern is occluded by another - classification performance suffers.

How can a pattern recognition and classification system better tolerate such variances and the introduction of noise? Is there an underlying feature or description of patterns that can better represent classes - one that better describes common pattern elements amongst unseen variances within the class? Humans provide existential proof: they can recognise patterns in these non-ideal circumstances with a high degree of accuracy.
1.3 Relevance of Symmetry

Symmetry is a structural property of geometric objects that describes the balance and similarity internally within a structure. Symmetry is invariant to many transformations: it is preserved when a transformation is applied to the whole object. It typically reflects the balance required by physical properties in nature. It is not a property of random artefacts. This makes symmetry highly relevant to the research problems of pattern recognition and classification systems.

Can symmetry provide a better representation of a pattern, one that describes the balance of structure of patterns belonging to a given class? Can the symmetry of a pattern assist in its recognition and classification after the pattern has been transformed? Can pattern recognition and classification systems tolerate noise better if they are based on symmetry features?

1.4 Overview

This thesis is an investigation into use of symmetry in pattern classification recognition using artificial neural networks. The main objective is to improve a network’s ability to perform generalisation and tolerate variances introduced to the patterns. Thus it presents processes and mechanisms for using symmetry to achieve these objectives.

There are two main points of departure from existing symmetry use in the pattern recognition domain. The first is the adoption of the theory that patterns can be categorised solely using a map of the symmetry features
that exist within the static pattern. The most similar existing work identifies individuals based on the symmetry of their gait, with respect to time [26]; these features contain considerably more information than the theory proposed here.

The second main point is the application of symmetry transforms to aid non-trivial recognition in patterns which are not intended to be perfectly symmetrical. A trivial case is considered to be the classification of objects into two sets: symmetrical and otherwise. Existing work describes how symmetry is applied to the detection of objects or features of objects known to be symmetrical. This includes the prior example of gait classification [26], as every individual’s gait is known to be symmetrical (with the exception of amputees). An additional example is transforming patterns known to contain human faces. In this case, symmetry is known to exist at various levels. Transforms are tailored to identify these known cases. More specifically, these works involve multiple classes but each class is known to share common symmetry features. This work investigates the recognition of multiple classes that do not share such symmetry features and as such are non-trivial pattern recognition tasks.

A method for measuring asymmetry (symmetry’s complement) is proposed and its affect on these mechanisms investigated and discussed. Finally, a new reflectional symmetry transform is presented.
1.5 Chapter Summary

Chapters 2 and 3 are of an introductory nature. Chapter 2 reviews the literature of related topics in symmetry. In particular, the paradigm of interpreting symmetry as a continuous measure in the domain of digitised image representations of objects and object shape. An overview of relevant investigations into the use of symmetry in nature is provided. The application of symmetry to problems involving digital images and their understanding or manipulation, are highlighted. A review of symmetry transforms from the literature is conducted.

Chapter 3 introduces artificial neural networks. The notions and terminology discussed are used throughout the rest of the thesis. This includes how artificial neural networks simulate their biological counterparts which inspired their creation. The functions and roles that networks can perform. The chapter presents the various atomic component types and the various structures that can be composed from them. It covers how networks learn or are trained to perform their functions. A review of several networks from the literature is provided.

Chapter 4 presents a means by which patterns can be solely classified by a representation of the measurement of symmetry in their structure. It introduces the problem of offline character recognition, which is used to test all the hypotheses in this thesis. The performance of such a classifier is compared to the method presently used to solve the problem.

Chapters 5 and 6 investigate the use of symmetry features as parameters in transforming the original features prior to their presentation to a
classifier. In particular, they examine affine transformations for normalising the position and rotation of patterns respectively. In the proposed way, a given classifier becomes invariant to changes in position or rotation. The experiments highlight deficiencies in existing symmetry transforms for which solutions are given.

Chapter 7 considers the importance of asymmetry in symmetry feature based classification and transformation. Modifications to existing transformations are proposed. These allow asymmetry to be measured separately and also as part of a unified symmetry-asymmetry measure.

Chapter 8 identifies significant issues with the Generalised Symmetry Transform which leads to the identification of three components that must be considered when determining a symmetry measure; and a new symmetry transform definition. The usefulness of the new transform’s features in pattern recognition is investigated as well as the effect of adding noise prior to extracting the symmetry features.

Finally, Chapter 9 describes the development of the main ideas of the thesis. It provides a summary, overview of the thesis’s contribution, conclusion and proposes further related work.

The cross-referencing conventions are as follows: section numbers are preceded by a ‘§’ and the first part of the subsequent number denotes the chapter. Hence, §2.6 is the sixth section in Chapter 2.
Chapter 2

Symmetry

2.1 Introduction

Symmetry is a property of many of the objects in the world around us, often a by-product of simplicity in design and manufacturing. This thesis focuses on the symmetry of geometric objects contained in patterns and images; and the way in which this information can be used in pattern recognition. This chapter serves as an introduction to symmetry and a review of the literature.

This chapter begins with definitions of symmetry and then reviews its use in nature and in computer science. Four methods for measuring symmetry are discussed in detail, providing the mathematical models, algorithmics and examples. These are then compared and contrasted.
2.2 Reflectional Symmetry

An object that is invariant to reflection about a given plane or axis has reflectional symmetry \([100]\). This plane is called the reflection plane. This is also called mirror symmetry, due to the reflective properties of mirrors, where the mirror serves as the reflection plane. An object may support more than one reflection plane.

An object has bilateral symmetry if it supports a reflection plane centrally on the object, where the reflection of one half of the shape completely describes the structure of the remaining half. The distance from the reflection plane at which the reflective invariant properties hold falls into two distinct categories: global symmetry, in which the invariance applies to the entire shape or image; and local symmetries, in which the invariance applies to a subset or region of the overall shape or image. A shape that supports global symmetry may consist of many local symmetries.

The intuitive notion of symmetry is that a shape either is symmetrical, or it is not. Such a description of symmetry is termed discrete. However, the concepts of local versus global symmetry; the observation in nature of objects that are nearly symmetrical, but not quite - such as the human face; and the need to cope with noise, errors and limited measurement accuracies, has led to the need for continuous descriptions or measures of symmetry. These descriptions rate the shape’s reflective invariance over a continuous interval.

Examples of discrete, global reflectional symmetries for a set of common shapes are depicted in Figure 2.1.
Figure 2.1: Basic geometric shapes with their respective planes by which they are invariant to reflection. Specifically, these are discrete, global reflection planes.
2.3 Rotational Symmetry

An object that is invariant to rotation about a point has rotational symmetry [100]. The number of rotations that the object is invariant to, is described as the order of rotations. For example, a square has rotational symmetry order of four because it can be rotated by $\frac{\pi}{2}$ radians four times and produce the exact same shape.

An object has circular symmetry if it is invariant to rotation in a two-dimensional space. For rotation in a three-dimensional space, an object is said to have spherical symmetry.

The concept of locality and degree of symmetry from the previous section can be extended to rotational symmetry. For example, the eyes of a human face have rotational symmetry. Hence, the human face supports two localised areas of rotational symmetry in addition to being nearly bilaterally symmetrical.

2.4 Symmetry in Nature

Symmetry can be detected by humans [5, 55, 56] and other animals (examples: swallows [68], pigeons [14]).

Barlow and Reeves [5] conducted experiments to explore under what circumstances humans can detect symmetry. Using random dot displays, they demonstrated that symmetry was not detected in a discrete manner. Mirror symmetry can be detected when the reflection plane is at various orientations, even when it is not central in the visual field. Our ability to detect
symmetry does not require both eyes (binocular vision). Delius and Nowak [14] show that monocular detection of symmetry applies to pigeons as well.

Locher and Nodine [56] tracked human eye movements when viewing symmetrical and non-symmetrical shapes. They concluded that the symmetry of a shape is detected globally across the visual receptive field. Once identified as bilaterally symmetrical, the eye scans only one half of the shape. When a shape is not symmetrical, the eye is tracked scanning the whole of the shape. Thus, efficiencies are made when possible, i.e. in symmetrical shapes when sufficient detail is described in only one half of the shape.

Palmer and Hemenway [71] measured the speed at which symmetry is identified. They showed that rotational symmetries take the longest. Reflectional symmetry detection depends upon the number of symmetry planes and their orientation. The greater the number of planes, the quicker the detection. Vertical symmetries are detected most quickly. Horizontal symmetries are detected more slowly and diagonal symmetries slower still.

Palmer also examined the role of local and global symmetries in shape perception [70]. The orientation of a group of shapes, each having multiple local symmetries is interpreted by humans to be oriented with respect to the shared global symmetry.

Møller [68] studied the mating preferences of swallows (*Hirundo rustica*). Male swallows with longer and more symmetrical tails mated earlier and with greater annual success than those swallows with shorter, less symmetrical tails. After discounting male-male aggressive interactions, Møller concluded that the male success was determined by female choice. It is hypothesised that long symmetrical sexual organs are perceived by female swallows as an
indication of the genetic and environmental quality. The experiment observed no effect on the quality of offspring due to tail length or symmetry. However, asymmetry in the tail would affect manoeuvrability in flight and therefore the ability to catch food.

Research has been conducted to explain how symmetry detection in animals developed [41]. Helm [96] describes an approach that supports the theory that symmetry detection emerged by natural selection. Johnstone’s [37] hypothesis is that preferences for symmetry evolved without a link between quality and symmetry. An experiment was conducted using an artificial neural network trained to recognise an abstract representation of a bird’s tail. This network favoured symmetrical unseen tails. Johnstone argues this is due to the average of the training tails being symmetrical. Enquist and Arak [16] conducted a similar experiment but showed that the preference was the result of the co-evolution of the input signal (i.e. the tails) and the networks.

2.5 Computational Use of Symmetry

This section will review the application of symmetry to computational problems.

Symmetry is used in medical image processing. It is used to detect the orientation of digital chest x-ray images by locating the bilateral symmetry of the rib cage [97]. It is also used to detect tumours. The presence of tumours in the human body creates asymmetry in areas that are usually symmetrical. This includes the head and neck [60], and in the airways [59] using Computed Tomography (CT) scans; and brain tumours using Magnetic
Resonance Imaging (MRI) scans [99, 95, 40].

In robotics, symmetry has been used to detect features in a scene [85, 34, 35] and features from panoramic images [101]; and for visual tracking and for grasping objects [48]. On the matter of object detection, symmetry has been used to detect human faces [91], relying on the near-bilateral symmetry of the face and in other cases the high circular symmetry of the eyes [20].

Symmetry features are used in autonomous vehicle and driver assistance systems. The rear of a vehicle is bilaterally symmetrical and this has been used to detect the presence of a vehicle in a retina [45, 42]. This led to the use of symmetry to vehicle following, where one vehicle detects another and maintains its distance and relative speed [102] and is also used for general tracking [84]. Much of the earlier work required the use of computers with far fewer resources and processing capabilities compared to modern systems (and these systems had to be small enough to fit inside the vehicles.) As a result, the real time processing requirement demanded simpler, more primitive detection methods. In a related use, symmetry features are used to detect road signs to assist drivers [57, 19, 6].

Local asymmetry features have been used for the classification of human expression (joy, anger and disgust) [67]. The near bilateral symmetry of the face is disrupted in different ways by the three facial expressions.

2.6 Measuring Symmetry

In using symmetry feature information to solve or assist computational problems, a means by which symmetry is measured or detected is required. Such
algorithms are termed *symmetry transforms* or *symmetry measures*. The previous section hinted at one variation among transforms: computational complexity. Problems with a real-time processing requirement demand less complex transformations which may limit the accuracy or completeness of other symmetry transforms. This section will review and compare four symmetry transforms which vary in computational complexity and ability.

Other symmetry transforms exist in the literature. However, some can only measure symmetry if the axes are provided [67, 1] and others are only applicable to shapes that are almost symmetric [62, 92, 93]. These are significant limitations which prevent their general use and will not be considered here. Kiryati et al [23, 24] proposed a transform that varies significantly from those described in this section. It is based on a probabilistic genetic algorithm. As useful as genetic algorithms (or evolutionary computing) are, they do not aid our understanding of symmetry and the principles by which it can be detected.

### 2.6.1 Qualitative Symmetry Transform

Huebner [34, 33] presented a one dimensional, reflectional symmetry operator. It was designed for use by small robotic devices, so it is not as computationally intensive as other transforms. It has been referred to as a compact symmetry operator and more recently as a *qualitative* symmetry operator (QST).

The symmetry operator calculates a measure of symmetry for every pixel in the input image. This measure is determined by comparing the intensity
values of the neighbouring pixels (up to a maximum distance). It is weighted so that comparisons nearer the point of measurement are more significant than those at the boundary of the neighbourhood.

The result of the transform is a *symmetry image* equal in size to the original image. The symmetry image represents the symmetry in a single dimension only: either horizontally or vertically. Non-maximal suppression applied to the symmetry image yields the symmetry maxima. The symmetry maxima can be overlaid onto the original image so that the symmetry axes can be visually inspected. An example image and its three representations of symmetry are depicted in Figure 2.2.

The transform’s mathematical model, being one dimensional, is described in terms of pixels, \( p \), in a given row, \( R \) of the image.

The symmetry measure is based on the normalised mean square error. For each pixel \( p_i \) in the row, we calculate the reflective symmetry, \( S(p_i, m) \) using:

\[
S(p_i, m) = 1 - \frac{1}{C \cdot m} \sum_{j=1}^{m} \sigma(j, m) \cdot g(p_{i-j}, p_{i+j})^2
\]

where \( m \) defines the number of pixels either side of pixel \( p_i \) in which to measure symmetry. \( m \) must be greater than zero. \( C \) is a normalising constant based on the colour space and \( \sigma(j, m) \).

The error function, \( g(p_{i-j}, p_{i+j}) \), is defined as the difference between a pair of pixels’ colour vectors, \( \bar{p}_{i-j} \) and \( \bar{p}_{i+j} \):

---

32
Figure 2.2: Huebner’s Qualitative Symmetry Transform showing the vertical symmetry of a 170 x 128 pixel image, with $m = 20$. Brightness corresponds to symmetry.
\[ g(p_{i-j}, p_{i+j}) = \begin{cases} 
\| \vec{p}_{i-j} - \vec{p}_{i+j} \| & \text{if } p_{i-j} \in R \land p_{i+j} \in R \\
c & \text{otherwise.} 
\end{cases} \] (2.2)

There is no error when two pixels share the same colour vector. They are said to be symmetrical. Maximum error occurs when the difference between the colour vectors is the greatest range in the colour space. In most colour spaces this would be when one pixel is white and another is black.

Close to the boundaries of the row, \( p_{i-j} \) or \( p_{i+j} \) may be outside of the row. For these values the error function does not have the colour vector information. This is because it does not exist within the image. In these cases, the error function assumes the maximum error, \( c \), occurs. In the case of a grey scale image, where the colour is represented by integers in the range of 0 to 255, \( c = 255 \).

The symmetry measure weights the error function based on distance. The distance-weight function, \( \sigma(j, m) \) is based on the distance, \( j \), relative to the neighbourhood maximum distance, \( m \):

\[ \sigma(j, m) = 1 - \frac{|j|}{m + 1} \] (2.3)

The addition of one in the denominator is superfluous. It implies that \( m \) may equal zero. However, there is no semantic value in measuring symmetry across a neighbourhood distance of zero pixels. It implies measuring the symmetry only inside a given pixel, the result of which has no meaning. Hence the neighbourhood maximum distance should always be greater than zero and the definition of the distance-weight function could be replaced with:
\[
\sigma(j, m) = 1 - \frac{|j|}{m} \tag{2.4}
\]

Implied, but not defined, is the value of \( C \). Assuming the symmetry measure, \( S(p_i, m) \) is to be limited to the continuous range zero to one; and that \( C \) is to be based on the maximum error at all relatively weighted distances, \( C \) is defined as:

\[
C = \frac{\sum_{j=1}^{m} \sigma(j, m) \cdot c^2}{m} \tag{2.5}
\]

It is now possible to reinterpret the sub components of the symmetry measure. \( S(p_i, m) \) can be defined in relation to an (asymmetry) operator, \( A(p_i, m) \):

\[
S(p_i, m) = 1 - A(p_i, m) \tag{2.6}
\]

where \( A(p_i, m) \) is defined as:

\[
A(p_i, m) = \frac{1}{C \cdot m} \sum_{j=1}^{m} \sigma(j, m) \cdot g(p_{i-j}, p_{i+j})^2 \tag{2.7}
\]

This new definition complements the sub component’s purpose to calculate the distance from the ideal symmetry case.

An algorithm for computing the symmetry measure is provided in Algorithm 2.1.

The computation time depends on the value of the size of the operator mask, \( m \). Increasing \( m \) increases the number of pixel pair comparisons. Increasing \( m \) also has a secondary effect: increasing the size of two fading
Algorithm 2.1 Huebner’s Qualitative Symmetry Transform

for each row in the image do
  for each pixel $p_i$ in the row do
    asym = 0
    for $j = 1$ to $m$ do
      calculate the distance weight
      if $p_{i+j}$ and $p_{i-j}$ are within the row’s boundaries then
        $g =$ colour vector difference
      else
        $g =$ maximum error
      end if
      asym += distance-weight * $g$
    end for
    $sym(p_i) = 1 - \text{normalized asym}$
  end for
end for

The fading borders are the result of the operator mask extending beyond the image boundaries. The appropriate size of the operator mask is application dependent. At $m = 50$, vertical symmetry is detected in the keyboard in the image in Figure 2.4b. However, the vertical symmetry that was detected in the lamp at $m = 20$ (to the left of the monitor - Figure 2.4a) is no longer present.

The size of the operator mask represents a dependency on a priori information about the symmetry resolution in the image. To determine the symmetry of multiple objects, each differing in size, the operator must be applied multiple times and the output combined in some meaningful way. This increases the computation time and the choice of multi-resolution scheme is highly application dependent.

Two possible multi-resolution schemes are presented in [35]. Both calcu-
Figure 2.3: Huebner’s Qualitative Symmetry Transform showing the fading border associated with values of $m$. 
Figure 2.4: Result of modifying the size of the operator mask, $m$, on the symmetries detected.
late the symmetry images for all values of $m$ in the range $m = 1$ to $m = \frac{w}{2}$, where $w$ is the width of the image. The first is *mean maximum qualitative symmetry*. This consists of adding the maxima of each symmetry image. This produces a grey scaled image. The second method is *maximum mean qualitative symmetry*. In this method, the symmetry images are summed before applying non-maximal suppression. The result is a binary image.

The maximum mean qualitative symmetry is used in [35] for range estimation. The horizontal and vertical maximum means were calculated. The resulting two binary images were combined using the logical *and* operator. The original image properties and derived symmetry information at these locations were used as feature points. Across two panoramic images (provided by stereo camera) they estimated the range, based on the movement of the feature points.

This symmetry operator has also been used for visual mobile robot localisation [34] in the Robocup competition. The symmetry operator detected the field lines using a symmetry pattern structure. After detection of symmetry, a search for the pattern Asymmetry-Symmetry-Asymmetry was performed. Symmetry occurs in the centre of a field line. Asymmetry occurs at the edge of the field line, where there is hard colour transition. It was noted that searching for just the pattern of Asymmetry-Symmetry would be sufficient. This is due to the symmetrical nature of the pattern, where the presence of Asymmetry-Symmetry implies Asymmetry-Symmetry-Asymmetry. A small symmetry operator mask was used.

Huebner states that the operator is invariant to changes in illumination. However, Figure 2.4 (a) visually highlights the problem with this claim. In-
spection of the figure reveals that illumination has affected the grey scale intensities of the monitor. Three separate vertical symmetries are detected following the colour contours created by illumination.

2.6.2 Generalised Symmetry Transform

Reisfeld et al [79] presented a two dimensional, reflectional or rotational symmetry operator. It is used as an attentional operator. Regions of high symmetry are identified for additional, more computationally intensive, processing.

The symmetry operator calculates a symmetry contribution for every pair of edges detected in the image. The contribution is a function of the gradient intensities and orientation. Areas of uniform colour, although symmetrical, are ignored. Like the Qualitative Symmetry Transform (§2.6.1), contributions are weighted by their distance. The greater the distance, the less significant the contribution.

The Generalised Symmetry Transform (GST) produces a map of the symmetry magnitude at each point and the symmetry’s orientation. As with Huebner’s Qualitative Symmetry Transforms, non-maximal suppression or thresholding can be applied post-transform depending upon the application’s requirements.

The first step is to approximate the gradient at each pixel in the input image, $I$. One method is to use the Sobel edge detector masks. The masks are separately convolved with the input image:
Figure 2.5:GST: Pixels $p_i$ and $p_j$, each associated with a vector $(r, \theta)$, contribute to the symmetry of a pixel $p$ mid-way between them.

\[
\delta_x p_k = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \ast I \tag{2.8}
\]

\[
\delta_y p_k = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \ast I \tag{2.9}
\]

where "\ast" is the two-dimensional convolutional operator.

The gradient, $\Delta p_k$ is a vector of the two convolution operations at a pixel point $p_k$:

\[
\Delta p_k = \left( \frac{\delta}{\delta_x} p_k, \frac{\delta}{\delta_y} p_k \right) \tag{2.10}
\]

The orientation $\theta_k$ of the pixel $p_k$ is defined as:
\[ \theta_k = \arctan \left( \frac{\frac{\delta}{\delta \sigma} p_k}{\frac{\delta}{\delta \sigma} p_k} \right) \] (2.11)

The symmetry of a point \( p \) is calculated from contributions by surrounding points which are equidistant from \( p \). A set \( \Gamma(p) \) consists of all the point pair combinations which share a mid-point, \( p \):

\[ \Gamma(p) = \left\{ (i,j) \mid \frac{p_i + p_j}{2} = p \right\} \] (2.12)

For each combination of pair points in \( \Gamma(p) \), a symmetry contribution, \( C(i,j) \) is calculated:

\[ C(i,j) = D_\sigma(i,j)P(i,j)r_ir_j \] (2.13)

The measure of symmetry is determined by a orientation based function, \( P(i,j) \):

\[ P(i,j) = [1 - \cos(\theta_i + \theta_j - 2\alpha_{ij})] \times [1 - \cos(\theta_i - \theta_j)] \] (2.14)

Two points are said to be symmetrical if the difference in their orientations is 90 degrees. These points are not symmetrical if the orientations are identical.

The contribution is weighted by the distance between the two points, which is the function of a Gaussian where \( \sigma \) controls the shape:

\[ D_\sigma(i,j) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{||p_i+p_j||}{2\sigma^2}} \] (2.15)
The final terms in the contribution are the logarithms of the two gradient intensities:

\[ r_k = \log(1 + \| \Delta p_k \|) \] (2.16)

The logarithms are taken to reduce the effect of large differences in gradient intensities.

The symmetry orientation, is the average of the two contributing orientations:

\[ \varphi(i, j) = \frac{\theta_i + \theta_j}{2} \] (2.17)

The above equation, however, assumes that the orientation difference, \( \theta_i - \theta_j \) is 90 degrees. This occurs when points \( p_i \) and \( p_j \) are exactly symmetrical by the definition of \( P(i, j) \). A better symmetry orientation definition [80], i.e. not based on the ‘best case’, is:

\[ \varphi(i, j) = \arctan \frac{2 \cos(\theta_i - \alpha_{i,j}) \cos(\theta_j - \alpha_{i,j})}{\sin(\theta_i + \theta_j - 2\alpha_{i,j})} \] (2.18)

This was proposed by the original authors particularly for use with objects modified by perspective transformation.

The magnitude of the symmetry at point \( p \), \( M_\sigma(p) \) is simply the sum of all contributions from pairs of points in the set \( \Gamma(p) \):

\[ M_\sigma(p) = \sum_{(i,j) \in \Gamma(p)} C(i, j) \] (2.19)

The symmetry orientation is defined as \( \phi(p) = \varphi(i, j) \), for the most sig-
nificant contribution from the set of point pairs, \( \Gamma(p) \). Hence, the symmetry at point \( p \) is:

\[
S_\sigma(p) = [M_\sigma(p), \phi(p)]
\]  

(2.20)

The algorithm for the Generalized Symmetry Transform is listed in Algorithm 2.2. The GST algorithm is quite computationally intensive. However, there is scope to implement the algorithm in parallel. The order in which each pixel’s symmetry magnitude is determined has no significance. Furthermore, the aggregation of the contributions (thus, the order in which contributions from pixel pairs are determined) has no significance. This lack of flow, output or anti-dependencies in the loops makes the algorithm ideal for parallel implementation.

Limiting the radius in which neighbouring pixels contribute to the symmetry magnitude will reduce the loop count. Beyond a given radius, for a given \( \sigma \) term, the distance weight function, \( D_\sigma(i,j) \) will make contributions beyond a given distance insignificant. So such a limit would prevent unnecessary computation.

The symmetry magnitude map can be interpreted as a regular image. Figure 2.6 shows the symmetry magnitude map of the computer desk scene, using a range of distance-weight settings.

Since its original proposal, modifications to the GST have been published that allow it to detect rotational symmetries [80]; be tolerant to noise in the input image [72]; perform a focusing operation on objects of a specific size [75]; detect symmetry in colour images [28]; speed up execution time [48];
Figure 2.6: Normalised isomorphic symmetry map (magnitude) of transforms with different distance-weight settings.
Algorithm 2.2 Generalized Symmetry Transform

Approximate Horizontal Gradient
Approximate Vertical Gradient
Calculate Log of Gradient Magnitude, \( r \)
Calculate Gradient Orientation, \( \theta \)

for all pixel point, \( p \), in image do
    for all pixel point pairs, \((i,j)\), with mid-point \( p \) do
        if \( i \) and \( j \) have gradient \( \neq 0 \) then
            Calculate Phase Function, \( P(i,j) \)
            Calculate Distance Function, \( D(i,j,\sigma) \)
            Contribution = \( P(i,j) \times D(i,j) \times r_i \times r_j \)
            \( M(p) = M(p) + C(i,j) \)
            if \( C(i,j) > \text{SingleMostContribution}(p) \) then
                SingleMostContribution = \( C(i,j) \)
                Orientation\((p) = \text{CalculateOrientation}(i,j) \)
            end if
        end if
    end for
end for

and to detect specific corner angles in objects depicted in images [12].

Cho et al [11] presented a symmetry detection technique that uses the GST’s phase function as its symmetry metric but differs in its approach of exploring the image for symmetry. It extracts local features using affine-invariant or scale-invariant detectors. From this, it obtains potential symmetric pairs of features which are evaluated using the first component of the GST’s phase function. The pairs are then organised into clusters, initially with each cluster consisting of only one pair. The technique then iterates through a process called symmetry propagation. Given a symmetric pair, a reflective transform is defined in terms of two normalising transforms which describe the transformation of one feature into its symmetric pair. Features in the vicinity are compared. If they have an equivalent reflective transform,
the pair joins the cluster (and are removed from the previous cluster). As
clusters grow and intersect, if sufficiently equivalent, they merge. The pro-
cess is repeated and the area of the clusters grows to form convex regions of
known bilateral symmetry. The method is effective at detecting symmetry
belonging to image objects of various sizes and is robust to outlier interfer-
ence.

2.6.3 Discrete Symmetry Transform

Di Gesù and Valenti [20, 21, 22] proposed a two dimensional, rotational
symmetry operator. It was designed to detect features of the human face.
These features are then used for face detection.

The transform calculates discrete axial moments in non-uniform areas of
an image. But unlike the GST, these areas are not necessarily edge points.
Grey scale values at selected radii are considered to be weights, acting around
a centre point. If the moments are normalised, the transform is size invariant.
The symmetry measure is weighted by the degree of non-uniformity.

For a given greyscale image, the transform produces a map of equal di-
mension with each point representing the degree of circular symmetry at the
 corresponding input pixel. The measure is thus continuous. The discrete
 element of the transform is the input: it uses discrete (pixel) values; and the
 kernel radii produce digital circles.

The measure of circular symmetry, $DST(i, j)$ at a point $(i, j)$ is the mea-
sure of symmetry, $T(i, j)$ weighted by the measure of non-uniformity, $E(i, j)$:
Figure 2.7: An example kernel at a point \((i, j)\) and the relationship of two points, \((p, q)\) and \((l, m)\) meeting the radius criteria.

\[
DST(i, j) = E(i, j) \times T(i, j) \tag{2.21}
\]

The measure of non-uniformity is the sum of the absolute greyscale differences between 4-way connected points of two digital circles:

\[
E(i, j) = \sum_{(l, m) \in C_r, (p, q) \in C_{r+1}} |g_{l, m} - g_{p, q}| \tag{2.22}
\]

\(C_r\) and \(C_{r+1}\) are the digital circles having radii of \(r\) and \(r+1\) respectively. \(g_{x, y}\) is the greyscale value at the pixel position \((x, y)\). The connection relationship between the points \((p, q)\) and \((l, m)\) is depicted in Figure 2.7 and defined as:

\[
(l - p)^2 + (m - q)^2 = 1 \tag{2.23}
\]
The measure of symmetry, $T(i, j)$ is defined as:

$$T(i, j) = 1 - \sqrt{\frac{\sum_k (T_k(i, j))^2}{n} - \left(\frac{\sum_k T_k(i, j)}{n}\right)^2}$$

(2.24)

where $T_k(i, j)$ is defined as:

$$T_k(i, j) = \sum_{(l,m) \in C_r} \left| (i - l) \sin \left(\frac{k\pi}{n}\right) - (j - m) \cos \left(\frac{k\pi}{n}\right) \right| \times g_{l,m}$$

(2.25)

for $n$ axial moments each with slope $\frac{k\pi}{n}$, with $k = 0, 1, ..., n - 1$. Hence $T_k(i, j)$ is the sum of all radial moments around a centre point $(i, j)$ at a given radius $r$ around a digital circle, $C_r$.

An algorithm that performs the DST is listed in Algorithm 2.3.

Figure 2.8 depicts the output and intermediate stage outputs of the DST applied to the computer-desk scene. Figure 2.9 depicts the affect on the transform output, varying the radius and number of axial moment parameters.

Of interest is Di Gesù and Valenti’s method of selecting relevant symmetry zones. This was used in their work on identifying facial features for face detection. After computing the DST of an image $I$, the relevant symmetry zones were detected using the Gold rule:

$$THRESH(DST(I)) = \begin{cases} 
1 & \text{if } DST(I) > \mu + 3\sigma, \\
0 & \text{otherwise.}
\end{cases}$$

(2.26)

Where $\mu$ is the mean value and $\sigma$ the standard deviation of $DST(I)$. A
Algorithm 2.3 Discrete Symmetry Transform

for $i = 0$ to image.Width do
    for $j = 0$ to image.Height do
        for $k = 0$ to N do
            $l = r \cdot \cos(k\pi/n) + i$
            $m = r \cdot \sin(k\pi/n) + j$
            $p = (r + 1) \cdot \cos(k\pi/n) + i$
            $q = (r + 1) \cdot \sin(k\pi/n) + j$
            
            if inBounds($l,m$) then
                distance = $((i - l) \cdot \cos(k\pi/n)) - ((j - m) \cdot \cos(k\pi/n))$
                moment = abs(distance * $g(l,m)$)
                $Tk(i,j) += moment$
                TkSquared(i,j) += moment * moment
            end if
            
            if inBounds($p,q$) then
                $E(i,j) += abs(g(l,m) - g(p,q))$
            end if
        end for
        $A = \sqrt{TkSquared(i,j)/n - (Tk(i,j)/n)^2}$
        $S = 1 - A$
        $DST(i,j) = S \cdot E(i,j)$
    end for
end for
Figure 2.8: The Discrete Symmetry Transform: for an input image, $I$, the DST is computed from an non-uniformity measure, $E$ and distance from perfect rotational symmetry, $T$. 
Figure 2.9: The output of the Discrete Symmetry Transform for varying radii, $r$, and number of axial moments, $n$. 

(a) $r = 5, n = 5$  

(b) $r = 5, n = 5, \text{GoldRule}$

(c) $r = 15, n = 15$

(d) $r = 15, n = 15, \text{GoldRule}$

(e) $r = 30, n = 30$

(f) $r = 30, n = 30, \text{GoldRule}$
comparison between the DST’s output, and the output thresholded using the Gold rule is depicted in Figure 2.9.

The DST and Huebner’s Qualitative Symmetry Transform share a similar symmetry measure structure. Both determine how much asymmetry is in the image. In the same way that Equation 2.1 was reinterpreted as Equations 2.6 and 2.7, the DST equation can be interpreted as:

\[ T(i, j) = 1 - A(i, j) \]  

(2.27)

where \( A(i, j) \) is a measure of asymmetry in the image:

\[ A(i, j) = \sqrt{\frac{\sum_k (T_k(i, j))^2}{n} - \left(\frac{\sum_k T_k(i, j)}{n}\right)^2} \]  

(2.28)

\( A(i, j) \) is calculating the distance from the ideal case, maximum rotational symmetry.

2.6.4 Fast Radial Symmetry Transform

Loy and Zelinksy [58] proposed the Fast Radial Symmetry Transform (FRST) to identify regions of interest within a scene. It is a two-dimensional rotational symmetry operator. It was designed to reduce computational effort in order to be used for real-time processing.

The FRST first estimates the gradients in the image using the Sobel edge operator. A magnitude projection image, \( M_n \), and an orientation projection image, \( O_n \), is calculated for each radius \( n \), from the image gradients. The use of image gradients and the production of two projections (for magnitude and orientation) is similar to the GST’s process.
Each pixel, \( p \), in the image contributes to the symmetry measures of two other pixels. The first is the pixel \( p_{+ve}(p) \), that the gradient points to, at a distance \( n \) from \( p \). This is termed the \textit{positively-affected pixel}. The second pixel, \( p_{-ve}(p) \) is the pixel that the gradient of \( p \) points away from, at the same distance \( n \). This is the \textit{negatively-affected pixel}. This contribution scheme differs from the GST in that in the FRST a local neighbourhood of pixels does not contribute to a central point.

The positively-affected pixel’s co-ordinates are

\[
p_{+ve}(p) = p + \text{round} \left( \frac{g(p)}{\|g(p)\|} n \right)
\]

(2.29)

and the negatively-affected pixel’s co-ordinates are

\[
p_{-ve}(p) = p - \text{round} \left( \frac{g(p)}{\|g(p)\|} n \right)
\]

(2.30)

where \( g(p) \) is the gradient vector of pixel \( p \), and ‘round’ is a function that rounds co-ordinates to the nearest integer, ensuring that co-ordinates directly align with pixel positions. The locations of \( p_{+ve}(p) \) and \( p_{-ve}(p) \) are depicted with respect to \( g(p) \) and \( n \) in Figure 2.10.

The projection images \( M_n \) and \( O_n \) are initially zero. The magnitude projection image, \( M_n \) is produced by incrementing the positively-affected pixel and decrementing the negatively-affected pixel by the gradient vector’s magnitude:

\[
M_n(p_{+ve}(p)) = M_n(p_{+ve}(p)) + \|g(p)\|
\]

(2.31)
Figure 2.10: The pixel $p$ affects two others: one positively, one negatively at a distance $n$ that the gradient points to and away from respectively.

\[ M_n(p_{-ve}(p)) = M_n(p_{-ve}(p)) - \|g(p)\| \] (2.32)

The orientation projection image is produced in a similar manner, except that the incrementing and decrementing value is 1:

\[ O_n(p_{+ve}(p)) = O_n(p_{+ve}(p)) + 1 \] (2.33)

\[ O_n(p_{-ve}(p)) = O_n(p_{-ve}(p)) - 1 \] (2.34)

The symmetry contribution for a radius $n$ is defined as a convolution:

\[ S_n = F_n * A_n \] (2.35)

where
\[ F_n(p) = \frac{M_n(p)}{k_n} \left( \frac{|\tilde{O}_n(p)|}{k_n} \right)^{\alpha} \]  

(2.36)

where \( \alpha \) is the **radial strictness parameter** and

\[ \tilde{O}_n(p) = \begin{cases} 
O_n(p) & \text{if } O_n(p) > k_n, \\
k_n & \text{otherwise.} 
\end{cases} \]  

(2.37)

\( k_n \) is a scaling factor, used to normalise the magnitude projection image and the orientation projection image. \( A_n \) is a two-dimensional Gaussian whose elements sum to \( n \).

The symmetry of the image is calculated by taking the arithmetic mean of the contributions from all radii under consideration:

\[ S = \frac{1}{|N|} \sum_{n \in N} S_n \]  

(2.38)

An algorithm for performing the Fast Radial Symmetry Transform on an image is listed in Algorithm 2.4.

Figure 2.11 depicts the output of the FRST for various sets of radii. White areas represent high symmetry in bright regions of the original image. Black areas depict high symmetry in dark regions. The radii directly affect the transform’s ability to detect symmetry in small and large objects. However, multiple radii can be used. It is not necessary to detect symmetries across a continuous interval of radii, hence the FRST is very flexible and can detect symmetry in various sized objects simultaneously.

When the radial strictness parameter, \( \alpha = 1 \), the FRST performs the
Figure 2.11: The output of the Fast Radial Symmetry Transform on the image from Figure 2.2a for various radius $n$ in the set of radii $N$. In each case, the radial strictness parameter, $\alpha = 3$. 

(a) $n = 1$  
(b) $n = 5$  
(c) $n = 7$  
(d) $n = 1, 5, 7$
Algorithm 2.4 Fast Radial Symmetry Transform

\texttt{g = SobelOperator(I)}
\begin{algorithmic}
\ForAll{radii \( n \) in \( \mathbb{N} \)}
\State \( M_n[] = 0 \)
\State \( O_n[] = 0 \)
\ForAll{pixel \( p \) in \( I \)}
\State Calculate \( p_{+,ve}(p) \) Co-ordinates
\State Calculate \( p_{-,ve}(p) \) Co-ordinates
\State Increment \( M_n(p_{+,ve}) \) by \( \|g(p)\| \)
\State Decrement \( M_n(p_{-,ve}) \) by \( \|g(p)\| \)
\State Increment \( O_n(p_{+,ve}) \) by 1
\State Decrement \( O_n(p_{-,ve}) \) by 1
\EndFor
\State Calculate \( F_n \)
\State Calculate \( S_n \)
\EndFor
\State \( S = \text{ArithmeticMean}\{S_1, S_2, ..., S_n\} \)
\end{algorithmic}

role of a bilateral reflectional symmetry transform, rather than a rotational symmetry transform. Figure 2.12 depicts this for various sets of radii. It also demonstrates setting the set of radii to a continuous range.

2.7 Taxonomy

Table 2.1 summarises the differences and similarities between the four symmetry measures described here in detail: the Qualitative Symmetry Transform; the Generalised Symmetry Transform; the Fast Radial Symmetry Transform; and the Discrete Symmetry Transform.

Each transform is described by a column. The rows represent features, limitations or capabilities of the transforms. The first row, \textit{symmetry type detected} refers to which type or definition of symmetry the transform aims or claims to measure or detect. The second row, \textit{information produced}, states
Figure 2.12: The output of the Fast Radial Symmetry Transform on the image from Figure 2.2a for various radius $n$ in the set of radii $N$. In each case, the radial strictness parameter, $\alpha = 1$. At this setting, the FRST performs the role of a bilateral symmetry transform.
what information the transform’s output represents. All four transforms measure the magnitude (or strength) of the symmetry at points in the image, but only the FRST and the GST identify its orientation. The third row, measures dimensionality, refers to the number of dimensions from which the symmetry is calculated. Only the QST is limited to measuring a single dimension, the remainder use two-dimensional information.

The fourth row describes the computational complexity of the algorithms used to implement the mathematical models of the symmetry transforms. $K$ represents the number of pixels in the input image. The QST is of order $KM$, where $M$ represents the width of the one-dimensional operator mask. The FRST is of order $KN$, where $N$ represents the size of the two-dimensional space under consideration. The GST is more complex, of order $KN^2$. The
DST is of order $KB$, where $B$ is the number of angular bins or moments considered for each pixel.

The fifth row describes which information in the input images is used to calculate symmetry; the sixth row describes the symmetrical significance of uniform regions in the image. Uniform regions are intuitively symmetric. However, only the QST reflects this in its measurement of symmetry. The remaining transforms do not measure the symmetry in uniform regions because they are either considered unimportant; because they reduce computational overhead by calculating the symmetry from only the non-uniform regions; or because they depend on information derived from the estimated gradient in the image.

The seventh row lists the parameters which can be varied to affect the output of the symmetry transforms. The eighth row states the way the symmetry transforms rate the degree of symmetry to input features as those features increase in distance from the point under consideration. In the case of the QST and GST, contributions or features further away from the potential reflection plane are less influential than those closer to it.

### 2.8 Contribution

This chapter has provided a review of symmetry in the literature. In particular, reflectional and rotational symmetry; its use in nature and in computer science; algorithms for identifying or calculating the symmetry in images, which were compared and contrasted.

The chapter identified that in two transforms, the degree of symmetry
was calculated by measuring the asymmetry in the image. These two transforms were the Qualitative Symmetry Transform (QST) [34] and the Discrete Symmetry Transform (DST) [20]. In these cases, the mathematical models were reinterpreted to produce an equation of the form:

\[
\text{Symmetry} = 1 - \text{Asymmetry} \quad (2.39)
\]

Both transforms aggregate component pair differences from the ideal symmetric case as a ratio. The ideal symmetric case is an equality or balance of two components. The difference (or imbalance) as a fraction of the maximum possible difference (or imbalance) is representative of the asymmetry of the components. As such a ratio, the asymmetry measure is bound to the range zero to one. By subtracting the measure from one, the transforms produce a symmetry measure. The authors of the introductory papers do not identify the asymmetry component explicitly. But by labelling the asymmetry component, the relationship between symmetry and asymmetry is well defined. In addition to better understanding the way in which the transforms work, this permits us to measure asymmetry, if such a measurement is desired. Specifically, Equations 2.6 and 2.27 define the asymmetry measures implied by the QST and DST respectively.
Chapter 3

Neural Networks

3.1 Introduction

Artificial Neural Networks (ANNs) are termed a connectionist approach to artificial intelligence because of their structure. They are composed of distributed processing units which take input and output signals from and to other processing units. The processing units are simple, with the network’s functionality being the result of the way in which the units are connected and the weighting assigned to each connection. Research in Neural Networks is motivated by the existential proof of their significance: the biological equivalent, the brain.

This chapter serves as an introduction to the topic of Neural Networks. The chapter begins by identifying their general usage, then introduces the biological neuron and its artificial equivalent. The structure of networks of neurons and the means by which they are trained to perform their role is described, followed by a review of some popular neural networks from the
4.3 Usage

Neural networks perform several tasks: Content Addressable Memory, Associative Memory and Function Interpolation.

**Content Addressable Memory**

A set of values presented to the network, called input, is mapped to a set of output values. The mapping can take the form of one-to-one or many-to-one. This forms the basis of pattern classification. Patterns are presented to the network and mapped to their respective classes.

**Associative Memory**

In a network performing the functionality of an associative memory, the input and output have the same structure. In the ideal case, the input is mapped to an identical output. When a previously unseen input is presented to the network, the output is the closest matching known input. Used in this way, the networks are performing error correction.

**Function Interpolation**

Neural networks can model mathematical functions. The model is based on sample input-to-output mappings. The network can be used to estimate the function’s value for new input values between the range of known input values. This is called interpolation.
3.3 Biological Neuron

The biological neuron is the atomic processing unit in the mammal brain. A basic neuron is depicted in Figure 3.1. It consists of four major components: synapses, dendrites, the cell body and the axon.

Sensory fibres, called dendrites, transmit pulsing signals into the cell body. Basic processing of the signals in the cell body yields an output signal which is transmitted along the axon. The activity of transmitting a signal from the axon is called firing.

Neurons are connected together by synapses to form a network. The synapses are junctions with electro-chemical interactions varying in strength (or weight). It is at the synapse that information is stored. They can be excitatory in which the signal promotes the firing of the neuron; or inhibitory in which the signal inhibits the firing of the neuron.
3.4 Artificial Neuron: Threshold Logic Unit

The basic structure of an artificial neuron is shown in Figure 3.2. Node and unit are synonymous terms for neuron. The unit in Figure 3.2 is a McCulloch and Pitts’ [64] Threshold Logic Unit (TLU).

A set of inputs $X = x_1, x_2, ..., x_n$ model the (pre-synapse) incoming signals and a set of weights $W = w_1, w_2, ..., w_n$ model the synapse strengths. The cell body is modelled by two components: the summation of the weighted inputs, called the activation $(a)$; and an activation-output function. The output $y$ models the function of the axon.

$$a = \sum_{i=1}^{n} w_i x_i$$ (3.1)

A TLU has a binary input and uses the following activation-output function:

---

Figure 3.2: Basic artificial neuron (the McCulloch and Pitts model - Threshold Logic Unit (TLU)). The sum of the weighted ($w$) inputs ($x$) is passed through an activation function yielding the output ($y$).
\[ y = \begin{cases} 1 & \text{if } a \geq \theta \\ 0 & \text{if } a < \theta. \end{cases} \] (3.2)

\( \theta \) is a threshold value that determines whether the TLU fires (output of 1) or not (output of 0). Like the biological counterpart, weights may be excitatory or inhibitory by being of positive and negative value respectively.

### 3.5 The Perceptron

The Perceptron is a modified TLU to include association units, as shown in Figure 3.3.

The Perceptron learns by supervised training. Weights are modified according to the following weight-adaption rule, where \( w_i \) is the \( i^{th} \) weight and \( x_i \) is its corresponding input value:

\[ w_i(t + 1) = w_i(t) + \eta [d(t) - y(t)] \cdot x_i(t) \] (3.3)
If the actual output $y(t)$ is equal to the target output $d(t)$, the weight is not updated. Otherwise, $[d(t) - y(t)]$ determines the direction of the weight update. $\eta$, a momentum term equal to a positive fraction less than one, controls the rate of learning.

The threshold, $\theta$, is trained as a weight $w_\theta$ that has a constant input $x_\theta$ of 1.

**Algorithm 3.1** Perceptron Learning Algorithm

```
set all weights and threshold to small random value
repeat
    randomize order of training vectors
    for each training vector do
        present inputs to network
        calculate actual output
        adapt weights
    end for
until actual output = desired output for all training vectors
```

The Perceptron can be used as a maximum likelihood classifier of two classes of input which fit gaussian probability distributions with equal variance [50]. This is done by setting, rather than training, the weights and threshold based on the mean of the input ($i$) for classes A ($M_{Ai}$) and B ($M_{Bi}$) and the common variance of the input ($\sigma_i$).

$$w_i = \frac{2(M_{Ai} - M_{Bi})}{\sigma_i^2}$$

$$\theta = \sum_{i=0}^{N-1} \frac{M_{ai}^2 - M_{bi}^2}{\sigma_i^2}$$

Minsky et al [66] proved that a single perceptron is not capable of classifying non-linearly separable classes (such as the Boolean Exclusive OR func-
tion). Hence there is a need for a network of perceptrons.

### 3.6 Multi-Layer Perceptron

The Multi-Layer Perceptron (MLP) is a feed-forward network consisting of an output layer and one or more hidden layers of sigmoidal nodes [49, 25]. Any non-linear function can be approximated by a two layer perceptron (given sufficient nodes) [36]. Figure 3.4 depicts a two-layer perceptron (one hidden layer) of five nodes each. Each node utilises a non-linear activation function (such as sigmoidal). Multiple layers of non-linear nodes can model complex decision regions in the decision space.

When training a single perceptron, weights are updated using an error term calculated by comparing the actual output with the target output of the node. This mechanism can only be applied when the target output is known. When training a MLP network, the target output of the output layer is known but the target outputs of nodes in the hidden layers is generally not
known. Hinton et al [82] detail the work of Werbos whose training regime (based on the Delta rule) propagates an error term backwards through the network. The error term is based on the error at the output.

The weight adaption rule is presented below in which $t$ represents time. $\eta$ is a gain term and $x'_i$ is the output of the previous node.

$$w_{ij}(t + 1) = w_{ij}(t) + \eta \delta_j x'_i \quad (3.4)$$

The error term, $\delta_j$, for node $j$ in the output layer is based on the difference between the actual output of the node, $y_j$ and its target output $d_j$:

$$\delta_j = y_j(1 - y_j)(d_j - y_j) \quad (3.5)$$

Nodes in the hidden layer use a different error term. It uses the weight’s corresponding input value, $x'_j$ instead of the node’s output. It uses the error terms of all nodes in the following layer ($k$) proportionally according to the weight between nodes $w_{jk}$.

$$\delta_j = x'_j(1 - x'_j) \sum_k \delta_k w_{jk} \quad (3.6)$$

Using a momentum term, $\alpha$, the weight changes can be smoothed to improve convergence time.

$$w_{ij}(t + 1) = w_{ij}(t) + \eta \delta_i x'_i + \alpha (w_{ij}(t) - w_{ij}(t - 1)) \quad (3.7)$$

The Backpropagation algorithm is presented in Algorithm 3.2.
Nguyen and Widrow [69] demonstrated that the learning speed of a MLP can be improved by initially setting all weights randomly between a central subrange of the permitted input values. For example, if $X = [-1, 1]$ then weights are randomly set in the range -0.5 to 0.5.

When using the MLP for classification, the number of nodes in the output layer is equal to the number of classes required. When training, it is desired that the output node of the target class be of a high value (e.g. 1) and that all other output nodes have a low value (e.g. 0).
3.7 Popular Networks

There are many network topologies other than the Multi-Layer Perceptron. This section will review a selection of commonly used networks: the Hopfield network; the Kohonen Self-Organising Feature Map; Radial Basis Function neural networks; and the Probabilistic Neural Network.

3.7.1 Hopfield

Hopfield Networks are single, fully laterally connected, recurrent networks of linear nodes. They are historically used as an associative memory for error correction and pattern classification. Weights can be prescribed or trained under supervision.

The Hopfield Network is named after John Hopfield who proposed it in 1982 [31]. The network’s design is influenced by observations of elementary components in physical systems (such as the magnetic system and in fluid flow). It is similar to a network proposed earlier by Little [51, 52], differing primarily in choice of recurrent dynamics.

The network uses TLU nodes. Each node is connected to every other node, but not to itself. However, the output of a given node still affects its own input: all nodes influence the activation of every other node. The nodes’ outputs are calculated asynchronously. Figure 3.5 depicts a Hopfield Network consisting of three nodes.
The activation of the nodes is the same as the model originally proposed by McCulloch and Pitts [64]. However, the nodes receive and transmit a polarised binary signal, i.e. the set $-1, 1$. In addition, the threshold, $\theta$, is assumed to be zero. Hence the activation-output function becomes:

$$y = \begin{cases} 
1 & \text{if } a \geq 0 \\
-1 & \text{if } a < 0.
\end{cases}$$ (3.8)

The node weights can be set using the prescription:
\[ w_{ij} = \sum_p v_i^p v_j^p \quad (3.9) \]

Weight \( w_{ij} \) is the weight between nodes \( i \) and \( j \). The weight is set by summing the products of the participating nodes input, \( v \) for each pattern \( p \) in the training set.

A biologically feasible learning regime is the Hebb Learning Rule [27]. Weights are randomly set then updated using the adaption rule:

\[ w_{ij}(t + 1) = w_{ij}(t) + \alpha v_i^p v_j^p \quad (3.10) \]

Here, \( t \) is used as a time index. Hence \( w_{ij}(t) \) represents the weight between nodes \( i \) and \( j \) prior to adaption and \( w_{ij}(t + 1) \) after adaption. Both \( v_i^p \) and \( v_j^p \) retain their meaning from the weight prescription rule. The term \( \alpha \) is the learning rate constant, the value of which is bounded within the range 0 to 1 (not inclusive). It determines the maximum amount by which the weights change per unit of time, \( t \). Small values of \( \alpha \) will lead to longer training times. Generally, a learning rate constant is used to adapt the weights to a fraction of the product of \( v_i^p v_j^p \).

It is useful to describe the Hopfield Network in the context of a state machine. Each combination of node outputs is a network state. The network moves between states until it settles in a state. After this, the network remains in the state unless nodes have their input modified by an external source. Such states are termed stable states. All other states are *transitional states*.

The purpose of weight training is to create a stable state for each pattern.
in the training set. Input patterns not matching those of the training set start the network in a transitional state. Such an associative memory should be capable of generalisation or error correction. Patterns of classes differing from the training set, or suffering from noise are transformed by the network to the closest matching stable state. This represents the closest matching training pattern.

Figure 3.6 depicts a Hopfield Network in three different states (at three different times). In this example, it is acting as an associative memory performing character recognition. The letter “L” is first presented to the network in (a). The pattern initially contains errors. Through time, the network moves through states such as (b) until it finally settles on the state representing the closest matching training pattern in (c).

One issue for these networks is their storage capacity. Hopfield proposed that the number of classes stored by the network should be less than 0.15 times the number of nodes in the network. McEliece et al [65] proposed the following number of nodes:
\[ m < \frac{N}{2} \log N \]  

(3.11)

where \( m \) is the number of classes and \( N \) is the number of nodes.

Unintentional stable states are also a potential issue with Hopfield Networks [3]. These spurious states prevent the network from correctly classifying by creating new classes for unseen patterns.

Hopfield Networks have been used for combinatorial optimisation (such as the travelling salesman problem [32]) and image processing [7]. Although generally regarded as a historical network, they can still be found employed in recent research. Alonso et al have recently used Hopfield Networks for on-line parameter estimation for a robotic arm [2].

### 3.7.2 Self-Organizing Map (SOM) / Kohonen

In Kohonen’s Self-Organizing Feature Map (SOFM) [43] nodes form a single (self-organizing) layer with no lateral connections, fully connected to both the inputs and the outputs. It is based on the “winning-node-takes-all” scenario. However, it is the node with the minimum distance between weight and input vectors that is selected, rather than the node with the largest activation. The nodes in proximity to the selected nodes are trained to respond strongly to that input pattern.

A *neighbourhood* is a set of nodes surrounding a given, centrally placed node. The neighbourhood *distance* is defined as the number of nodes (inclusive) from the central node to the most outlying node within the neighbourhood. The self-organising layer is a one- or two-dimensional structure.
Neighbourhood schemes may vary, based on the layer structure. Three such schemes are depicted in figure 3.7.

![Neighbourhood Schemes](image)

(a) Linear  
(b) Rectangular  
(c) Hexagonal

Figure 3.7: Self-Organizing Layer and Neighbourhood Schemes with Distances of 1, 2 and 3 nodes displayed from the neighbourhood’s central node (shown in black) from Figure 8.13 of [25].

The Kohonen SOFM uses an unsupervised learning regime. Input patterns are presented to the network in random order with no output pattern preference specified. In the absence of lateral connections, traditional search methods are used to find the node $k$ with the minimum distance between input and weight vector:

$$\| \vec{w}_k - \vec{x} \| = \min_i \{ \| \vec{w}_i - \vec{x} \| \}$$  \hspace{1cm} (3.12)

A neighbourhood, $N_k$ is formed with node $k$ as its central node with a distance sufficient to cover greater than half of the network. The weights of nodes in the neighbourhood are adapted for a constant number of training cycles each with a decreasing learning rate using the following weight-update rule:

$$\Delta \vec{w}_j = \begin{cases} 
\alpha (\vec{x} - \vec{w}_j) & \text{if } j \text{ is in } N_k \\
0 & \text{if } j \text{ is not in } N_k 
\end{cases}$$  \hspace{1cm} (3.13)
The size of the neighbourhood is decreased and the weight adaption rules are applied again. This process continues until the minimum neighbourhood size has been reached. An algorithm for performing this training is presented in Algorithm 3.3.

**Algorithm 3.3 Kohonen Self-Organizing Feature Map Algorithm**

- randomize all weights
- randomize order of training vectors
  - for each training vector do
    - present inputs to network
    - calculate distance between the input and output
    - select node with minimum distance
    - for neighbourhood size N to minimum do
      - for 1 to Max Cycles do
        - for each node $j$ in neighbourhood do
          - update weights
        - decrease learning rate
      - end for
    - end for
  - end for

After a sufficient number of cycles, the training results in a natural ordering of the outputs. Similar input patterns are mapped in clusters of nodes. Whilst possible, it is not necessary to use an additional activation-output function: the output of the node can be the activation. Similar to using MLP for classification, when presenting an input pattern to the network, one output node will have a high output (representing its class) and all others will have a low output.
Figure 3.8: The lateral connections depicted as inhibitory (-) and excitatory (+) described as “On Centre off surround” by Gurney in Figure 8.2 of [25]. Kohonen SOFMs do not have these lateral connections.

Variants

Whilst the notion of the feature map is based on biological evidence, the Kohonen SOFM is not biologically realistic. Without lateral connections in the self-organizing layer, the initial highest output value node must be found by a search algorithm. An alternative scheme is for each node in the self-organizing layer to have the lateral weights depicted in Figure 3.8.

Each node is connected to every other node within a fixed distance neighbourhood. Between any two nodes in the same neighbourhood there are two
inhibitory connections of equal weight. Each node is also connected to itself by an excitatory connection. When a pattern is presented to the network, the node with the initial maximum output value will suppress all other nodes. The result is that one node will have a high output value and every other node will have a low value.

The lateral weights to nodes in the immediate neighbourhoods can be set to a very small positive (excitatory) value. This results in small neighbourhoods of nodes with high outputs, rather than just a single node.

The use of lateral connections in this way requires that each node use “leaky-integrator” dynamics. In this form of dynamics, the output of the node increases gradually (over time) to maximum with sufficient input. Without sufficient input, the output of the node gradually decreases back to zero. In addition, after each training step, the weights and vectors must be normalised.

3.7.3 Radial Basis Function (RBF) Network

Radial Basis Function (RBF) Neural Networks are three-layer, feed-forward networks consisting of both non-linear and linear nodes. They are commonly used for multi-variable function interpolation. Various supervised training methods exist, optimising training time or network size.

The theory on which RBF networks is based was proposed by Powell [77]. However, it was Broomhead and Lowe who proposed the network model [8].

RBF networks typically consist of two layers plus the input layer. The hidden layer consists of non-linear nodes using Gaussian based activation-
Figure 3.9: The structure of the Radial Basis Function Neural Network. ‘g’ denotes that the activation-output functions of the hidden layer are Gaussian based. The nodes of the output layer are linear.

Output functions. These nodes shape the decision boundaries. These are fully-connected to the input layer nodes. The output layer consists of linear nodes, which identify which decision region the input falls in, based on the previous layer. Figure 3.9 depicts the structure of an RBF network.

Nodes in the hidden layer use an activation based on the vector difference of the node’s inputs and the node’s weights:

\[ a = \|x - w\| \]  

(3.14)

The activation-output function is a Gaussian:
\[ y = \exp \left( -\frac{a^2}{2\sigma^2} \right) \] (3.15)

Where \( \sigma \) controls the width. Hence \( w \) acts as a centre of expected input. The function’s width, \( \sigma \), need not be identical for all nodes. The number of nodes in the hidden layer is arbitrary, but should be sufficient in number to adequately describe the decision region. The output function nodes are simply linear nodes, the number of which is application dependent: for example, a single node for function interpolation; one node per class for classification.

With respect to training, the weights of the hidden nodes correspond to the patterns identified as members of a particular classification. The output nodes are trained using the Widrow-Hoff learning rule \([10]\), where the error, \( \delta \) is defined as:

\[ \delta = t - o \] (3.16)

Where \( t \) is the target output and \( o \) is the actual output. Hence the adjustment to the weight \( w_i \) for the input \( x_i \) from the hidden layer node \( i \) is

\[ \Delta w_i = \eta \delta x_i \] (3.17)

Where \( \eta \) is the learning rate.

Alternative methods for training include the self-organisation of the hidden layer combined with supervised learning in the output layer, with the option of learning the width(s) of the activation-output functions in the hidden layer \([25]\); or randomly selecting centre values from the training patterns, and...
solving the linear output unit weights by minimising the sum of the square errors between the target and actual outputs [8].

A similar network is the Probabilistic Neural Network, which will now be described in detail.

### 3.7.4 Probabilistic Neural Network

Probabilistic Neural Networks (PNN) are four-layer, feed-forward networks consisting of both linear and non-linear nodes. They are commonly used as content-addressable memories for pattern classification. The weights of the network are prescribed. Hence ‘training’ is almost instantaneous. If known, the probability of an outcome can be incorporated into the network. For real problems, PNNs are typically very large requiring a lot of memory.

The theory behind PNNs was first proposed in the early 1960s by Donald Specht. However, the computational requirements were too high at that time to be applied. The network model was proposed in the late 1980s after sufficiently sized random-access memories became available [89].

PNN classification is based on a two-category Bayes decision rule:

\[ d(X) = \begin{cases} \theta_A & \text{if } h_A l_A f_A(X) > h_B l_B f_B(X) \\ \theta_B & \text{if } h_A l_A f_A(X) < h_B l_B f_B(X). \end{cases} \]  

(3.18)

Where \( \theta_A \) and \( \theta_B \) represent the classes \( A \) and \( B \). \( h_A \) and \( h_B \) are the prior probabilities of a pattern belonging to classes \( A \) and \( B \) respectively. A pattern either belongs to class \( A \) or class \( B \) exclusively, hence:
\[ h_A = 1 - h_B \] (3.19)

\( l_A \) is the loss associated with the decision that a pattern belongs to a class \( A \), when it actually belongs to class \( B \). \( f_A(X) \) is the probability density function (PDF) for class \( A \).

PNNs estimate the PDFs using Parzen [76] windows. Multivariate Parzen windows [9] with Gaussian kernels take the form:

\[
f_A(X) = \frac{1}{(2\pi)^{\frac{p}{2}}} \frac{1}{\sigma^p m} \sum_{i=1}^{m} e^{-\frac{(X - X_{Ai})(X - X_{Ai})}{2\sigma^2}}
\] (3.20)

Where \( m \) is the number of training patterns; \( i \) is the pattern number; \( X_{Ai} \) is the \( i^{th} \) training pattern of class or category \( A \); \( p \) is the dimensionality of measurement space; and \( \sigma \) is the Gaussian smoothing parameter.

PNNs consist of four layers. The first layer consists of the network’s inputs. The second layer is the Pattern Unit Layer. This is fully connected to all of the inputs. Each pattern unit is a small multivariate Gaussian distribution, the mean values of which correspond to a particular training pattern. The summation units sum the multivariate Gaussian distributions of those training patterns belonging to a particular class. Hence, these are only connected to those pattern units belonging to that class. The output units are linear and have only two inputs. It is here that the classification decision is made based on the Bayes decision rule in Equation 3.18. A diagrammatic overview of the structure of a PNN is given in Figure 3.10).
The pattern units are similar to the semi-linear neurons of an MLP except that they use an exponential activation-output function:

$$g(Z_i) = e^{\frac{Z_{i-1}}{\sigma^2}}$$  \hspace{1cm} (3.21)$$

Where $Z_i$ is the activation of the $i^{th}$ node and $\sigma$ is the Gaussian smoothing parameter. The activation is the dot product of the input pattern vector $X$ and the node’s weights, $W$:

$$Z_i = X \cdot W_i$$  \hspace{1cm} (3.22)$$
The vectors X and W are assumed to be normalised to unit length and therefore equivalent to:

\[ e^{-\frac{(w_i-x)^t(w_i-x)}{2\sigma^2}} \] (3.23)

This takes the same form as the definition of Parzen windows.

The summation units simply sum the output of pattern units belonging to the same class:

\[ f_A(X) = \sum_{i \in A} g(Z_i) \] (3.24)

Where A represents class A.

The output units have two inputs and a binary output. The first input is from the summation unit of patterns belonging to that class, \( f_A(X) \). The second input is from a summation unit of non-class members, \( f_B(X) \). Only the second input has a variable weight, \( C \). Hence, the activation of the output units is:

\[ a = f_A(X) - C.f_B(X) \] (3.25)

When known, \( C \) is set to a ratio of prior probabilities, multiplied by the ratio of losses, divided by the ratio of samples:

\[ C = -\frac{h_B l_B \cdot n_A}{h_A l_A \cdot n_B} \] (3.26)

Where \( n_{A_i} \) and \( n_{B_i} \) are the number of training patterns from categories A and B respectively. Where statistical information is not known, \( C \) is used
as an inverter \((C = -1)\). The activation-output function of the nodes in the output unit layer, \(y\), is:

\[
y = \begin{cases} 
+1 & \text{if } a \geq 0 \\
-1 & \text{if } a < 0 
\end{cases}
\]

(3.27)

Training consists of weight prescription at the pattern unit and output unit layers. The weights at the pattern units are set to coincide with the training patterns. PNNs have one remaining parameter to be set: The smoothing parameter, \(\sigma\), affects the network’s PDF estimate. As \(\sigma \to 0\), the network acts as a nearest-neighbour classifier. As \(\sigma \to \infty\), the network creates a non-linear decision boundary approaching the Bayes optimal.

In practise, it is not difficult to find the optimal value for \(\sigma\) and the value can be modified after training [81].

When the prior probabilities and related costs of classes are not known, PNNs are functionally similar to RBF networks. An RBF hidden layer consisting of one node per class, using a Gaussian activation function would behave similarly to the pattern unit layer (depending upon the weight settings). The use of summation units feeding into the output units also is structurally and functionally similar to that of the Perceptron and its use of association units.

The advantage gained using PNNs over other networks is the near instantaneous training time. Additionally, the network can learn new patterns easily by adding new pattern units. However, using one pattern unit for each training pattern results in the large memory requirement. Historically, this
was prohibitive. This limitation is of less concern with modern large and affordable memories.

3.8 Contribution

This Chapter has reviewed the Neural Network literature. The Chapter provided an introduction to the biological neuron and identified the mathematical model counter-parts. The Chapter covered the Threshold Logic Unit; the Perceptron; the Multi-Layer Perceptron; and some popular networks: the Hopfield network; the Kohonen Self-Organising Feature Map; the Radial Basis Function neural networks; and the Probabilistic Neural Network.

The Chapters that follow will describe the investigation conducted for this thesis. The theory of this Chapter and the previous Chapter will be used as a basis on which this investigation builds upon.
Chapter 4

Classification of Symmetry Features

4.1 Introduction

This Chapter describes the research of this thesis into the classification of patterns using their symmetry features. It discusses the choice of a suitable pattern recognition problem. This problem is used throughout the thesis to test the hypotheses and proposed mechanisms. A symmetry transform is selected based on the related work and the literature review conducted in Chapter 2. This transform is used to produce features which are presented to a chosen neural network for classification. Finally, the performance of this classifier is discussed along with its implications for the significance of symmetry features for pattern recognition.
4.2 Motivation

Symmetry is a property of many objects in the world around us. It is often a by-product of simplicity in design and manufacturing. Yet, symmetry is not widely used to assist artificial neural networks to perform pattern recognition. Neural networks applied to pattern recognition tasks fall short of the performance achieved by humans in many cases. It is therefore important to investigate the usefulness or significance of symmetry in pattern recognition.

4.3 Hypothesis

The first hypothesis is that patterns can be classified by their symmetry features. This implies that feature differences between classes are maintained. That is, the feature set which describes a member of one class, do not also describe a member of another class. This hypothesis is based on two premises: that the structure (or shape) of the patterns is sufficient to classify them; and that symmetry is a property directly related to object structure.

The second is that generalisation in pattern recognition is improved when classifying the symmetry features. This is based on the premise that class members share a similar structure and hence share symmetry features. Specifically, it is proposed that symmetry features will not be greatly affected by the variances amongst members of the same class.
4.4 Rationale

In designing pattern recognition systems, features are often selected that best represent the diversity of patterns and classes [15]. Chapter 2 described in detail how symmetry is a feature or property of shapes. In patterns, it is an underlying, hidden feature. It is a property of the pattern, but it is not explicitly presented. It can be measured by examining the structure of the pattern: comparing components, their relative distances and their orientations.

Park et al [74] in evaluating the effectiveness of symmetry transforms on hand labelled images, identified the potential of symmetry detection in object recognition in the same way that symmetry detection plays an important role in human perception. The investigation of this Chapter seeks to test that potential. This investigation transforms the underlying hidden shape description into an explicit description. This new representation is a pattern in itself. The investigation takes the view that the resulting symmetry pattern is more suitable for classification. The rationale behind this is that the patterns vary for many reasons. But, that the variances have little effect on the overall symmetry description of the class representing the pattern. Thus, the symmetry features will vary less amongst class members than the original patterns.
4.5 Related Work

Mancas et al [61] proposed an effective symmetry based mechanism for detecting tumours in human tracheas. The mechanism analyses two dimensional image slices of the trachea. A normal trachea is circular in shape and quite symmetrical. When tumours are present in the trachea, the trachea becomes asymmetrical. Their mechanism detects symmetry in the image slices. The formal classification is based on a symmetry threshold. This is a simple classification problem: there are only two classes. From a symmetry feature perspective, it is also simple. A single symmetry feature is used. This is the measure of reflectional symmetry across the entire image slice. Furthermore, there is a one-to-one mapping of features to class.

Sharvit et al [86] use symmetry for image database indexing. Whilst this is neither a pattern recognition problem nor does it use connectionist theory, it has an important similarity. Their work involves matching hand-drawn sketches and grey scale images to those in a database. This is similar to classification in that the database matches should belong to images of similar objects. Their approach involves describing shapes using a ‘shock’ language. That is, a hierarchy of formal shape deformation descriptions. This description mechanism is not intended for connectionist systems. Whilst the approach does not fit exactly with the use of symmetry features for neural network based classification, their use of symmetry features to match shapes supports the theory that patterns can be classified by their symmetry features.

Hayfron et al [26] use symmetry features for an effective biometric security
system. Their work performs a symmetry analysis of human gait over time. From the perspective of this research, this mechanism is aided by additional information. It is comparable to the difference between offline versus online handwriting recognition. Online recognition aids the classifier because how the pattern was produced is presented in addition to the overall finished shape. Similarly, the change of symmetry of the gait with respect to time is used for classification. Hence this work does not demonstrate that symmetry features can be used to classify patterns. Rather, it only demonstrates that the change in symmetry over time can be used.

Symmetry has been used to assist in the classification of characters of non-Latin based character sets. Premaratne and Bigun [78] performed experiments on the recognition on the 59 characters and 17 modifier symbols of Sinhala script (used by 80% of Sri Lankans.) Sinhala characters are generally more rounded and complex, lacking the straight edges of Latin characters. They classified the characters by their local orientation. This is considered to be a localised, linear symmetry measure because in an ideal neighbourhood, the grey scale values will only change in one direction. From this, the orientation angle and a certainty measure are derived. Frequency domain analysis was employed to measure the symmetry. A high confidence of symmetry signifies a small change in grey scale values (much like the comparison of intensity values at equidistant positions from the potential reflection plane in Heubner’s work [34].) This is a very simple metric, only applicable locally and does not provide the insight into the potential benefits of less localised or global symmetry features for pattern recognition. Hence, more research is required.
Fan et al [17] used symmetry in *coarse* classification of Chinese characters. Their Chinese character set contains 5401 frequently used characters. It is inefficient to classify a character against the complete set of all characters. In such cases it is necessary to pre-classify characters into subsets. The classification of each subset is performed by different classifiers trained on characters belonging only to that subset. In the case of the Chinese character sets, previous work had pre-classified characters into ten subsets [98] based on the positioning of the character radicals. Computation time can be reduced and classification accuracy increased by having many subsets each with small character counts. However, due to the nature of the language and the chosen pre-classification method, some subsets are inevitably larger than others. Fan et al [17] extended this method by testing for bilateral symmetry about four orientations (vertical, horizontal and both diagonals) in a specific target area of each subset. Hence, splitting each of the ten subsets further based on symmetry. Their experiment demonstrated that classification performance using the additional symmetry pre-classification was 97%, up from 93% without it. The method, however, is highly dependent on the pre-classification steps: if they do not assign a character to its required subset correctly, the respective classifier cannot classify it. In this work, the features of symmetry are only used to assign characters to one of a group of classifiers. Traditional classification of the stroke intensities is still performed, rather than the classification of the characters’ symmetry features.
4.6 Classification Problem

To test the hypothesis, a suitable classification task is selected. Such a task has multiple classes each represented by many example patterns. The patterns must vary sufficiently to describe all members of their respective classes. The nature of the task itself must not contribute to the ease with which it can be solved. The objective of the experiment is to test only the performance when using symmetry features.

Offline handwritten digit recognition is chosen. Ten different classes is sufficient to measure classification performance. Furthermore, it does not equate easily to a one-to-one feature to class ratio as that of the experiment conducted by Mancas et al [61]. Being handwritten, the digits vary sufficiently amongst members of the same class. Such a classification task is also a reasonable practical application. Offline recognition permits only the symmetry analysis of the completed shape. Unlike the experiment of Hayfron et al [26], time is not a variable or contributing factor. The performance is measured by calculating the percentage of patterns that are correctly classified.

4.7 Symmetry Features

Four factors must be considered in selecting the symmetry features to be used for classification: How descriptive must the features be in order to distinguish between classes? What types of symmetry do the patterns support? Which types of symmetry are supported by biological evidence or other natural
existential proof? Does the literature propose symmetry features which may be good candidates for classification?

The typical view of symmetry as a discrete property does not lend itself to the classification of patterns by symmetry features. For example, three or more classes cannot be distinguished by an all-or-nothing approach to symmetry, let alone the 10 classes of handwritten digits. A continuous measure of symmetry (possibly of different types) at various points in the pattern is more likely to produce the variation in features required.

Handwritten digits are two-dimensional structures. In this case, a third dimension representing greyscale intensity is also present. Reflectional symmetry could be measured in one or two dimensions. In two dimensional space, the reflectional symmetry can exist at various orientations (other than simply the horizontal or vertical). It could either be measured at particular orientations or for all. Circular rotational symmetry can also be measured.

From a biological perspective, Palmer and Hemenway [71] showed that humans quickly detect reflectional symmetries but take longer to identify rotational symmetries. Locher and Nodine [56] showed that when scanning a shape, the eye makes efficiencies where reflectional symmetries are identified. Biology supports the use of two-dimensional reflectional symmetry.

The related work described in this Chapter also use two-dimensional reflectional symmetries. Hayfron et al used the Generalised Symmetry Transform (GST) [26] at various time indexes combined. The GST calculates a continuous measure of the reflectional symmetry magnitude and its orientation at each point in the image. It is these two sets of symmetry features that is used to test the hypothesis in this experiment.
4.8 Experiment

The experiment will compare the recognition rates of symmetry feature based classifiers with the that of the traditional (control) features. The network sizes depend on the pattern sizes, which in turn is dependent on the chosen symmetry transform. Figure 4.1 depicts the proposed pattern recognition system process. A symmetry transform will produce symmetry features from an input image. These features are normalised prior to being presented to a neural network for classification.

Figure 4.1: Overview of the proposed pattern recognition system structure used in the experiment.

4.8.1 Symmetry

The symmetry information produced by the GST has two components. The first is the symmetry magnitude at each point. The second is the orientation of the most significant contribution to the symmetry magnitude. This investigation will test the use of these components separately and together. Using both components increases the dimensionality but has the benefit of providing twice as much information.
4.8.2 Classifier

Multi-Layer Non-Linear Feed-Forward networks are traditionally used for handwritten digit recognition [54] although non-neural network mechanisms are also effective. (These neural networks are typically incorrectly labelled as Multi-Layer Perceptrons (MLPs) in the literature. The term is incorrect because they do contain association units.) Le Cun et al [46] have conducted experiments using fully-connected and hand-crafted networks. Probabilistic Neural Networks (PNNs) have been shown to outperform MLPs in optical character recognition tasks [90]. The additional main advantage is that their training is trivial and near instantaneous [4]. For this recognition task, the size of a PNN is significantly larger than an MLP, it is not prohibitively so. PNNs also have only one parameter: the PNN smoothing parameter which has to be adjusted; although it is not usually difficult to find a suitable value [63]. The performance of an MLP, on the other hand, is dependent on the learning rate, moment term and shape of the sigmoids in order to avoid local minima. This multi-variate optimisation would have added a complexity to the experiment. The success of which would have directly interfered with the overall performance. It would not have been known whether the performance was the result of the good or poor optimisation, or the result of using the symmetry features. Thus in using PNNs, the outcome of the experiment can be more reliably attributed to the use of the symmetry features.

As described in Chapter 3, the size of the network depends upon the size of the training set.

The performance of the classifier is measured by how successfully it clas-
sifies patterns. The recognition rate, $r$, by pattern set $P$, is calculated using:

$$r_P = \frac{c_P}{|P|}$$

(4.1)

where $c_P$ is the number of patterns in $P$ correctly classified, and $|P|$ is the number of patterns in the set.

### 4.8.3 Training and Test Patterns

The experiment uses the United States Postal Service (USPS) Data Set. It was created by collecting samples from envelopes passing through the Buffalo, New York Post Office. It consists of 9298 patterns and labels split into two sets: a training set and a test set. The training set consists of 7291 patterns. These are presented to the network to train the weights. A further 2007 patterns are reserved for measuring the performance of classifiers.

Each pattern consists of a 16-by-16 pixel grey scale image. The data set is commonly used to test handwritten digit recognition [87, 83, 88, 94, 39, 13].

The size of each pattern in the original USPS data set is 16-by-16 pixels, for a total of 256 inputs for the network. These are presented unprocessed as a control set with which to compare the performance limitations imposed by the network architecture on the symmetry feature based classification.

The GST estimates the gradients in the pattern. This is accomplished by convolving (in two dimensions) the 16-by-16 pixel image with a 3-by-3 matrix. The result of the convolution is a matrix of increased size. Each dimension increases by two pixels. Hence, the output of the GST on each pattern is a 18-by-18 pixel image for a total of 324 inputs for the network.
The size of the networks can now be deduced. The control network is $256 \times 7291 \times 20 \times 10$, for nodes in the input, pattern, summation and output layers respectively. The magnitude only test uses a network of size $324 \times 7291 \times 20 \times 10$. The orientation feature only test uses a network of size $648 \times 7291 \times 20 \times 10$. The magnitude and orientation feature test uses a network of size $972 \times 7291 \times 20 \times 10$.

### 4.8.4 Parameters

Two parameters control the experiment: the Probabilistic Neural Network smoothing parameter; and the maximum distance parameter in the Generalised Symmetry Transform’s Distance function. Confusingly, both use the same symbol in the literature, $\sigma$. This is not coincidental. Both define the variances in the Gaussian functions in the respective systems. To avoid further confusion, the GST distance function parameter is denoted by $\sigma_{\text{GST}}$; the PNN’s by $\sigma_{\text{PNN}}$.

**Generalised Symmetry Transform Distance-weight Function**

In the case of the GST distance-weight function, the parameter controls the significance of contributions from point pairs according to the distance between the two points. When this parameter is less than the greatest possible distance between two points, some contributions will be weighted such that the contribution is insignificant. Therefore, symmetry information provided by these points would be lost. The approach of this initial work is to prevent loss of potential valuable information. Hence, the Gaussian of the distance-
weight function is shaped such that the variance is equal to the greatest
distance between two points. This can be calculated using Pythagoras’ the-
orem:

\[ \sigma = \sqrt{h^2 + w^2} \]  

(4.2)

For a given input retina’s height \( h \), and width \( w \). For the 16-by-16 inputs
in this experiment, the variance of the Gaussian in the GST is set to \( \sigma_{\text{GST}} = 23 \), the nearest whole integer to \( \sqrt{512} \).

Probabilistic Neural Network Smoothing Parameter

The remaining parameter, the PNN smoothing parameter, is varied across
a range between 0.1 and 2.0. Comparing the performance of the classifier
against the value of the smoothing parameter yields insight into the pattern-
feature space.

4.8.5 Normalisation

The pixel intensities have the continuous range \(-1\) to \(1\) inclusive in the pat-
terns of the control network. The symmetry magnitude maps are normalised
to the same range using Algorithm 4.1. Because of their circular nature, each
element of the orientation map is presented to the network twice: once as a
function of cosine, once as a function of sine. Hence, these inputs are also
normalised to the continuous range \(-1\) to \(1\). It is important to use the same
input range in order to fairly compare and contrast the network.
Algorithm 4.1 Intensity Normalisation Algorithm

```
UpperIntensity = 1
LowerIntensity = -1

// Determine pre-normalisation set range
minimum = min(trainSet)
maximum = max(trainSet)

// Reduce to 0,1
trainSet = trainSet - minimum
testSet = testSet - minimum
maximum = maximum - minimum

// Change to UpperIntensity,LowerIntensity
range = UpperIntensity - LowerIntensity
trainSet = trainSet * range
testSet = testSet * range
trainSet = trainSet + LowerIntensity
testSet = testSet + LowerIntensity
```

4.9 Results

The peak test set recognition rates (generalisation performance) were: 95.2% for the traditional feature control set; 92.5% for the normalised symmetry magnitude feature set; 90.9% for the normalised symmetry orientation feature set; and 92.8% for the normalised symmetry magnitude and orientation feature set. The results are tabulated in Table 4.2 and visually depicted in Figure 4.2.

The circular nature of the orientation variates required the normalisation of the data. The performances above can be fairly compared because they are all normalised to the same input range. However, the symmetry magnitude need not be normalised. The magnitude normalisation test demonstrates that the peak recognition rates are not significantly affected by the normalisation. The peak performances were achieved at $\sigma_{\text{PNN}} = 1.5$ for the non-normalised
Table 4.1: Experimental results comparing generalisation performances across the four experimental networks: the traditional, control network; symmetry magnitude features network; symmetry orientation features network; and both symmetry magnitude and orientation features network. Peak generalisation performances are shown in bold. Generalisation performance is measured as the recognition rate on the test set of patterns. The control data used was the original USPS patterns with out the application of any symmetry transforms, as is common in the application of neural networks in character or digit recognition. (Tabulation of data in Figure 4.2)

<table>
<thead>
<tr>
<th>σ</th>
<th>Control</th>
<th>Normalised Symmetry Features</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Magnitude</td>
<td>Orientation</td>
<td>Both</td>
</tr>
<tr>
<td>0.1</td>
<td>53.3</td>
<td>92.3</td>
<td>17.9</td>
<td>17.9</td>
</tr>
<tr>
<td>0.2</td>
<td>92.7</td>
<td>92.4</td>
<td>21.2</td>
<td>18.9</td>
</tr>
<tr>
<td>0.3</td>
<td>95.0</td>
<td>92.5</td>
<td>41.2</td>
<td>25.0</td>
</tr>
<tr>
<td>0.4</td>
<td>95.1</td>
<td>91.1</td>
<td>76.2</td>
<td>32.2</td>
</tr>
<tr>
<td>0.5</td>
<td>95.2</td>
<td>89.2</td>
<td>90.8</td>
<td>41.4</td>
</tr>
<tr>
<td>0.6</td>
<td>95.2</td>
<td>85.3</td>
<td>90.9</td>
<td>52.6</td>
</tr>
<tr>
<td>0.7</td>
<td>95.2</td>
<td>81.2</td>
<td>90.9</td>
<td>66.9</td>
</tr>
<tr>
<td>0.8</td>
<td>95.0</td>
<td>78.5</td>
<td>90.9</td>
<td>80.0</td>
</tr>
<tr>
<td>0.9</td>
<td>95.0</td>
<td>74.3</td>
<td>90.9</td>
<td>88.4</td>
</tr>
<tr>
<td>1.0</td>
<td>95.0</td>
<td>71.6</td>
<td>91.0</td>
<td>91.6</td>
</tr>
<tr>
<td>1.1</td>
<td>94.9</td>
<td>69.4</td>
<td>91.0</td>
<td>92.5</td>
</tr>
<tr>
<td>1.2</td>
<td>94.8</td>
<td>67.2</td>
<td>91.0</td>
<td>92.8</td>
</tr>
</tbody>
</table>

set and \( \sigma_{\text{PNN}} = 0.3 \) for the normalised set. The results are tabulated in Table 4.2 and visually depicted in Figure 4.3.

### 4.10 Discussion

Simard et al [87] reported a USPS data set human performance error rate of 2.5\% (or a recognition rate of 97.5\%). Various classifiers have been tested in the literature, their recognition rates are: relevance vector machine, 94.9\%
Figure 4.2: Graph showing the relationship between the PNN smoothing parameter and the comparative generalisation performances the traditional feature recognition, and symmetry feature recognition using magnitude and orientation features separately; and both classifying both symmetry feature types. Generalisation performance is measured as the recognition rate on the test set of patterns. The traditional method outperforms the classification of symmetry features. The magnitude features outperform the orientation features, but performs worse than using both magnitude and orientation features.
Figure 4.3: Comparison of pattern recognition using normalised versus non-normalised symmetry magnitude features. The peak performances are identical for both the training and the test sets. Normalising the inputs of the patterns reduces the distances between the multi-variate Gaussian distributions in pattern space.
Table 4.2: Data to show the effect of classifying normalised versus non-normalised symmetry magnitude features. Peak recognition performances are highlighted in bold, where the smoothing parameter maximises both test and training set performances. (Tabulation of data in Figure 4.3)

<table>
<thead>
<tr>
<th>( \sigma )</th>
<th>Non-normalised</th>
<th>Normalised</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Training Set</td>
<td>Testing Set</td>
</tr>
<tr>
<td>0.1</td>
<td>100.0</td>
<td>25.9</td>
</tr>
<tr>
<td>0.2</td>
<td>100.0</td>
<td>29.8</td>
</tr>
<tr>
<td>0.3</td>
<td>100.0</td>
<td>35.1</td>
</tr>
<tr>
<td>0.4</td>
<td>100.0</td>
<td>45.0</td>
</tr>
<tr>
<td>0.5</td>
<td>100.0</td>
<td>58.6</td>
</tr>
<tr>
<td>0.6</td>
<td>100.0</td>
<td>71.3</td>
</tr>
<tr>
<td>0.7</td>
<td>100.0</td>
<td>82.7</td>
</tr>
<tr>
<td>0.8</td>
<td>100.0</td>
<td>88.7</td>
</tr>
<tr>
<td>0.9</td>
<td>100.0</td>
<td>91.4</td>
</tr>
<tr>
<td>1.0</td>
<td>100.0</td>
<td>92.2</td>
</tr>
<tr>
<td>1.1</td>
<td>100.0</td>
<td>92.4</td>
</tr>
<tr>
<td>1.2</td>
<td>100.0</td>
<td>92.4</td>
</tr>
<tr>
<td>1.3</td>
<td>100.0</td>
<td>92.4</td>
</tr>
<tr>
<td>1.4</td>
<td>100.0</td>
<td>92.4</td>
</tr>
<tr>
<td>1.5</td>
<td>100.0</td>
<td>92.4</td>
</tr>
<tr>
<td>1.6</td>
<td>99.0</td>
<td>92.4</td>
</tr>
<tr>
<td>1.7</td>
<td>99.0</td>
<td>92.4</td>
</tr>
<tr>
<td>1.8</td>
<td>99.0</td>
<td>92.4</td>
</tr>
<tr>
<td>1.9</td>
<td>99.0</td>
<td>92.3</td>
</tr>
<tr>
<td>2.0</td>
<td>99.0</td>
<td>92.3</td>
</tr>
</tbody>
</table>
feed-forward neural network, 95.8% [46]; invariant support vectors, 97.0% [83]; nearest neighbour classifier, 94.4% [39]; and statistical classifiers with two-sided tangent distance, 97.0%, [39]. Other classifiers yield better results but either had an extended training set or used to the testing set for training [13].

The experiment control consisted of classifying the USPS data set without performing the symmetry transform, in line with other accepted works [54, 53]. The PNN used as the control in this experiment yielded a recognition rate of 95.2%. This performance is very similar to the two closest matching classifiers: the feed-forward neural network (95.8%) [46] and the nearest neighbour classifier (94.4%) [39].

When comparing the non-normalised and normalised symmetry magnitude classification performances, no significant difference arises. The slight variation is likely the result of not scanning sufficiently small increments between values of the PNN smoothing parameter. Normalising the symmetry magnitude reduces the range of input values and requires a smaller smoothing parameter value. This indicates that the multi-variate Gaussian functions representing training patterns are closer in pattern space when normalised. More importantly, it indicates that the performance of the normalised magnitude features is not affected and hence can be compared to the other symmetry feature sets.

None of the three sets of symmetry features yielded a classifier that improved upon the generalisation performance of the original, traditional feature classifier (control); nor upon human performance. This result does not support the second hypothesis. Out of the three symmetry feature sets, the
set using both the symmetry magnitude and orientation features performed generalisation best. This was anticipated as it used more information than the remaining two sets.

The symmetry magnitude set outperformed the symmetry orientation set. Interestingly, the performance difference between using both features and using only the magnitude features was small: a percentage point difference of 0.3. The symmetry magnitude patterns are each one third the size of the patterns using both magnitude and orientation. And as a result, the number of input nodes in the neural network classifying magnitude features is also one third the number of that classifying both. Serial implementations of such networks will require less time to classify magnitude-only patterns; true parallel implementations will require less processing elements. In addition, the feature extraction computational overhead can be further reduced by not calculating or storing the symmetry orientations. This leads to the proposal that if symmetry features are to be used, only the symmetry magnitude features should be classified.

The recognition rates on the symmetry feature training sets reached 100%. This indicates that the pattern classes are sufficiently different as to differentiate between them. This supports the first hypothesis.

Though the generalisation performance was reduced using symmetry features, the difference was only 2.6 percentage points. This suggests that symmetry features are worth investigating. Had the performance been less than 50%, it would have suggested the features were not playing a useful role. But this is not the case. The small difference suggests that useful features or components are lost as the result of the symmetry transform.
4.11 Contribution

This chapter investigated the classification of patterns by their symmetry features. These patterns were of an ‘offline’ nature, that is no temporal information was included. A method was presented for classifying patterns using normalised features produced using the Generalised Symmetry Transform. Though its generalisation performance was lower than that of the traditional method, this classifier could classify and generalise patterns. This initial experiment suggests symmetry features can be useful in pattern recognition and merit further research.
Chapter 5

Position Invariance

5.1 Introduction

In the previous chapter, the symmetry features of patterns were presented to a neural network for classification. In this chapter, symmetry features will be used to provide information about the patterns. However, these features will not be presented to the network. Rather, the features will be used to modify the original pattern prior to presenting it to a network.

5.2 Hypothesis

The research in this chapter tests the hypothesis that symmetry features of patterns can be used to normalise their position prior to their classification; and that this will aid the generalisation of previously unseen patterns.
5.3 Motivation

The motivation for researching this hypothesis is that there are two potential benefits associated with a classifier using the symmetry features in this manner.

The first potential benefit is to make the classification method positionally invariant. That is, the input pattern may be located anywhere on the input retina. After the normalisation process, the pattern will be repositioned ready for classification. Hence a neural network that cannot normally tolerate variations in position will be able to classify successfully.

The second potential benefit is to assist the network in the pattern space. Some patterns share common features which may be useful in their definition or description. Using characters as an example, consider the lowercase letters ‘p’, ‘q’, ‘b’ and ‘d’. Common to all is a circular component, ‘o’, and a line component ‘l’. The four letters differ in the placement of the line with respect to the circular component.

5.4 Rationale

Common spatial normalisation methods yield patterns which may not be assisting the classifier. One such example scans the retina and produces an input pattern in which the character occupies as much of it as possible. This is similar to a method described in [54, 53]. The output of such a method, for the four example letters, would be similar to that in Figure 5.1.

To most neural networks, common features across several patterns are
Figure 5.1: Output of a common method for spatial normalisation prior to classification for four characters sharing two common features or subcomponents. Each character can be distinguished by the relative positions of the sub components.
defined in terms of shared values at the input node level. In Figure 5.1, the line sub components of the letters will produce similar node inputs for the letter ‘b’ with ‘p’ and separately for ‘d’ with ‘q’. The circular sub components will produce similar node inputs for the letter ‘b’ with ‘d’ and separately for ‘p’ with ‘q’. With the exemption of the overlap of the line and circular sub components: Each one of the four above letters shares features with only two others. For example, ‘b’ has similar inputs to ‘d’ and ‘p’. Overlaying all four characters from Figure 5.1 produces Figure 5.2.

Normalising the position of the characters may yield input patterns that differ significantly. Circles support an infinite number of reflection planes.
Consequently, the output of a symmetry transform on a circle will produce a magnitude map with a maximum at the centre of the circle where all supported reflection planes overlap. Consider the normalisation of position of the characters such that point of maximum symmetry magnitude lies at the centre of the image. Common input node values will occur on the circular component. This will be shared amongst each of the four letters. In addition, each would have a unique set of input node values where the line segment of the character is positioned in the image. This is unlike the typical case example described above. To the neural network, all four share a common feature and yet each has its own unique feature. Could this aid recognition? Figure 5.3 depicts this. This intuitive example is specific to the four characters. But normalising patterns with respect to their symmetry features may still aid pattern recognition in non-obvious ways.

5.5 Position Invariance for Pattern Recognition

The basic idea is to make use of a continuous measure of symmetry to provide information with which to transform a pattern containing a foreground object. The result of the transformation is an image that is identical regardless of the object’s original position in the pattern.

Consider an optical character recognition (OCR) system. A neural network performs classification on a set of individual characters (rather than words). The characters are input to the network in either a binarized or
Figure 5.3: Potential alignment of characters by normalising their position based on the location of the point with the greatest symmetry magnitude.
intensity-range limited form. Each class is represented by an output node which is activated if the input matches that class. Limited positional variances may be tolerated by hidden layers with connections limited to a small neighbourhood. Not all neural networks are structured in this way. Some may not tolerate even small variances in position. Such systems can be improved using the symmetry features of the input character. The symmetry features for the input image are calculated and a central symmetry position is found. The image is then transformed so that the central symmetry position is at the centre of the image. The network performs the classification on this transformed image. Thus large variations in position are removed prior to being presented to the network.

5.6 Transformation of Image

Positional invariance is achieved by repositioning all the pixels in an image before presenting it to a neural network. The aim of this transformation is to present the neural network with the same input pattern regardless of the original position of the object within the input space.

All pixel transformations presented here are based on affine transformation in Euclidean space. Affine transformations have the general form:

\[
\begin{bmatrix}
  x' \\
y'
\end{bmatrix} = A \times \begin{bmatrix}
  x \\
y
\end{bmatrix} + B
\]  \hspace{1cm} (5.1)

Where A is a 2-by-2 matrix defining rotation, scaling and shear; and B is a column matrix defining shift. In order to reposition the pixels in the
image, only the B matrix is defined, which is equivalent to:

\[
A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, B = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}
\] (5.2)

A symmetry analysis of the input image identifies a Focal Point \((f_1, f_2)\). The pixels are transformed such that the focal point is now located at the centre of the image, \(C = (c_x, c_y)\). For such a transformation, the B matrix is defined as:

\[
\begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} c_x - f_x \\ c_y - f_y \end{bmatrix} \] (5.3)

When the input and the output images of this transformation have the same dimensions, pixels will not be outside the image boundary if:

\[
0 \leq x + b_1 \leq w \] (5.4)

and:

\[
0 \leq y + b_2 \leq h \] (5.5)

Where \(w\) and \(h\) are the image’s width and height respectively. Where this behaviour is not desired, the modular arithmetic operator can be applied to prevent loss of information:

\[
\begin{bmatrix} x'' \\ y'' \end{bmatrix} = \begin{bmatrix} x' \mod w \\ y' \mod h \end{bmatrix} \] (5.6)
Thus, the image appears to ‘wrap around’ the dimension limits imposed on the two axes.

The focal point must be unique with respect to the relative position within the structure of the object to be classified, such as a single machine printed character. The hypothesis is that the symmetry features can provide a meaningful focal point.

5.7 Focal Point

It is proposed to use the co-ordinates of a point representing the most significant degree of reflectional symmetry in the pattern. This insight is aligned to the fact that the presence and orientation of reflectional symmetry is quickly identified by humans [5], whereas the detection of rotational symmetry requires considerably more effort. Patterns (such as characters, faces) are often represented in at least two dimensions. Additional dimensions may be used for grey scale or colour.

The Generalised Symmetry Transform acts on grey scale images. For each point in the image, a continuous measure of reflectional symmetry and its orientation is calculated:

\[ S_\sigma(p) = [M_\sigma(p), \phi(p)] \]  \hspace{1cm} (5.7)

Hence the GST is a suitable transform for extracting useful symmetry features for position normalisation. A point \( p \) is used as the focal point, where \( p \) has the greatest symmetry magnitude in the pattern (or image) \( I \).
\[ \forall i \in I : M_\sigma(p) > M_\sigma(i), p \neq i \quad (5.8) \]

The greatest magnitude is likely to occur where two or more reflection planes intersect. Where only one reflection plane is present, the greatest magnitude is likely to occur at the midpoint of the plane. The greatest symmetry magnitude will occur at the centroid for objects that are perfectly symmetrical. For objects that are not perfectly symmetrical as a whole, but have symmetrical components, this will occur in the centre of the symmetrical region. For objects lacking symmetry, this will occur at the best fitting position.

A traditional search algorithm is used to identify the point with the largest symmetry magnitude.

5.8 Experiment

The experiment will compare the recognition rates of a traditional handwritten digit recognition system and a similar system equipped with the position invariance mechanism proposed earlier in this chapter. In particular, each system’s ability to cope with variances in position are tested.

Figure 5.4 depicts the proposed pattern recognition system process. A symmetry transform produces symmetry features. These are used to transform the position of the features in the original pattern. These are presented to a network for classification.
5.8.1 Patterns

The original United States Postal Service (USPS) data set consists of 9298 patterns each representing a handwritten digit (the numerals zero to nine inclusive). In order to test the effectiveness of the position normalisation method, the input dimensionality is increased from 16-by-16 pixels to 32-by-32 pixels. The training set consists of 7291 patterns. Each has 16-by-16 original pixels located centrally, surrounded by a border of 8 pixels. These patterns are then processed by the position normalisation algorithm. The output of this transform is used to train the classifier.

5.8.2 Symmetry Transform

The Generalised Symmetry Transform [79] described in §2.6.2 and used in the previous experiment (of Chapter 4) is used in this experiment. The GST distance-weight parameter, $\sigma_{GST} = 23$. Though the pattern’s size has increased, the size of the object contained in the pattern has not. By keeping $\sigma_{GST}$ constant, the effect of increasing the pattern’s size on the classifier’s
performance can be measured relative to the results of the previous experiment.

The orientation of symmetry at each point is not used in this experiment. To ensure no programming errors are introduced, the same algorithm is employed. However, since this information is ultimately discarded, researchers repeating this experiment can reduce computation time by calculating only the magnitude.

5.8.3 Position Normalisation Method

The experiment uses the affine based transformation mechanism described in §5.6 using the position of the point with the greatest symmetry magnitude as the focal point. No wrap-around method is used. To avoid issues of intensity values not aligning directly with pixel positions, only integers are used in the affine transform. Hence no intensity interpolation method is required or used.

5.8.4 Intensity Normalisation

The USPS data set is already normalised to the continuous range $[-1, 1]$. Hence, colour intensity normalisation is not employed. The position normalisation mechanism does not affect the range of colour intensity values.

5.8.5 Classifier

Two Probabilistic Neural Networks provide a control and an experimental classifier. The control network is trained on the original features placed cen-
trally on the enlarged pattern. These patterns are not transformed in any other way prior to training. The experimental classifier is trained on the original features placed centrally on the enlarged pattern, which are transformed to normalise the position prior to classification.

Due to the size of the patterns (§5.8.1), the networks’ input unit layer size increased to 1024 nodes, from 256 nodes in Chapter 4. The pattern unit layer has 7291 nodes; 20 nodes in the summation unit layer; and 10 nodes in the output unit layer.

The PNN smoothing parameter, $\sigma_{\text{PNN}}$ is set based on the optimum performances identified by the results of the previous experiment (§4.9). Hence, $\sigma_{\text{PNN}} = 0.7$.

### 5.8.6 Performance Measurement

A position offset distance is defined as the difference in position in both the $x$ and $y$ dimensions. For example, a position offset distance of 1 is a difference in position by 1 pixel in both dimensions. That is, for a position offset distance $d$:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} d \\ d \end{bmatrix} \tag{5.9}$$

By varying the position offset distance of the original patterns, normalising and then classifying them, the ability to consistently normalise the position can be tested. To test the classifier’s generalisation performance using the position normalised data, 2007 previously unseen digits are pre-
sented. Like the training patterns, they are positioned centrally and then normalised.

A second classifier is trained and tested using a control data set. This is to check that the invariance to changes in position are the result of the normalisation method, and not an attribute of the network. Furthermore, a network that is known to tolerate variances in position is avoided. The Probabilistic Neural Network does not limit connectivity to neighbourhoods between layers, as in the case with the Neocognitron or handcrafted Multi-Layer Feed-Forward networks.

5.9 Result

The optimal performance of the control classifier is 95.1%, compared to that of the position normalised classifier of 94.7%. The rate at which the performance decreases with respect to the position offset distance is depicted in Figure 5.5 and tabulated in Table 5.1.

5.10 Discussion

Increasing the dimensionality of the input space did not affect the performance of the classifier. The control network’s test set performance was equal to that in the previous chapter. Normalising the position of the training data reduced the performance slightly, but not significantly at only a percentage point difference of 0.4. However, this demonstrates that the normalisation of position with respect to symmetry features has not aided classification in
Figure 5.5: Graph showing the effect of shifting the object position with respect to the recognition rate. As the position offset distance increases, the object moves diagonally across (and away from) the centre of the pattern. Beyond 8 pixels, the object is occluded by the boundary of the pattern.
Table 5.1: Recognition rates for the two networks, listed by data sets, against the Position Offset Distance.

<table>
<thead>
<tr>
<th>Distance</th>
<th>Recognition Rate, %</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Control</td>
<td>Position Normalised</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Training Set</td>
<td>Test Set</td>
</tr>
<tr>
<td>0</td>
<td>100.0</td>
<td>95.1</td>
<td>100.0</td>
</tr>
<tr>
<td>1</td>
<td>30.8</td>
<td>29.7</td>
<td>100.0</td>
</tr>
<tr>
<td>2</td>
<td>16.3</td>
<td>16.8</td>
<td>100.0</td>
</tr>
<tr>
<td>3</td>
<td>12.3</td>
<td>13.8</td>
<td>99.5</td>
</tr>
<tr>
<td>4</td>
<td>12.8</td>
<td>15.7</td>
<td>98.9</td>
</tr>
<tr>
<td>5</td>
<td>16.9</td>
<td>18.6</td>
<td>97.7</td>
</tr>
<tr>
<td>6</td>
<td>19.9</td>
<td>20.5</td>
<td>94.9</td>
</tr>
<tr>
<td>7</td>
<td>18.5</td>
<td>19.6</td>
<td>85.9</td>
</tr>
<tr>
<td>8</td>
<td>14.9</td>
<td>17.0</td>
<td>73.7</td>
</tr>
<tr>
<td>9</td>
<td>15.6</td>
<td>17.3</td>
<td>61.9</td>
</tr>
<tr>
<td>10</td>
<td>16.1</td>
<td>18.2</td>
<td>51.2</td>
</tr>
<tr>
<td>11</td>
<td>15.8</td>
<td>18.3</td>
<td>39.8</td>
</tr>
<tr>
<td>12</td>
<td>14.1</td>
<td>15.6</td>
<td>28.0</td>
</tr>
<tr>
<td>13</td>
<td>11.7</td>
<td>12.3</td>
<td>19.7</td>
</tr>
<tr>
<td>14</td>
<td>10.5</td>
<td>11.1</td>
<td>7.6</td>
</tr>
<tr>
<td>15</td>
<td>10.0</td>
<td>9.9</td>
<td>6.6</td>
</tr>
<tr>
<td>16</td>
<td>10.0</td>
<td>9.9</td>
<td>7.6</td>
</tr>
</tbody>
</table>
this case.

The control network was not tolerant of positional variances. Offsetting the pattern by only one pixel in both dimensions reduced the performance to only 30.8%. Whereas, using symmetry based position normalisation, the performance remained unchanged up to a distance of three pixels and then dropped slightly for three pixels. After six pixels, the rate of recognition-rate-loss with respect to the position offset distance increased.

A significant drop is to be expected beyond an offset distance of eight pixels. At this point, there is no remaining padding around two edges of the pattern. Further increasing the position offset distance forces parts of the object to move beyond the boundary of the pattern and become occluded. These occluded components cannot be recovered. Loss of performance is to be expected. Interestingly, despite the occlusion, the classifier’s degrading performance is fairly gradual.

The drop in performance occurs earlier than anticipated. This is attributed to the method of estimating the gradient in the image as part of the Generalised Symmetry Transform. A two-dimension convolution of the image with a 3-by-3 matrix applied to the visible area of the pattern is the cause.

The results show that an affine transformation using information from the symmetry features in the pattern is effective as a position normalisation mechanism for classification. In particular, the position of the greatest symmetry magnitude is a useful feature to identify. However, generalisation was not improved by positioning objects according to their symmetry features. This demonstrates that repositioning alone, at least, is not enough. However,
generalisation may be improved by normalising other spatial differences such as rotation.

5.11 Contribution

This Chapter has proposed an original mechanism for normalising the position of objects in patterns. It uses the location of symmetry features as parameters in an affine transformation. When used, a neural network classifying the resulting patterns is invariant to changes in object position in the pattern.

This investigation has provided an insight into the usefulness and importance of symmetry features. In particular, the position of the point with the greatest reflectional symmetry magnitude. This feature is invariant to positional transforms. That is, the relative position of this feature with respect to the object never changes. And this chapter proved its suitability as a focal point.

The main departure from related methods is the pattern’s symmetry properties. Other methods are applied to patterns known to have identical symmetric features. The features are shared across all patterns in all classes. The proposed mechanism’s performance is unaffected by how symmetric the feature is. That is, the classes of patterns do not need to share symmetric features.

This Chapter presented a hypothesis concerning the repositioning of objects whose classes do not share symmetric features: that generalisation performance would increase because the new positions would diverge the classes.
in pattern space. However, the results of the investigation conducted in this Chapter were that this is not the case. However, only one particular pattern recognition problem was considered and it cannot be said to be true or false of other problems.
Chapter 6

Orientation Normalisation

6.1 Introduction

In Chapter 5, patterns were positionally normalised using information about the reflectional symmetry in the pattern. Affine transformations mapped pixels such that the point with the greatest symmetry magnitude was positioned directly in the centre of the image. An experiment demonstrated the usefulness of such image processing: patterns placed anywhere in the image are normalised yielding a pattern much more similar to that previously seen by a classifier. Hence a classifier that is not usually capable of tolerating differences in position can still perform adequately. The only limiting factor is that the entire object must be represented in the pattern.
6.2 Hypothesis

The research in this chapter is concerned with the hypothesis that symmetry features can be used to classify patterns with invariance to rotation. Specifically, that the orientation of the point that has the greatest symmetry magnitude (in the pattern) can be used to orientate the pattern to a common normal.

6.3 Motivation

The motivation for researching this hypothesis is that there are two potential benefits of a classifier using symmetry features in this manner.

The first potential benefit is that a neural network that cannot usually tolerate variances in rotation would be capable of doing so. Combining this with the research on position normalisation would produce a classifier invariant to position and rotation.

The second potential benefit is to assist the network in the pattern space in a similar manner as proposed in Chapter 5. Orientating the patterns by their symmetry features may increase the Hamming distance between patterns of different classes. Assuming that patterns within a class share symmetry features, the Hamming distance between patterns of the same class may decrease. That is, the pattern variance will increase between classes and decrease within classes.

Whilst there are potentially two benefits of this research, the normalisation of orientation presents a problem which must be solved. Some classes
of patterns are defined as being equal to another class but under rotation. In the case of character recognition, ‘d’ and ‘p’ have the same structure but differ in rotation of 180 degrees. The same applies to ‘b’ and ‘q’. If the orientation is normalised, the pattern presented to the network will be the same for both ‘d’s and ‘p’s. A method for normalising orientation is unsuitable if a classifier is then unable to distinguish between such classes.

6.4 Related Work

Using symmetry features to normalise the orientation of a pattern prior to classification is not new. Di Gesú et al normalised the orientation of faces prior to recognition [21]. The eyes on the face were detected by performing a symmetry transform on the image. This detected rotational symmetries. Their mechanism used the fact that the eyes are usually level with each other on the face. The orientation was normalised such that a line connecting the two eyes was parallel to the horizontal axis. The mechanism proposed here differs in several significant ways.

There are no assumptions about the types or quantity of symmetry present in each and every object in the new mechanism. Di Gesú et al relied on the fact that each face has two eyes, which have a very high degree of rotational symmetry. And that the eyes could be normalised to one orientation shared by all patterns (i.e. all faces).

In this work, it is expected that quantity, degree, position and orientation of symmetry features differs from pattern to pattern. It is hypothesised that patterns within a class share more of these features than they differ from
patterns in other classes. Hence, the mechanism is not searching for known features. A direct result of this is that the mechanism could be applied to other pattern recognition problems. Di Gesú et al.’s method is limited to faces. This mechanism is designed for all two-dimensional (image based) patterns.

The proposed mechanism does not aim to normalise the orientation to a conceptual norm. In the cases of faces, the normal orientation is when the eyes are level, and above the nose, which in turn is above the mouth. It is true that characters do have a specific orientation according to the language that defines them. However, when applied to OCR, this mechanism does not aim to normalise the characters back to that original orientation. The aim of normalising the highest degree of symmetry is to create a wider difference between different classes and reduce the difference amongst patterns belonging to the same class. In this way, it is hypothesised that generalisation performance may improve over the original orientation input.

6.5 Symmetry Orientation

The proposed normalisation mechanism is similar to that proposed in Chapter 5. The patterns are analysed for their symmetry features. Using the symmetry information, the pixels of the original pattern are transformed spatially. The transformed pattern is then presented to the network for classification.

Recall that pixel transformations using affine transformations in Euclidean space have the general form:
\[
\begin{bmatrix}
x' \\
y'
\end{bmatrix} = A \times \begin{bmatrix} x \\ y \end{bmatrix} + B
\] (6.1)

Pure rotational transformations are performed using:

\[
A = \begin{bmatrix}
\cos \theta & -\sin \theta \\
\sin \theta & \cos \theta
\end{bmatrix},
B = \begin{bmatrix} 0 \\ 0 \end{bmatrix}
\] (6.2)

Where \( \theta \) is the angle of the rotation. A suitable value for \( \theta \) must be derived from a symmetry transform. Reflectional symmetry is chosen due to evidence that animals are capable of readily detecting it. The angular information associated with reflectional symmetry is the orientation of the reflection plane. To calculate this, a transform must be a two-dimensional operator. The Generalised Symmetry Transform takes an image, \( I \). For each point \( p \in I \), the transform yields the symmetry at point \( p \), \( S_\sigma(p) \). \( S_\sigma(p) \) consists of a vector, such that:

\[
S_\sigma(p) = [M_\sigma(p), \phi(p)]
\] (6.3)

Where \( M_\sigma(p) \) is the magnitude and \( \phi(p) \) is the orientation. It is proposed to use the orientation of the best fitting reflection plane in the pattern. That is, set the rotation to the orientation of the point with the greatest symmetry magnitude:

\[
\theta = \phi(p)
\] (6.4)

where:
\( \forall i \in I : M_\sigma(p) > M_\sigma(i), p \neq i \) (6.5)

The symmetry orientation of a point, \( \phi(p) \) is calculated from the most significant contribution of pixel pairs equidistant from the point. The orientation of a contribution, \( \varphi(i,j) \) is the average of the two contributing point’s orientations:

\[
\varphi(i,j) = \frac{\theta_i + \theta_j}{2} \tag{6.6}
\]

When performing pure rotational affine transformations, new co-ordinates may not fall exactly on pixel positions. That is, \( x', y' \) or both may not be integers. In such cases it is necessary to interpolate the intensities in the image.

Figures 6.1, 6.2, 6.3 depict the orientation normalisation of three digits: one, five and seven respectively. For each digit, three additional patterns are shown. These patterns were produced from the original by rotating by 90, 180 and 270 degrees.

From visual inspection of the three figures, the performance of this mechanism can be analysed. In each case, the mechanism produces two unique output orientations. Compared to the aim which was to produce just one unique output orientation, this method falls short. However, two orientations from four still has some use. Previously a classifier could not classify new patterns under differing rotations than used in the training set. By training a classifier with both possible normalisation cases, it could potentially classify the patterns under rotation - at least for right-angular bins as demonstrated.
Figure 6.1: Results of the orientation normalisation applied to a digit representing the number one at four different input orientations. Although the four outputs look similar, Figure 6.1h differs by 180 degrees.
Figure 6.2: Results of the orientation normalisation applied to a digit representing the number five at four different input orientations. There are almost two distinct output orientations but they are not identical.
Figure 6.3: Results of the orientation normalisation applied to a digit representing the number seven at four different input orientations. The transform has produced two distinct output orientations. Figures 6.3b and 6.3f share a common orientation, as do Figures 6.3d and 6.3h.
in the figure. This is not ideal. The size of the training set would double. For most networks, this would significantly increase the training time. In the case of the Probabilistic Neural Network, such a training set would require twice as many nodes in the pattern layer.

The authors of the paper proposing the GST [80], also proposed an alternative definition for calculating the symmetry orientation:

$$\varphi(i, j) = \arctan \frac{2 \cos(\theta_i - \alpha_{i,j}) \cos(\theta_j - \alpha_{i,j})}{\sin(\theta_i + \theta_j - 2\alpha_{i,j})}$$  \hspace{1cm} (6.7)

Where $\theta_i$ and $\theta_j$ are the orientations of points $i$ and $j$ respectively; and $\alpha_{i,j}$ is the angle between the horizontal axis and a line connecting points $i$ and $j$.

This definition addresses the issue that not all contributions meet the ideal case. That is, when $P(i, j) = 4$. The GST is a continuous measure of symmetry and this second definition calculates the orientation appropriately when $\theta_i + \theta_j \neq \pi$.

Hence, Figures 6.4, 6.5, 6.6 depict the orientation normalisation of the three digits, from four different input orientations each, using the second definition for calculating the symmetry orientation.

The figures show no common output orientation. The second definition has not produced a usable normalisation method. In order to create a normalisation method for patterns under all rotations (0 to 360 degrees), a new definition of symmetry orientation is required.
Figure 6.4: Results of the orientation normalisation (using the alternative GST definition) applied to a digit representing the number one at four different input orientations. Although there appears to be two distinct output orientations, closer visual inspection of the digit’s curve demonstrates that there are four different output orientations.
Figure 6.5: Results of the orientation normalisation applied (using the alternative GST definition) to a digit representing the number five at four different input orientations. There is no common output orientation in this case.
Figure 6.6: Results of the orientation normalisation (using the alternative GST definition) applied to a digit representing the number seven at four different input orientations. There are no common output orientations.
Figure 6.7: Visual depiction of calculating the average of two angles using the continuous variate arithmetic mean. Two different input angle combinations are shown to share the same average. This behaviour is not desired.

6.6 Modification of the GST

A normalisation method based on the definition of symmetry orientation in Equation 6.6 performs better than that using Equation 6.7. Equation 6.6 calculates a simple arithmetic mean average of the orientations of two contributing points. This is a suitable equation for calculating the average of two continuous variables. However, the variables in this case represent angles. Angles are circular, ‘wrapping-around’ at $360 \, / \, 0$ degrees or $2\pi \, / \, 0$ radians. Figure 6.7 demonstrates the problem associated with using Equation 6.6 for the normalisation of orientation.

In Figure 6.7, two cases are shown each with different contributing point orientations but sharing an average orientation. Such specific angles in the first case could be: $\theta_i = 20$, $\theta_j = 340$ degrees. And the second case: $\theta_i = 160$, $\theta_j = 200$ degrees. The average of both cases is 180 degrees. However, it is
intuitively known that the average of the first case is 0 degrees.

A more appropriate method for calculating the average of two angles is:

$$\varphi'(i, j) = \arctan \frac{\sin \theta_i + \sin \theta_j}{\cos \theta_i + \cos \theta_j}$$  \hspace{1cm} (6.8)

Where $\theta_i$ and $\theta_j$ are the orientations of two points, $i$ and $j$, respectively. Figure 6.8 depicts average of the angles calculated by this equation for the same two cases previously described.

Figures 6.9, 6.10, 6.11 depict the orientation normalisation of the three digits, from four different input orientations each, using the symmetry orientation definition provided by Equation 6.8.

One shared output orientation is depicted for the character in each of the Figures 6.9, 6.10, 6.11. For the digit five in Figure 6.10, visual inspection reveals there is a slight variance across the output, but they are consider-
Figure 6.9: Results of the orientation normalisation (using circular variate mean definition) applied to a digit representing the number one at four different input orientations. There is a single, common output orientation.
Figure 6.10: Results of the orientation normalisation (using circular variate mean definition) applied to a digit representing the number five at four different input orientations. There is a single, common output orientation.
Figure 6.11: Results of the orientation normalisation (using circular variate mean definition) applied to a digit representing the number seven at four different input orientations. There is a single, common output orientation.
ably more similar to each other than using the previous definitions of the symmetry orientation.

The visual inspection of these digits is useful as an indicator of the potential performance of such an orientation normalisation mechanism. However, it by no means proves its suitability or performance. And it would not be practical to visually inspect a large data set of digits. Hence it is necessary to conduct an experiment to measure the performance of a classifier using the proposed mechanism.

6.7 Experiment

The experiment tests the hypothesis that the definition of symmetry orientation in Equation 6.8 can be used to normalise the orientation of patterns prior to their classification.

Figure 6.12 depicts the proposed pattern recognition system process. A symmetry transform produces symmetry features. These are used to transform the orientation of the features in the original pattern, which are then presented to the network for classification.

![Figure 6.12: Overview of the proposed pattern recognition system structure to achieve position invariance using symmetry, as used in the experiment.](image)
6.7.1 Patterns

The input data is based on the United States Postal Service digit data set as used in the previous chapters. For this experiment, a border of eight pixels is added to the patterns to act as padding. This is to allow the rotation of the contents of the pattern without risking loss of information at the image boundaries. This is depicted in Figure 6.13. As demonstrated in Chapter 5, adding such a border does not affect the performance of the classifier.

A total of 6562 patterns are used for training. These represent digits zero to eight. These patterns were obtained by removing those patterns belonging to the class representing the digit nine from the USPS training data. The normalisation mechanism is designed to be invariant to complete rotation. The digits six and nine are not rotationally separable. That is, the digit six becomes the digit nine when it is rotated by 180 degrees (and vice versa). This is demonstrated in Figure 6.14.
6.7.2 Symmetry Transform

The Generalised Symmetry Transform is applied to each training pattern. Equation 6.6 is replaced with Equation 6.8. The point with the greatest symmetry magnitude is located using a traditional search algorithm.

6.7.3 Orientation Normalisation

The affine transformation parameter is the orientation of the point with the greatest symmetry magnitude, as per §6.5. A bi-cubic interpolation [38] is used to estimate the pixel intensities in the output pattern. Once transformed, the symmetry features are no longer required. This information can be discarded.

6.7.4 Classifier

The normalised patterns are used to train a Probabilistic Neural Network in the same manner as the previous chapters. Without knowing the prior
probabilities and associated expected losses, the modifiable weights in the last layer of the network are set to -1. The PNN smoothing operator, $\sigma_{\text{PNN}} = 0.7$ based on the experiment in Chapter 4. Both the control and the test networks sizes are $1024 \times 6562 \times 18 \times 9$ nodes. The increased size in the input layer reflects the increase dimensionality of the patterns. The decreased size in the pattern layer reflects the reduced size of the training set from removing the class of patterns representing the digit nine. Similarly, there is one less output node, and two less summation nodes.

### 6.7.5 Performance Measurement

In order to test the network’s classification performance of patterns under rotation, the original patterns must be rotated to a new orientation, normalised and presented for classification. Hence, three additional sets are used: the original patterns rotated at 45, 90, 180 and 270 degrees. These right-angular bins have the benefit that pre-normalisation rotation is guaranteed to align to pixel positions. That is, $x'$ and $y'$ in Equation 6.1 will be integers.

In order to test the network’s generalisation performance, an additional 1807 patterns are presented to the network. They are presented at their original orientation and also at 45, 90, 180 and 270 degree rotations.

### 6.8 Results

The results of the experiment are tabulated in Table 6.1.
### Table 6.1: The results of the orientation-invariant pattern classification experiment for right angular input (prior to normalisation) orientations.

<table>
<thead>
<tr>
<th>Input Orientation</th>
<th>Performance</th>
<th>Control</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Training Set</td>
<td>Testing Set</td>
</tr>
<tr>
<td>Original</td>
<td>100.0%</td>
<td>82.5%</td>
</tr>
<tr>
<td>45 Degrees</td>
<td>47.8%</td>
<td>48.5%</td>
</tr>
<tr>
<td>90 Degrees</td>
<td>94.1%</td>
<td>81.0%</td>
</tr>
<tr>
<td>180 Degrees</td>
<td>99.9%</td>
<td>82.5%</td>
</tr>
<tr>
<td>270 Degrees</td>
<td>94.2%</td>
<td>81.1%</td>
</tr>
</tbody>
</table>

#### 6.9 Discussion

The first observation is the performance drop having trained and tested on normalised patterns, compared to that of the control classifier. The performance drops by 12.5 percentage points to 82.5%. Hence the modification to the variation between patterns and classes after orientation normalisation has not assisted the classifier.

The performance of the normalisation mechanism is determined by comparing the recognition rate of the training and test sets at 90, 180 and 270 degrees to the original features. The training set performances vary by up to 5.9 percentage points; the test sets by up to 1.5 percentage points. The relatively small performance range suggests the orientation normalisation mechanism performs reasonably, though there is room for improvement.

The classifier performed significantly worse when processing patterns rotated by 45 degrees. When rotated by angles that are not right-angular, the pixel values have to be interpolated. This particularly affects the pixel intensity values of gradients or edges of uniform regions. The measurement of symmetry is dependent on the gradient information from the pattern. Rotat-
ing the pattern effects all gradients or edges in a way that does not maintain the symmetry information. This suggests a gradient or edge based symmetry transform is not suitable for providing symmetry information for this task.

At 180 degrees, the control classifier’s performance increases compared to using patterns transformed by 90 or 270 degrees. This is the result of patterns which are naturally invariant to rotation about 180 degrees. Digits ‘0’, ‘1’ and ‘8’ are mostly naturally invariant to rotation about 180 degrees.

Generally, the orientation normalisation mechanism demonstrates the ability to normalise inputs transformed in right-angular bins but at the expense of peak classification performance. The symmetry transforms reliance on edges and gradients makes it unsuitable for non-right-angular bins.

This investigation has provided the opportunity to reduce the number of features used for classification. This should be the most obvious benefit of the presence of reflectional symmetry in a pattern. Consider an object which is invariant to reflection about a given plane. To classify such an object, only half of its structure should need to be examined. The remaining half can be inferred as a reflection of the first. As highlighted in Chapter 2, Locher and Nodine [56] provided evidence showing that humans actually utilise this. They tracked the movements of the eye as it scanned the shape.

The position normalisation, described previously, identifies a point on the best fitting reflection plane. Associated with this point is the orientation of the reflection plane. Using these two pieces of information, the pattern can be divided in to two halves. Each half lies on one side of the reflection plane, or the other. If the object is perfectly symmetrical one half of the data can be discarded or ignored.
But what if the pattern is not quite perfectly symmetrical? Which half of the data should be discarded? What if the pattern recognition problem consists of a mixture of symmetrical and asymmetrical patterns? Is it still useful to discard data based on the best fitting reflection plane even if it is weakly supported? A more realistic approach would involve identifying a threshold measure to determine what is sufficiently symmetrical to discard data.

6.10 Contribution

This investigation has proposed and tested a symmetry feature based orientation normalisation mechanism. Though its performance is not perfect, it indicates the usefulness of symmetry features for this purpose. The features identified and used in this Chapter if calculated from a transform not based on edge gradients, could provide the performance desired.

The investigation detailed in this chapter identified an issue with the Generalised Symmetry Transform that has not been documented in related literature: the calculation of the orientation of the symmetry contribution does not take into account the circular nature of the orientation variables.
Chapter 7

Asymmetry

7.1 Introduction

Where Chapters 4, 5 and 6 investigated the importance and use of symmetry features in pattern recognition, this Chapter investigates its complement, asymmetry. Two original modifications of the phase weight function of the Generalised Symmetry Transform are proposed. The resulting transforms are named the Generalised Asymmetry and Symmetry Transform (GAST) and the Generalised Asymmetry Transform (GAT).

Symmetry and asymmetry feature classification tests are performed to determine their significance. The tests take the same form as those from Chapter 4.


7.2 Motivation

Chapters 2 and 4 detailed the absence of research into the usefulness of symmetry features for pattern recognition. This despite the obvious presence of symmetry in the environment and its wide spread use in man-made objects. Hence, Chapters 4, 5 and 6 investigated two distinct methods of utilising symmetry features for pattern recognition. The use of symmetry features was divided into two distinct methods. The first produced symmetry features which were then classified by a neural network. And the second used the symmetry features of a pattern to spatially transform the pattern prior to its classification. In the case of the former, whilst the performance did not improve upon the existing method, it did demonstrate that patterns could be classified by their symmetry features. In the latter method, it was demonstrated that position and rotation invariance could be achieved by normalising the position and orientation of the pattern in accordance with its symmetry features.

Asymmetry may be regarded as the complement of symmetry. That the classification of symmetry features was similar to but not equal to the performance of the existing method, suggests that some vital information of the pattern has been lost. Could this by the asymmetry information? With little other research into the classification of such features, it is important to investigate the significance of asymmetry in symmetry feature based pattern recognition.
7.3 Hypothesis

This first hypothesis is that patterns can be classified by their asymmetry features. That is, the asymmetry features alone are sufficient to describe the vast majority of the members of each class. Equally, the majority of patterns’ asymmetry features are sufficiently similar to those matching the one and only one class to which they belong.

The second hypothesis is that classification and generalisation performance improves using the full symmetry spectrum. The performance is with respect to that of just the symmetry features, as measured in Chapter 4. The term full symmetry spectrum is used to emphasise that the features of interest include the existence of both symmetry and asymmetry. This is in contrast to the research of this thesis so far (Chapters 4, 5 and 6), and the research of others (Chapter 3). Thus far, some component that is not symmetrical is asymmetrical. The important difference of this research is the premise that the symmetry spectrum consists of a range. At one extreme (say positive) are perfectly symmetrical objects. At the other extreme (negative) are perfectly asymmetrical objects. In the middle of the two, are objects which are neither symmetrical nor asymmetrical. Within this hypothetical spectrum, such objects are zero-valued. Previously, such objects were 50% symmetrical. Adopting a bi-polar approach to symmetry measurements is likely to affect a transform where symmetry at a given point is calculated from the sum of many smaller contributions.
7.4 Rationale

To describe the rationale of the hypotheses, an example based on the use of the Generalised Symmetry Transform will be considered.

The symmetry measure of a given point is derived from the contributions of point pairs at equal distances from it. Or to put it another way: pairs of points contribute to the symmetry measure of the point mid-way between them. A nearly perfectly symmetrical pair will contribute a high positive value. A nearly perfectly asymmetrical pair will contribute a very low positive value. But perfectly asymmetrical pairs contribute nothing (neither positive nor negative.) The number of contributing pairs depends only on the number of edge points at equal distance surrounding it.

This presents an issue. Any given measure of a symmetry at a point may be the result of one of two situations. The first, is a relatively low number of contributing pairs, each of which is highly symmetric. Alternatively, a relatively high number of contributing pairs may exist around the point, each of which is not very symmetric, but is not perfectly asymmetric either. The two situations may result in the same value of symmetry. Of course, it is also possible for a mixture of the two situations. But in the transform’s resulting output, it is not known which situation resulted in the measure. A premise naturally follows this discussion: a large number of nearly asymmetrical contributions does not equal a highly symmetric point.

The proposed solution is to introduce a bi-polar symmetry measure. In the new measure, a single contribution from an asymmetrical pair negates a contribution from a pair with an equal degree of symmetry. If the number
and degree of the symmetric contributions outnumbers and outweighs the asymmetric contributions, a point is considered to be symmetric. And vice versa. But now, a large number of nearly asymmetrical contributions equals a highly \textit{asymmetrical} point.

\section{7.5 Related Work}

Chapter 2 provided a literature review which included symmetry detection methods. The chapter contributed a new interpretation of two particular transforms: Huebner’s Qualitative Symmetry Transform (QST) \cite{34} and the Discrete Symmetry Transform (DST) \cite{20}. In both cases, the measure of symmetry can be defined in terms of the pattern’s measure of asymmetry. This took the form:

\begin{equation}
\text{Symmetry} = 1 - \text{Asymmetry} \quad (7.1)
\end{equation}

Where the asymmetry measure was defined uniquely per symmetry transform. Thus, the chapter provided two methods for calculating the asymmetry of an object. The first, Equation 2.7 derived from the QST:

\begin{equation}
A(p_i, m) = \frac{1}{C \cdot m} \sum_{j=1}^{m} \sigma(j, m) \cdot g(p_{i-j}, p_{i+j})^2 \quad (7.2)
\end{equation}

The second, Equation 2.28 from the DST:

\begin{equation}
A(i, j) = \sqrt{\frac{\sum_k (T_k(i, j))^2}{n} - \left( \frac{\sum_k T_k(i, j)}{n} \right)^2} \quad (7.3)
\end{equation}

The equations and transforms are described in much greater detail in
Chapter 2. The concept to note is that both equations calculate a distance from an ideal case of symmetry.

In Chapters 4, 5 and 6, the GST was chosen over the QST and DST with reason. It calculated symmetry in two dimensions more appropriately than combining two single dimension QST applications. It is a reflectional symmetry transform, unlike the DST which measures rotational symmetries. Chapter 4 justified the use of reflection symmetries over rotational symmetries: experiments indicated it is used by mammals (including humans) and aves.

The GST differs from the QST and the DST. Intensity values and the distance between points serve only to weight the measure of symmetry. The degree of symmetry is determined by the contributing point orientations and is calculated by the phase function [80]. Figure 7.1 shows the relationship between the phase function’s two inputs (the orientations of two contributing points) and the phase function’s output, given below:

\[ P(i, j) = (1 - \cos(\theta_i + \theta_j - 2\alpha_{ij}))(1 - \cos(\theta_i - \theta_j)) \]  

(7.4)

Where \( \theta_i \) and \( \theta_j \) are the orientations of points \( i \) and \( j \) respectively. The angle between a line connecting both points, and the horizontal axis - \( \alpha \) - is defined as:

\[ \alpha = \arctan \left( \frac{y_j - y_i}{x_j - x_i} \right) \]  

(7.5)

Where \( x \) and \( y \) are co-ordinates, as in \( i = (x_i, y_i) \) and \( j = (x_j, y_j) \).

Of importance is the definition of symmetry and asymmetry according to
Figure 7.1: The Generalised Symmetry Transform (GST) Phase Function, $P(i,j)$. The two horizontal axes represent the orientations of points $i$ and $j$ respectively. Asymmetrical combinations yield zero. Symmetrical combinations yield positive output.

the function. Four orientation pairs, $(\theta_i, \theta_j)$ are defined as perfectly symmetrical. They are listed in Table 7.1.

Perfect asymmetry is defined as:

$$\theta_i - \theta_j = 0$$ (7.6)

<table>
<thead>
<tr>
<th>$\theta_i$</th>
<th>$\theta_j$</th>
<th>$P(i,j)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>$\pi$</td>
<td>4</td>
</tr>
<tr>
<td>$\pi$</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>$\pi$</td>
<td>$2\pi$</td>
<td>4</td>
</tr>
<tr>
<td>$2\pi$</td>
<td>$\pi$</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 7.1: Combinations of point orientations defined as representing perfect symmetry by the Generalised Symmetry Transform’s Phase Function, $P(i,j)$. 
The Noise Tolerant Generalised Symmetry Transform (NTGST) Phase Function, $P(i, j)$. The two horizontal axes represent the orientations of points $i$ and $j$ respectively. Like the GST, asymmetrical combinations yield zero. Two different symmetry combination types are defined: negative and positive function values when the difference between orientations $i$ and $j$ approaches positive $\pi$ or negative $\pi$ respectively.

That is, there orientations are equal ($\theta_i = \theta_j$). The value of the phase function in this case is, $P(0, 0) = 0$.

As stated in §7.4, a more useful approach would be a bi-polar measure requiring a bi-polar phase function. In this case, $P(0, \pi) = +1$ and $P(0, 0) = -1$. The Noise Tolerant Generalised Symmetry Transform (NTGST) uses a bi-polar phase function: its output ranges from one to minus one [73]. Its relationship between its two inputs and output is depicted in Figure 7.2 and defined as:

$$P(i, j) = \sin\left(\frac{\theta_j + \theta_i}{2} - \alpha_{ij}\right) \times \sin\left(\frac{\theta_j - \theta_i}{2}\right) \quad (7.7)$$

161
However, asymmetry is still represented by zero, that is \( P(0, 0) = 0 \). Instead, the NTGST has distinguished between two cases of symmetry: divergent and convergent. These terms related to the whether the orientations are away or towards each other, respectively.

A more useful bi-polar phase function would range from one (for perfectly symmetrical pair combination) to minus one (for perfectly asymmetrical pair combinations). Such a phase function is proposed in the next section.

### 7.6 The Generalised Asymmetry and Symmetry Transform (GAST)

An alternative definition of the phase function is proposed:

\[
P(i, j) = \frac{(1 - \cos(\theta_i + \theta_j - 2\alpha_{ij}))(1 - \cos(\theta_i - \theta_j))}{2} - 1 \tag{7.8}
\]

It is depicted in Figure 7.3. It uses the original phase function as a base, but rescales the output to the range 1 to \(-1\). Thus, for asymmetrical contributions, \( P(i, j) = -1 \) and for symmetrical contributions \( P(i, j) = +1 \). For contributions which are neither symmetrical nor asymmetrical, \( P(i, j) = 0 \).

The measurement of the orientation requires a subtle modification. The current method measures the orientation of the contribution from two points. The most significant contribution (in terms of its magnitude) is stored as the orientation of the symmetry at that given point. If only this method is used, such an orientation is useless in certain situations. When \( M(p) < 0 \), point
Figure 7.3: The Generalised Asymmetry and Symmetry Transform (GAST) Phase Function, $P(i, j)$. The two horizontal axes represent the orientations of points $i$ and $j$ respectively. Asymmetrical pair combinations yield negative output. Symmetrical pair combinations yield positive output.

$p$ is asymmetric. Its orientation $\phi(p)$ still represents the orientation of the most significant symmetry contribution - not the most significant asymmetry contribution. Hence, it is proposed that the symmetry of a point $p$, $S_\sigma(p)$ is:

$$S_\sigma(p) = [M_\sigma(p), \phi_a(p), \phi_s(p)]$$  \hspace{1cm} (7.9)

Where $\phi_a$ and $\phi_s$ are the orientations of the most significant asymmetry and symmetry contributions. A post processing step could create a unified (a)symmetry orientation, $\phi$ by selecting the appropriate orientation, that is:

$$\phi(p) = \begin{cases} 
\phi_a(p) & \text{if } M_\sigma(p) < 0, \\
\phi_s(p) & \text{if } M_\sigma(p) \geq 0.
\end{cases}$$  \hspace{1cm} (7.10)
The transform that uses this new definition of the phase weight function and symmetry orientation is termed the Generalised Asymmetry and Symmetry Transform (GAST). This is to reflect its ability to produce a single map of symmetric and asymmetric regions.

### 7.7 The Generalised Asymmetry Transform (GAT)

The proposed GAST transform meets the requirements for a bi-polar phase function for separating symmetry and asymmetry. And thus, both can be measured and represented at the same time. In this section, an additional modification to the phase function is proposed. It is the complement of the GST’s phase function. When used to calculate the contribution of two points, it will suppress symmetric contributions in favour of asymmetric contributions. The purpose of this transform is to test the first hypothesis. By classifying the features produced by this transform, the resulting recognition rate will indicate how useful asymmetry is for pattern recognition. The transform using this phase function is named the Generalised Asymmetry Transform (GAT). The proposed phase function is:

\[
P(i,j) = -1 \times \frac{(1 - \cos(\theta_i + \theta_j - 2\alpha_{ij}))(1 - \cos(\theta_i - \theta_j))}{4}
\]  

(7.11)

The function’s range is one to zero. This is in contrast to the GST’s range of four to zero. Hence it is not exactly the complement of the GST phase
Figure 7.4: The Generalised Asymmetry Transform (GAT) Phase Function, $P(i,j)$. The two horizontal axes represent the orientations of points $i$ and $j$ respectively. Asymmetrical pair combinations yield positive output. Symmetrical pair combinations yield zero.

function. However, there is no obvious reason for the upper limit of four. The range one to zero can be easily interpreted as a fraction or percentage.

The method for calculating the orientation of asymmetry at a given point is the same as that for the GST. Since the asymmetry contributions are positive, no modifications are required.

Figure 7.4 shows the function’s output for its two orientation inputs.

With two additional transforms defined, it is now possible to test this chapter’s hypotheses.
7.8 Experiment

The tests measure network’s ability to classify asymmetry features compared to the original features and symmetry features from Chapter 4. Figure 7.5 depicts the proposed pattern recognition system process. A symmetry or asymmetry transform will produce symmetry features from an input image. These features are normalised prior to being presented to a neural network for classification.

![Diagram of proposed pattern recognition system structure](image)

Figure 7.5: Overview of the proposed pattern recognition system structure used in the experiment.

7.8.1 Feature Extraction

In order to test the hypothesis that patterns can be classified by their asymmetry features, the GAT transform (described above) will be used to extract asymmetry features. These features will then be presented to a classifier. Features produced by the GAST transform are also classified. This tests the hypothesis that features from the full symmetry spectrum will improve performance.

7.8.2 Classifier

As in the previous chapters, Probabilistic Neural Networks perform the role of the classifier. Its single parameter, the smoothing parameter $\sigma_{PNN}$, is
varied in the range 0.1 to 2.0 in 0.1 intervals. The single prescribed weights
of each node in the output layer is set to −1 to act as a pure inverter.

The feature classification control network uses the original patterns. All
other networks in the feature classification tests use symmetry features de-
rived from the original patterns. The original patterns’ dimensionality is
smaller than that of the symmetry features. As described in Chapter 4, this
is due to the two-dimensional convolution step of GST-based transforms.

Hence, the feature classification network size is $324 \times 7291 \times 20 \times 10$,
extcept for the control which is $256 \times 7291 \times 20 \times 10$.

The performance of the classifier is measured after training by presenting
each pattern in turn. The recognition rate, by pattern set, is calculated using
Equation 4.1 from §4.8.2. The recognition rate is reported for the training
and the test set, at each value of the smoothing parameter used.

### 7.8.3 Patterns

As in the previous chapters, the United States Postal Service (USPS) data
set is used. All 7291 training and 2007 test patterns are used. The $16 \times 16$
pixel pattern is processed by either the GST, GAT or GAST to produce a
$18 \times 18 \times 2$ point feature set. Chapter 4 shows that the symmetry magnitude
features are more likely to yield the best performance, whilst minimising
pattern dimensionality. Discarding the orientation information yields a $18 \times
18$ pattern, that is 324 inputs for the network.
7.8.4 Intensity Normalisation

The output of the symmetry transforms is not normalised. In this investigation, these features and separately normalised features will be used. For the sake of comparison, the former will be referred to as non-normalised. The algorithm used to normalise the inputs is listed in Algorithm 4.1. It normalises the training and test sets using the training maximum and minimum for the limits.

7.9 Results

The results are listed in Tables 7.2 and 7.3 for non-normalised and normalised network inputs respectively. Figures 7.6 and 7.7 are graphical representations of the two respective tables.

The peak performances using the training set were: GST, 100.0%; GAT 100.0%; and GAST 100.0%. This was achieved using non-normalised network inputs. The peak performances using the test set were: GST, 92.4%; GAT 92.2%; and GAST 93.2%.

7.10 Discussion

As in Chapter 4, the difference in performance between the normalised and non-normalised features is to require differing settings of the PNN smoothing parameter in order to achieve the peak recognition rate.

It was anticipated that a classifier using the GAT features would perform similarly to a classifier using the GST features. The results confirm this. The
Table 7.2: The results of the feature classification tests using non-normalised input. The classification and generalisation performance provided by the recognition rates of the training set and test set respectively. Comparison is made between the three GST based symmetry transforms. Performance is measured for a range of Probabilistic Neural Network smoothing parameters, $\sigma$. 

| PNN Smoothing Parameter, $\sigma$ | Recognition Rate | | | | | | |
|---|---|---|---|---|---|---|
|   | GST | GAT | GAST | GST | GAT | GAST |
| 0.1 | 100.0 | 100.0 | 100.0 | 25.9 | 35.3 | 29.8 |
| 0.2 | 100.0 | 100.0 | 100.0 | 29.8 | 75.0 | 46.8 |
| 0.3 | 100.0 | 100.0 | 100.0 | 35.1 | 92.0 | 73.3 |
| 0.4 | 100.0 | 100.0 | 100.0 | 45.0 | 92.0 | 90.0 |
| 0.5 | 100.0 | 100.0 | 100.0 | 58.6 | 92.0 | 92.9 |
| 0.6 | 100.0 | 100.0 | 100.0 | 71.3 | 92.0 | 93.0 |
| 0.7 | 100.0 | 100.0 | 100.0 | 82.7 | 92.1 | 93.0 |
| 0.8 | 100.0 | 100.0 | 100.0 | 88.7 | 92.1 | 93.0 |
| 0.9 | 100.0 | 100.0 | 100.0 | 91.4 | 92.1 | 93.0 |
| 1.0 | 100.0 | 100.0 | 100.0 | 92.2 | 92.1 | 93.0 |
| 1.1 | 100.0 | 100.0 | 100.0 | 92.4 | 92.2 | 93.0 |
| 1.2 | 100.0 | 100.0 | 100.0 | 92.4 | 92.1 | 93.0 |
| 1.3 | 100.0 | 100.0 | 100.0 | 92.4 | 92.2 | 93.0 |
| 1.4 | 100.0 | 100.0 | 100.0 | 92.4 | 92.2 | 93.1 |
| 1.5 | 100.0 | 100.0 | 100.0 | 92.4 | 92.2 | 93.1 |
| 1.6 | 100.0 | 100.0 | 100.0 | 92.4 | 92.2 | 93.0 |
| 1.7 | 100.0 | 100.0 | 100.0 | 92.4 | 92.0 | 93.1 |
| 1.8 | 100.0 | 100.0 | 100.0 | 92.4 | 92.1 | 93.1 |
| 1.9 | 100.0 | 100.0 | 100.0 | 92.3 | 91.9 | 93.1 |
| 2.0 | 100.0 | 100.0 | 100.0 | 92.3 | 91.8 | 93.2 |
Table 7.3: The results of the feature classification tests using intensity normalised input. The classification and generalisation performance provided by the recognition rates of the training set and test set respectively. Comparison is made between the three GST based symmetry transforms. Performance is measured for a range of Probabilistic Neural Network smoothing parameters, $\sigma$. 

<table>
<thead>
<tr>
<th>PNN Smoothing Parameter, $\sigma$</th>
<th>Recognition Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Training Set</td>
</tr>
<tr>
<td></td>
<td>GST</td>
</tr>
<tr>
<td>0.1</td>
<td>100.0</td>
</tr>
<tr>
<td>0.2</td>
<td>100.0</td>
</tr>
<tr>
<td>0.3</td>
<td>99.9</td>
</tr>
<tr>
<td>0.4</td>
<td>99.0</td>
</tr>
<tr>
<td>0.5</td>
<td>96.8</td>
</tr>
<tr>
<td>0.6</td>
<td>93.6</td>
</tr>
<tr>
<td>0.7</td>
<td>88.8</td>
</tr>
<tr>
<td>0.8</td>
<td>84.1</td>
</tr>
<tr>
<td>0.9</td>
<td>80.4</td>
</tr>
<tr>
<td>1.0</td>
<td>77.3</td>
</tr>
<tr>
<td>1.1</td>
<td>74.7</td>
</tr>
<tr>
<td>1.2</td>
<td>72.2</td>
</tr>
<tr>
<td>1.3</td>
<td>69.7</td>
</tr>
<tr>
<td>1.4</td>
<td>67.4</td>
</tr>
<tr>
<td>1.5</td>
<td>65.4</td>
</tr>
<tr>
<td>1.6</td>
<td>63.0</td>
</tr>
<tr>
<td>1.7</td>
<td>60.4</td>
</tr>
<tr>
<td>1.8</td>
<td>57.2</td>
</tr>
<tr>
<td>1.9</td>
<td>54.3</td>
</tr>
<tr>
<td>2.0</td>
<td>51.1</td>
</tr>
</tbody>
</table>
Figure 7.6: The results of the feature test. The network input values were non-normalised. The recognition rate is provided per symmetry transform (GST, GAT and GAST) and by data set (training or test set).
Figure 7.7: The results of the feature test. The network input values were normalised. The recognition rate is provided per symmetry transform (GST, GAT and GAST) and by data set (training or test set).
GAST outperforms (albeit slightly) both. These relative results suggest that the extreme cases are significant. That is, perfect asymmetry and perfect symmetry contributions are important in defining the overall pattern.

This chapter did not test the GAST or the GAT’s impact on a positional or rotational invariant mechanism, as described in Chapters 5 and 6 respectively. If this were tested in the future, these results suggest no significant improvement in performance will be gained.

None of the feature sets used resulted in a classifier that improved upon the control network of Chapter 4. This suggests that either symmetry and asymmetry features can not improve upon the traditional features or that the Generalised Symmetry Transform and its derivatives are unsuitable for the task.

7.11 Contribution

In this chapter, two original modifications of the phase function of the Generalised Symmetry Transform were proposed. These are termed the Generalised Asymmetry Transform (GAT) and the Generalised Asymmetry and Symmetry Transform (GAST). The naming reflects the significance each transform gives to particular features.

The feature classification test demonstrated that patterns can be classified by their asymmetry features alone. The experiment used the proposed GAT transform, in which symmetric regions were suppressed. When using features produced by the GAST transform, the classification and generalisation performance was improved (in comparison to the GST or GAT). This in
turn demonstrates that the asymmetry features are important for recognition of the patterns.
Chapter 8

Proposing a new Symmetry Transform

8.1 Introduction

This chapter investigates the possibility that the Generalised Symmetry Transform is not capable of detecting all reflectional symmetry planes. The role of the component functions are examined which leads to new definitions and a new symmetry transform, the Reflectional Symmetry Transform (RST). A comparison is made between the outputs of the GST and RST for ten basic shapes. The transform is applied to pattern recognition.

8.2 Motivation

Chapters 4, 5 and 6 investigated the use of symmetry in pattern recognition. The proposed methodology was further extended in Chapter 7 to investigate
Figure 8.1: The input and resulting output symmetry magnitude map after processing by the GST. Two reflectional symmetry planes are present, but there should be four.

the role of asymmetry. The performances of the pattern recognition systems were adequate. But the proposed systems were not an improvement on existing methods. One explanation for this is that information may be lost when the symmetry transform is performed.

Alternatively, the symmetry transform used might not be detecting symmetry adequately. The transform primarily used in this thesis is the Generalised Symmetry Transform (GST). Figure 8.1b depicts the output of the GST for a perfect square, Figure 8.1a.

Part (b) of the figure depicts the symmetry magnitude map, $M_\sigma$. This map clearly depicts two reflection planes: one horizontal and one vertical. A perfect square actually supports four reflection planes. In addition to the two shown in the figure, there are two diagonal planes. This simple example provides the motivation to re-examine the GST in greater detail.
8.3 Hypothesis

The first hypothesis is that the Generalised Symmetry Transform is not a true symmetry operator: it is an opposing edge mid-point detector. That is, the GST identifies regions lying between two points whose orientation differs by approximately $\pi$ radians. The difference is an approximation due to the continuous nature of the phase function.

The second hypothesis is that a continuous measure of reflectional symmetry in images is a function of three independent measures based on the comparison of pixel pairs. The first measure is the degree of variation of the two colour intensity values. The second measure is the degree of variation of the distance between the reflection plane and each point. The final measure is the degree of variation of the internal angles of the points. The internal angle is the angle between a pixel’s orientation and the other pixel.

8.4 Related Work

Heidemann [28] extended the GST to detect colour symmetries in order to detect feature points of interest that would be lost if represented in grey scale. As part of his commentary on the original GST, he proposed the following modification to the Phase Weight Function, the grey scale equivalent of which would be:

$$P(i,j) = \cos^2 (\theta_i + \theta_j) \times \cos^2 (\theta_i) \times \cos^2 (\theta_j) \quad (8.1)$$

where $\theta_i$ and $\theta_j$ are the orientations at points $i$ and $j$ respectively; and
\[ \cos^2(x) = \frac{1}{2} (1 + \cos(2x)) \]  

(8.2)

The objective of the modification was to abandon favouring only ‘opposing edges’ by changing the periodicity from \(2\pi\) to \(\pi\). However, this modification does not produce a classifier that identifies the remaining two reflection planes of the perfect square from the problem defined in §8.2.

In identifying a means by which the computational overhead could be reduced, he proposed the removal of the distance weight function. The grey scale equivalent contribution function becomes:

\[ C(i, j) = P(i, j) \times G(i, j) \]  

(8.3)

where Heidemann defined a gradient weight function, \(G(i, j)\) as:

\[ G(i, j) = (1 + \log \| \nabla p_i \|) \times (1 + \log \| \nabla p_j \|) \]  

(8.4)

Though these modifications do not address the issues that motivate this investigation, they serve as an insight into potential areas for improvement.

Kovesi [44] identified that the GST is not contrast invariant: that a bright circle is considered more ‘symmetric’ that a low contrast one. Kovesi proposed a contrast invariant symmetry and asymmetry transform using localised frequency information derived from the Wavelet Transform. In the analysis of frequencies, an axis of symmetry corresponds to the components at the maximum and minimum amplitudes; asymmetry at the inflexion points. Kovesi’s method only identifies bilateral symmetry and only using intensity
values in one dimension. It is possible to extend this to two dimensions by applying at multiple orientations and aggregating the results. The modifications to the GST in this chapter are made such that the resulting transform is contrast invariant.

8.5 Rationale

The definition of reflectional symmetry provided in §2.2 is more functional than prescriptive. Though an intuitive concept, a more specific definition is more useful for basing a new transform, or more robustly assessing an old one (e.g. the GST). The GST is based on contributions from pairs of points. Hence a formal definition in terms of point pair contributions is required.

Two points are symmetrical if three conditions hold. These conditions can be tested using a physical reflection plane: a mirror. The first condition is that the colours of the two points are identical. The second is that the distance between each point and the reflection planes are identical. The remaining condition is based on the orientations of the points. However, it is easier to describe the relationship in terms of the interior angles. The interior angle is that between the point’s orientation and a line connecting the two points. The orientations are symmetric if the interior angles are of equal magnitude but opposite direction. There is no evidence in nature to suggest that any one of the three conditions is more important than the other two. Hence each should be weighted equally. In addition, no matter how perfectly symmetric two conditions are, if the third deviates completely from the symmetric ideal, then the two points are not symmetric.
A new symmetry transform is now proposed based on this definition. Chapter 2 identified the motivation behind using a continuous measure over a discrete one. Hence the new symmetry transform can tolerate deviations from the ideal case. The tolerance can be adjusted for each of the three conditions (or functions) of symmetry.

It is proposed that the new symmetry transform provides the same information as the GST. Hence, the output of the transform is a symmetry map, $S$, for each point $p$ in the input image, consisting of the magnitude, $M(p)$, and the orientation, $\phi(p)$, of symmetry at that point:

$$S(p) = [M(p), \phi(p)]$$ (8.5)

By providing the same information and structure, the new transform can be used in place of the GST wherever it has previously been used.

The proposed approach for calculating the symmetry information is to follow the algorithmics of the GST. That is, symmetry is calculated by comparing every combination of edge point pairs. Each pair provides a contribution to a point in the image located mid-way between them. However, substantial modification of the function definitions is proposed. The new transform calculates the reflectional symmetry in the image. Hence, this new transform is termed the Reflectional Symmetry Transform (RST).

The definitions of the symmetry magnitude and orientation will now be discussed.
8.5.1 Symmetry Magnitude

Figure 8.2 depicts the three components that determine the degree of symmetry of two points. These are: the internal angles of each point to its pair; the intensity values; and each pair’s distance to the reflection plane. Each component is considered as significant as the others. There is no evidence or existing argument to suggest anything to the contrary. Hence, two points, $i$ and $j$, symmetry magnitude contribution, $C(i,j)$ is defined as:

$$C(i,j) = P_{\sigma_{P}}(i,j) \times I_{\sigma_{I}}(i,j) \times D_{\sigma_{D}}(i,j)$$  \hspace{1cm} (8.6)

where $P_{\sigma_{P}}(i,j)$ is the symmetric phase weight function; $I_{\sigma_{I}}(i,j)$ is the symmetric intensity weight function; and $D_{\sigma_{D}}(i,j)$ is the symmetric distance weight function. Each symmetric component function is weighted equally and yields the same range of output values: zero to one inclusive. Thus, if the intensities and orientations are symmetric, but the distance is not, the contribution will be insignificant or nothing. All three components must satisfy their symmetry criteria for the contribution to be symmetric. The components used to calculate the magnitude are depicted in Figure 8.2.

The contribution has the same range of values: zero to one inclusive. Zero signifies that the two points are not symmetric; One signifies that the two points are symmetric.

The symmetry magnitude of a point $p$ is the sum of all contributions from edge points whose mid-point is the point $p$: 

181
Figure 8.2: The three relevant components associated with the two points used to determine the magnitude and direction of a potential reflection plane. These are colour intensity, \( I(p) \); distance to the reflection plane, \( d \); and internal angle, \( \psi \). Only points on non-uniform areas are considered. This example demonstrates two points on the edge of a shape.
\[ M(p) = \sum_{(i,j) \in \Gamma(p)} C(i,j) \] (8.7)

Where \( \Gamma(p) \) is the set of points whose mid-point is the point \( p \), as defined by the GST, given in Equation 2.12.

**Symmetric Phase Function**

The basic idea is to calculate the reflected orientation of a point, given its original orientation and the position of the reflected point relative to the first. This target orientation is compared to the actual reflected orientation. As the difference between the target and actual orientations increases, the value of symmetry assigned to the pairs decreases.

One alternative is to compare the interior angles of the two points. The interior angle is that between the point’s orientation and a line connecting the two points. If symmetrical, the magnitude of the two interior angles will be equal, but in opposite directions. A second alternative is to calculate the orientation of the target reflection plane and compare to the average orientation of the two points. The reflection plane should lie at a right angle to a line connecting the two points. If symmetrical, the average of the orientations will also lie at a right angle to the connecting line.

The circular-circular association function is employed as it is well documented [18, 47], appropriate function for the task. The expected orientation is calculated using

\[ \theta'_j = -(\theta_i + 2\alpha_{ij} (\text{mod} 2\pi)) + \pi \] (8.8)
Where $\theta_i$ is the orientation of point $i$; $\theta'_j$ is the expected (target) orientation of point $j$; and $\alpha_{ij}$ is the angle between a line connecting points $i$ and $j$, and the horizontal access.

The equality (or inequality) of the target and actual orientations of point $j$ are measured using a non-linear function:

$$P(i, j) = e^{-\frac{(\theta_j - \theta'_j)^2}{\sigma_P^2}}$$ \hspace{1cm} (8.9)

Where $\sigma_P$ is the symmetric phase function’s tolerance parameter. The parameter $\sigma_P$ determines the tolerance to deviations from the perfect symmetrical case. Smaller values (e.g. $\sigma_P = \frac{\pi}{32}$) are less tolerant than larger values (e.g. $\sigma_P = \frac{\pi}{4}$). A less tolerant function will behave much more like a discrete measure of symmetry than a more tolerant function.

Figure 8.3 demonstrates the difference between the GST Phase Weight Function and the RST Symmetric Phase Function.

**Minimum phase difference**

The phase function of the GST discriminates against symmetries detected within the edges of a shape. The authors decided that such symmetries were ‘uninteresting’ in the case of an attentional operator. With respect to this work, such symmetries are considered artificial. They are a by-product of the edge based measurement and are not true to the intuitive concept of symmetry. To remove these symmetries, only contributions from pairs of points whose phase difference is greater than a set minimum are permitted. A minimum difference greater than zero is used because of the discrete nature
Figure 8.3: A comparison of the GST’s phase weight function with the RST’s symmetric phase function. In both sub-figures, $\alpha_{ij} = 0$. In the case of the RST, the phase tolerance parameter, $\sigma_p = \frac{\pi}{8}$.
of digitised images and the requirement to approximate the pixel gradients.

**Symmetric Intensity Function**

The GST does not identify an intensity weight function. However, the contribution is defined by the product of four terms. The final two terms are \( r_i \) and \( r_j \). A similar function is \( In(i, j) = r_i \times r_j \) which Heidemann [28] called the gradient weight function. However, this function is not a measure of symmetry. The output of the function is irrespective of the symmetric nature of the two points.

The proposed symmetric intensity function calculates the measure of symmetry based on a comparison of two points’ grey scale intensity. In terms of grey scale intensity, two points are perfectly symmetrical when the two intensities are equal. Like the phase weight function, it is non-linear:

\[
In_{\sigma_I}(i, j) = e^{-\frac{|I(i) - I(j)|^2}{\sigma_I^2}}
\]  
(8.10)

Where \( I(i) \) and \( I(j) \) are the grey scale values of points \( i \) and \( j \) respectively. The parameter \( \sigma_I \) controls the tolerance to deviations from the ideal symmetric case. Figure 8.4 depicts the proposed function.

**Symmetric Distance Function**

The GST’s distance weight function serves to make contributions closer to the reflection plane more significant than those further away. Figure 8.5 depicts this. There is little justification for this. By the widely accepted formal definition of symmetry, objects are not considered less symmetrical because
Figure 8.4: The symmetric intensity function of the Reflectional Symmetry Transform. The function peaks when the two intensities are equal. As the two intensities differ, the function’s output drops at a rate controlled by $\sigma_I$. In this example, $\sigma_I = 0.1$.

they are far apart relative to other symmetrical objects. What matters is that the distances between the objects and the reflection plane are equal. Placing a mirror on the site of the reflection plane proves this.

For the RST, the symmetric distance function replaces the distance weight function. Unlike the GST, the distance between the two contributing points does not affect the measure. Instead, it yields its maximum output value when the distance between point $i$ and the reflection plane, and the distance between the reflection plane and point $j$ are equal. When the two distances are not equal, the two contributing points are not symmetric. It is a continuous function. That is, small differences in these two distances are relatively more symmetric that larger differences. In order to be consistent with the other symmetric functions, it too is a non-linear function:
Figure 8.5: Three pairs of points sharing a single reflection plane. In each case, the distance between point $i$ and the reflection plane; and the distance between the reflection plane and point $j$ are equal. In the case of the Generalised Symmetry Transform, the top pair are more symmetrical than the bottom pair because in the top case, the distance between points $i$ and $j$ is less than the that in the bottom case. This is not consistent with the functional definition of reflectional symmetry.
Figure 8.6: The Reflectional Symmetry Transform’s symmetric distance function. When the distance between point \(i\) and the reflection plane is equal to the distance between point \(j\) and the reflection plane, the output is maximum. As the two distances differ, the functions output drops at a rate controlled by \(\sigma_D\). In this example, \(\sigma_D = 1\)

\[
D_{\sigma_D}(i, j) = e^{-\frac{|d_i - d_j|^2}{\sigma_D^2}}
\]  
(8.11)

Where \(d_i\) is the distance between point \(i\) and the reflection plane and \(d_j\) is the distance between point \(j\) and the reflection plane.

Hence, the rate at which the symmetry measure decreases with respect to increasing differences in the two distances is controlled by \(\sigma_D\). This permits the strictness of the function to be controlled. Figure 8.6 depicts the function output for a range of distances \(d_i\) and \(d_j\).

Although this chapter has discredited the biasing of the contribution of
two points using the distance between them, there may be cases or applications where such behaviour is desired. The next function suppresses contributions from all point pairs with a distance between them greater than a given maximum:

\[
D_{l_{d_{\text{max}}}}(i, j) = \frac{1}{1 + e^{d_{\text{max}} - \|p_i + p_j\|}}
\]  

(8.12)

Where \(d_{\text{max}}\) is the maximum distance between the two contributing points.

A graph depicting this function is provided in Figure 8.7a.

Alternatively, it may be desirable to suppress contributions from all point pairs that have a distance less than a given minimum:

\[
D_{g_{d_{\text{min}}}}(i, j) = \frac{1}{1 + e^{\|p_i + p_j\| - d_{\text{min}}}}
\]  

(8.13)

Where \(d_{\text{min}}\) is the minimum distance. This is depicted in Figure 8.7b.

The key difference between both the maximum and the minimum distance functions, and the GST’s distance weight function is that they use sigmoid based functions rather than Gaussian. This is because it is considered more appropriate that all points are equal to others on their side of the threshold.

In practice, only those points that are equidistant from the reflection plane are considered. This is to reduce computation time. This is equivalent to using very small values of \(\sigma_D\). Where it is desired to implement a maximum or minimum distance discrimination function, the algorithmics of the GST (and thus the RST) can be modified to only consider pixels up to or beyond that distance.
Figure 8.7: Alternative functions to the GST’s distance weight function. These are only appropriate where suppression of point pairs’ contributions is desired based on the distance between each point.
8.5.2 Symmetry Orientation

The symmetry orientation definition of the GST is replaced with that from Chapter 6. The investigation of using symmetry to normalise pattern orientation lead to the observation that the GST’s definition is incorrect. This is due to the circular nature of angular measurements. Figures 6.7 and 6.8 visually depict the difference in definitions. Hence, the definition of symmetry orientation in the RST is:

\[ \varphi(i,j) = \arctan \frac{\sin \theta_i + \sin \theta_j}{\cos \theta_i + \cos \theta_j} \]  

(8.14)

8.6 Visual Verification

To assess the effectiveness of the RST, the transform will be tested using simple shapes and compared to the GST and the intuitively known expected reflection planes. The shapes are represented by a 122x122 pixel, grey scale image. The shape foreground is white, on a black background.

The GST’s single parameter, the distance weight function variance, \( \sigma \), is set as the nearest whole integer to \( \sqrt{122^2 + 122^2} \), \( \sigma = 173 \). The output is normalised to the continuous interval \([0, 1]\).

The RST has three parameters: the phase tolerance, \( \sigma_P = \frac{\pi}{64} \); intensity tolerance is \( \sigma_I = 50 \); and minimum phase difference of \( \frac{\pi}{8} \). The RST’s output is also normalised to the continuous interval \([0, 1]\).

The input, GST and RST output images are tabulated in Figures 8.8 and 8.9 for the following shapes: a square, a rectangle, an isosceles triangle, an equilateral triangle, a right-angled triangle, a circle, two orientations of
a cross, a hexagon and a six-sided star. In the output images, white pixels denote symmetrical regions; black denote the absence of symmetry.

Beginning with the square, the RST correctly identifies four reflectional symmetry planes. The GST identifies only two. In the case of the rectangle, the GST correctly identifies the two planes. The RST depicts planes connecting the corner points to the horizontal mid-plane. These are not globally supported reflection planes. However, they are correct with respect to the RST’s continuous, local nature.

The three triangle inputs yield significantly different transform outputs. The RST yields the actual reflection planes. In the case of the right-angled triangle, it also identifies the local symmetries connecting the corners to the mid-point of the opposing edges. The GST’s output varies starkly from the expected planes.

The circle supports an infinite number of reflection planes, all crossing at the centre. Normalised, the symmetry map is expected to consist of a single point or small cluster of points of high symmetrical value at the centre of the circle. Both the GST and the RST produce the expected pattern.

The two crosses demonstrate the GST’s nature to favour opposing edges whose angular difference is $\pi$ radians. The RST, on the other hand, detects the four reflection planes successfully. With respect to the hexagon, the GST detects three of six planes: the RST detects all six - albeit two of which appear very faint on the figure depicted. Similarly, the GST detects only three of the planes of the six-sided star.

The improvements yielded by the RST are clearly identifiable from the visual inspection. The remaining imperfections are the direct result of cal-
<table>
<thead>
<tr>
<th>Input</th>
<th>GST</th>
<th>RST</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Square" /></td>
<td><img src="image" alt="GST for Square" /></td>
<td><img src="image" alt="RST for Square" /></td>
</tr>
<tr>
<td><img src="image" alt="Rectangle" /></td>
<td><img src="image" alt="GST for Rectangle" /></td>
<td><img src="image" alt="RST for Rectangle" /></td>
</tr>
<tr>
<td><img src="image" alt="Isosceles Triangle" /></td>
<td><img src="image" alt="GST for Isosceles Triangle" /></td>
<td><img src="image" alt="RST for Isosceles Triangle" /></td>
</tr>
<tr>
<td><img src="image" alt="Equilateral Triangle" /></td>
<td><img src="image" alt="GST for Equilateral Triangle" /></td>
<td><img src="image" alt="RST for Equilateral Triangle" /></td>
</tr>
<tr>
<td><img src="image" alt="Right Angled Triangle" /></td>
<td><img src="image" alt="GST for Right Angled Triangle" /></td>
<td><img src="image" alt="RST for Right Angled Triangle" /></td>
</tr>
</tbody>
</table>

Figure 8.8: Comparison of the output of the GST and RST.

194
Figure 8.9: Comparison of the output of the GST and RST.
8.7 Experiment

Visual inspection of the RST on primitive shapes demonstrated that it detected reflectional symmetries better than the GST. This experiment tests a pattern recognition system using symmetry magnitude features produced by the RST.

This experiment will also test a pattern recognition system’s ability to tolerate noise when it is using symmetry features. Chapter 7 made reference to the Noise Tolerant Generalised Symmetry Transform (NTGST) of Park et al [73] with which a performance comparison will be made.

8.7.1 Hypothesis

The hypothesis is that a pattern recognition system performing handwritten offline digit recognition will perform generalisation better using symmetry features produced by the RST than the GST and its more similar derivatives, the GAST and NTGST. This is based on the visual inspection of its performance on the primitive shapes.

In addition, it is hypothesised that performing pattern recognition with symmetry features will better tolerate noise in the patterns than

8.7.2 Classifier

As in the previous experiments, classification will be performed by a Probabilistic Neural Network (PNN). The PNN smoothing parameter, $\sigma_{PNN}$, will
be varied for the previously untested symmetry transforms. Good performance at low values will indicate that the classifier is acting as a nearest-neighbour classifier. Good performance at high values will indicate that the classifier is approaching the Bayes-optimal. The symmetry feature classification network size is $324 \times 7291 \times 20 \times 10$, and the control network for the traditional features is $256 \times 7291 \times 20 \times 10$.

8.7.3 Patterns

As in the previous chapters, the United States Postal Service (USPS) data set is used. All 7291 training and 2007 test patterns are used. In the noise tolerance test, zero mean Gaussian noise is added to each of the patterns in the test set only. The test varies the Gaussian noise variance between 0.01 and 0.05.

8.7.4 Symmetry Transform

Four symmetry transforms will be tested and compared. They are: the Generalised Symmetry Transform [80]; the Generalised Asymmetry and Symmetry Transform (described in §7.6); the Noise Tolerant Generalised Symmetry Transform [73] (described in §7.5); and the Reflectional Symmetry Transform proposed in this Chapter. Based on the insight gained from the investigation in Chapter 4, only the symmetry magnitude features will be used.

In this experiment, the RST’s symmetric distance function is not used. Computation time is reduced by creating a set of point pairs and calculating their contribution to a point mid-way between them. The alternative is to
compare all possible point pairs around a point, where a line connecting the pair would cross the point they contribute a symmetry measure to. However, the symmetric distance function would make contributions negligible that are not equidistant.

In the case of the RST, the transform’s phase tolerance and intensity tolerance values will be varied, but always equal. Set together, they become the RST’s symmetry tolerance parameter. The effect of this parameter is described in §8.5.1. Neither the maximum nor the minimum distance functions are employed. Contributions do not need to meet a minimum phase difference to be included.

8.8 Results

Figure 8.10 depicts the data in Table 8.1 showing the performance of the classifier using the four symmetry transforms (one of which at two different tolerances). All five could classify the training data completely (100%) after training and did so using a small spread (PNN smoothing parameter, $\sigma = 0.1$).

Figure 8.11 depicts the data in Table 8.2 showing the performance of the classifier using the symmetry transforms when recognising the previously unseen testing data set. The peak performances, in order, are: the GAST (93.0%); the GST (92.5%); the NTGST (91.7%); the RST with a high tolerance (0.27) for deviations from the ideal symmetry case (91.6%); and then less tolerant (0.03) RST (71.3%).

The classification and generalisation experiment was also run for the RST
Figure 8.10: Comparison of pattern recognition on the training data set using the Generalised Symmetry Transform (GST); the Noise Tolerant GST (NTGST); the Generalised Asymmetry and Symmetry Transform (GAST); and two configurations of the Reflectional Symmetry Transform: low and high tolerances to deviations from the ideal symmetric case.

setting the tolerance parameters (equally) in the range 0.01 to 0.34. This result is depicted in Figure 8.12.

The results of the noise tolerance test are given in Table 8.3 and depicted in Figure 8.13.

8.9 Discussion

Each of the transforms produced patterns which were sufficiently different for the classifier to recognise each class. The results of the GST and GAST
Figure 8.11: Comparison of pattern recognition generalisation on the test data set using the Generalised Symmetry Transform (GST); the Noise Tolerant GST (NTGST); the Generalised Asymmetry and Symmetry Transform (GAST); and two configurations of the Reflectional Symmetry Transform: low and high tolerances to deviations from the ideal symmetric case.
Figure 8.12: Graph to show the effect of increasing the tolerance to deviations from the ideal symmetric case when performing pattern recognition using the Reflectional Symmetry Transform.
Figure 8.13: Comparison of symmetry transforms performing pattern recognition with increasing Gaussian noise. The control results are based on a Probabilistic Neural Network performing recognition using the traditional features (with noise).
Table 8.1: Data to show the ability of the classifier to recognise the training set using symmetry features for the Generalised Symmetry Transform (GST); the Noise Tolerant GST (NTGST); the Generalised Asymmetry and Symmetry Transform (GAST); and two configurations of the Reflectional Symmetry Transform: low and high tolerances to deviations from the ideal symmetric case.

<table>
<thead>
<tr>
<th>Spread</th>
<th>NTGST</th>
<th>GST</th>
<th>GAST</th>
<th>RST (Low)</th>
<th>RST (High)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>100.0</td>
<td>100.0</td>
<td>100.0</td>
<td>100.0</td>
<td>100.0</td>
</tr>
<tr>
<td>0.2</td>
<td>100.0</td>
<td>100.0</td>
<td>100.0</td>
<td>100.0</td>
<td>99.2</td>
</tr>
<tr>
<td>0.3</td>
<td>99.9</td>
<td>99.9</td>
<td>100.0</td>
<td>99.8</td>
<td>97.1</td>
</tr>
<tr>
<td>0.4</td>
<td>99.2</td>
<td>99.0</td>
<td>99.5</td>
<td>97.1</td>
<td>93.9</td>
</tr>
<tr>
<td>0.5</td>
<td>97.6</td>
<td>96.8</td>
<td>98.0</td>
<td>90.0</td>
<td>89.6</td>
</tr>
<tr>
<td>0.6</td>
<td>95.5</td>
<td>93.6</td>
<td>95.1</td>
<td>82.4</td>
<td>85.5</td>
</tr>
<tr>
<td>0.7</td>
<td>93.1</td>
<td>88.8</td>
<td>90.8</td>
<td>73.7</td>
<td>81.4</td>
</tr>
<tr>
<td>0.8</td>
<td>90.4</td>
<td>84.1</td>
<td>86.2</td>
<td>63.2</td>
<td>76.9</td>
</tr>
<tr>
<td>0.9</td>
<td>87.3</td>
<td>80.4</td>
<td>82.5</td>
<td>54.5</td>
<td>72.3</td>
</tr>
<tr>
<td>1.0</td>
<td>84.1</td>
<td>77.3</td>
<td>79.1</td>
<td>48.0</td>
<td>68.1</td>
</tr>
<tr>
<td>1.1</td>
<td>80.6</td>
<td>74.7</td>
<td>76.4</td>
<td>44.0</td>
<td>64.1</td>
</tr>
<tr>
<td>1.2</td>
<td>77.5</td>
<td>72.2</td>
<td>74.0</td>
<td>41.1</td>
<td>60.9</td>
</tr>
</tbody>
</table>
Table 8.2: Data to show the ability of the classifier to generalise by performing recognition on the test set using symmetry features for the Generalised Symmetry Transform (GST); the Noise Tolerant GST (NTGST); the Generalised Asymmetry and Symmetry Transform (GAST); and two configurations of the Reflectional Symmetry Transform: low and high tolerances to deviations from the ideal symmetric case.

<table>
<thead>
<tr>
<th>Spread</th>
<th>NTGST</th>
<th>GST</th>
<th>GAST</th>
<th>RST (Low)</th>
<th>RST (High)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>91.6</td>
<td>92.3</td>
<td>93.0</td>
<td>71.3</td>
<td>91.6</td>
</tr>
<tr>
<td>0.2</td>
<td>91.7</td>
<td>92.4</td>
<td>93.2</td>
<td>72.1</td>
<td>90.4</td>
</tr>
<tr>
<td>0.3</td>
<td>91.4</td>
<td>92.5</td>
<td>93.3</td>
<td>71.9</td>
<td>87.8</td>
</tr>
<tr>
<td>0.4</td>
<td>90.4</td>
<td>91.1</td>
<td>92.5</td>
<td>69.5</td>
<td>85.1</td>
</tr>
<tr>
<td>0.5</td>
<td>89.1</td>
<td>89.2</td>
<td>90.1</td>
<td>65.3</td>
<td>81.6</td>
</tr>
<tr>
<td>0.6</td>
<td>87.2</td>
<td>85.3</td>
<td>87.0</td>
<td>60.6</td>
<td>77.8</td>
</tr>
<tr>
<td>0.7</td>
<td>85.3</td>
<td>81.1</td>
<td>83.8</td>
<td>55.3</td>
<td>73.4</td>
</tr>
<tr>
<td>0.8</td>
<td>82.7</td>
<td>74.5</td>
<td>80.3</td>
<td>50.9</td>
<td>70.0</td>
</tr>
<tr>
<td>0.9</td>
<td>78.9</td>
<td>74.3</td>
<td>77.1</td>
<td>47.5</td>
<td>67.2</td>
</tr>
<tr>
<td>1.0</td>
<td>76.2</td>
<td>71.9</td>
<td>74.2</td>
<td>44.5</td>
<td>63.7</td>
</tr>
<tr>
<td>1.1</td>
<td>73.9</td>
<td>69.4</td>
<td>71.5</td>
<td>42.2</td>
<td>60.5</td>
</tr>
<tr>
<td>1.2</td>
<td>71.1</td>
<td>67.2</td>
<td>70.0</td>
<td>40.9</td>
<td>57.2</td>
</tr>
</tbody>
</table>

Table 8.3: Data to show the effect of noise on the performance of a Probabilistic Neural Network performing digit recognition using traditional features (the control) and symmetry features produced by the Generalised Symmetry Transform (GST); the Noise Tolerant GST (NTGST); the Generalised Asymmetry and Symmetry Transform (GAST); and two configurations of the Reflectional Symmetry Transform: low and high tolerances to deviations from the ideal symmetric case.

<table>
<thead>
<tr>
<th>Noise</th>
<th>NTGST</th>
<th>GST</th>
<th>GAST</th>
<th>RST (Low)</th>
<th>RST (High)</th>
<th>Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>91.7</td>
<td>92.5</td>
<td>93.3</td>
<td>71.3</td>
<td>91.6</td>
<td>95.2</td>
</tr>
<tr>
<td>0.01</td>
<td>62.4</td>
<td>61.1</td>
<td>66.0</td>
<td>19.0</td>
<td>27.3</td>
<td>28.1</td>
</tr>
<tr>
<td>0.02</td>
<td>57.6</td>
<td>55.9</td>
<td>60.7</td>
<td>18.2</td>
<td>25.9</td>
<td>28.2</td>
</tr>
<tr>
<td>0.03</td>
<td>54.4</td>
<td>53.2</td>
<td>56.3</td>
<td>16.1</td>
<td>24.6</td>
<td>27.6</td>
</tr>
<tr>
<td>0.04</td>
<td>54.2</td>
<td>51.2</td>
<td>53.5</td>
<td>14.8</td>
<td>23.5</td>
<td>27.2</td>
</tr>
<tr>
<td>0.05</td>
<td>52.9</td>
<td>47.6</td>
<td>48.7</td>
<td>15.1</td>
<td>22.4</td>
<td>27.1</td>
</tr>
</tbody>
</table>
were known from the investigations in Chapters 4 and 7 respectively. The performance of NTGST and the RST were not previously known. When classifying the previously unseen test data set, using RST produced symmetry features did not outperform a classifier using the traditional features. Nor did it improve upon any other the other symmetry transforms.

Figure 8.12 provides an insight into the usefulness of symmetry features for pattern recognition. The graph shows that using a strict definition of symmetry yields features which produce a poor performing classifier. Increasing the tolerance to deviations from the ideal symmetric case increases the performance of the classifier to generalise patterns previously unseen. This suggests that symmetry magnitude features alone do not provide a sufficient description of the members of class; or that in this use of pattern recognition, the various classes share very similar symmetry magnitude features. This is reinforced by the PNN smoothing parameter value which in all cases is set low such that the PNN acts as a nearest neighbour classifier.

In the noise tolerance test, the GAST and the NTGST outperformed the GST when Gaussian noise was added. However, their performance dropped noticeably to below 70%. However, this was an improvement over the tradition features based classifier which dropped below 30%. Interestingly, the RST performed worst of all even when it was set to be more tolerant to deviations from the ideal symmetric case. The noise changed the resulting symmetry magnitude features. These features are based on many contributions from evaluating pairs of points in the pattern. When evaluating pairs, the noise had affected one or both of the points such that the sum of all contributions at a given point were sufficiently different to affect generalisation.
This may be compounded by the small size of the patterns which only consist of 256 pixels. Each pixel exerts significant influence over the final symmetry magnitude feature map.

Though noise is random and highly unlikely to be symmetrical, it affects the symmetry transform’s output (in every symmetry transform considered.) A better approach to assist a classifier tolerate noise might be to identify symmetric regions or components of a pattern and only present them to the classifier. The approach of this investigation differed in that the features presented to the network described potential reflection plane information. The proposed approach could be described as using symmetry as a mask which excludes the asymmetric features (including the noise.) However, if asymmetry is a key part of a class of features, it may not be possible to employ such a mask without affecting overall classification performance. This is supported by Figure 8.12 which demonstrates that a stricter symmetry transform based pattern recognition system does not perform as well as a more tolerant one: the non-symmetrical features may be significant.

The pattern recognition experiments do not provide a means of testing the effectiveness of the RST in measuring symmetry. Park [74] et al proposed a performance evaluation system for testing the effectiveness of symmetry detection algorithms. They hand labelled three widely used data sets with human perceived symmetry features. Future work on the RST should assess its performance against other transforms using this evaluation system.
8.10 Contribution

This chapter has identified issues with the Generalised Symmetry Transform: its phase function favours four orientation pairs, failing to adequately represent other, symmetrical orientation pairs. Its intensity and distance weight functions are not functions of symmetry.

A new symmetry transform has been proposed. The Reflectional Symmetry Transform uses three functions of symmetry. Each function compares the expected reflective counterpart with the actual counterpart. No difference between the two yields a maximum value of symmetry (of one.) As the difference increases, the value of symmetry decreases (to zero.) The difference in output between the GST and RST for ten basic shapes was produced and compared in order to demonstrate the improvement.

This is a significant contribution (albeit to a limited field.) Applications and research has used the symmetry information provided by the GST. This chapter has demonstrated its failings in this regard. The RST produces the required information correctly. Because it uses the same inputs and outputs, it can replace the GST in many of these previous works. Its effects on the research of this thesis, and the work of others may lead to improvements. Alternatively, it may provide an insight into the importance of symmetry in those applications. It may be that a more accurate symmetry transform leads to worse performance. In these cases, the implication is that it is not symmetry which is useful, but some other property.

This chapter also tested the effectiveness of using NTGST and RST produced symmetry features for pattern recognition and the effect of noise added
prior to extracting the symmetry features on the generalisation performance.
Chapter 9

Discussion

9.1 Development of Ideas

The idea to investigate the importance of symmetry in pattern recognition grew from the intuitive notion of symmetry. An object that has reflectional symmetry can be completely described or represented by only one half of the shape. The remaining half is known to be the reflection of the first. The idea of reducing the dimensionality of a symmetrical pattern was viewed as constructive for pattern recognition. Determining which half should be used to describe the shape lead to the investigation of a positional and rotational normalisation mechanism in Chapters 5 and 6.

The realisation that in a perfectly symmetrical object, the most symmetrical point lies at the centroid of the object resulted in an investigation: Could it be used as the focal point in the positional and rotational normalisation mechanism? Not all the patterns used in the experiments were symmetrical. This is either in its ideal form or the result of deformation due to personal
style or ability. This lead to the idea that in such cases, the most symmetrical point would deviate from the centroid. It was proposed that it would deviate in a similar manner by all members of a given class, but deviate differently between classes. And hence, the result of the normalisation would diverge the class boundaries in pattern space with the potential to improve classification and generalisation.

The identification of individuals by their gait lead to the idea of classifying patterns by their symmetry features in Chapter 4. The work was interpreted as a classification of the change of symmetry in a manner associated with (if not directly with respect to) time. The investigation focused on classification without the temporal dimension. Many pattern recognition problems do not have a temporal dimension. These are typically termed offline recognition. The thesis thus focused on such a problem. The biometric system used the Generalised Symmetry Transform (GST). Its capability to measure reflectional symmetry, and the investigations identifying the use of reflectional symmetries in nature, was critical in justifying its use in this investigation.

The reduced performance of the classifier described in Chapter 4 suggested information had been lost in the process of performing the symmetry transform. Reviewing the premises behind the symmetry transform lead to a focus on the phase weight function. In particular, its range of output values and the propagation of information to the symmetry contribution. This resulted in the knowledge that perfectly asymmetrical point pairs yielded a phase weight of zero, which would not affect the overall symmetry map. This was the reason for the investigation in Chapter 7 of the hypothesis that asymmetry was also significant for pattern classification.
The asymmetry inclusive transforms did not yield the anticipated performance increase. Revisiting the GST led to the observation that it failed to identify two out of four reflection planes of a perfect square. This led to an analysis of the Generalised Symmetry Transform. In particular, the components of the transform’s point-pair contribution. It was noted that two out of three of the functions were not symmetry based. The third, though identified as detecting symmetry, only detected opposing edges. This led to the proposal for replacement functions, which combined with the orientation work of Chapter 6, formed a new transform: the Reflectional Symmetry Transform.

9.2 Taxonomy of GST-based Symmetry Transforms

This thesis has made extensive use of the Generalised Symmetry Transform in its experimental work. Furthermore, Chapter 7 proposed two derivative transforms for detecting asymmetry features and both symmetry and asymmetry features. Table 9.1 summarises the differences between the four GST-based transforms discussed or proposed: the original Generalised Symmetry Transform (GST) [80]; the Noise Tolerant Generalised Symmetry Transform (NTGST) [73]; the Generalised Asymmetry Transform (GAT) (§7.7); and the Generalised Asymmetry and Symmetry Transform (GAST) (§7.6).

Each transform is described by a column. The rows represent features or capabilities of the transforms. The first row, influence of edge points, refers
Influence of logarithmic edge points

<table>
<thead>
<tr>
<th>Influence of edge points</th>
<th>GST</th>
<th>NTGST</th>
<th>GAST</th>
<th>GAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Representation of Divergent Symmetry</td>
<td>positive</td>
<td>negative</td>
<td>positive</td>
<td>zero</td>
</tr>
<tr>
<td>Representation of Convergent Symmetry</td>
<td>positive</td>
<td>positive</td>
<td>positive</td>
<td>zero</td>
</tr>
<tr>
<td>Representation of Asymmetry</td>
<td>zero</td>
<td>zero</td>
<td>negative</td>
<td>positive</td>
</tr>
<tr>
<td>Sign of Magnitude</td>
<td>positive</td>
<td>positive</td>
<td>both</td>
<td>positive</td>
</tr>
<tr>
<td>Provides orientation of</td>
<td>symmetry</td>
<td>symmetry</td>
<td>both</td>
<td>asymmetry</td>
</tr>
<tr>
<td>Peak Classification Performance</td>
<td>100.0%</td>
<td>100.0%</td>
<td>100.0%</td>
<td>100.0%</td>
</tr>
<tr>
<td>Peak Generalisation Performance</td>
<td>92.5%</td>
<td>91.7%</td>
<td>93.3%</td>
<td>92.2%</td>
</tr>
</tbody>
</table>

Table 9.1: A tabulated taxonomy of the GST-based transforms.

to the involvement of a pattern’s edge points to calculate the contribution of two point pairs. The GST, GAST and GST take the logarithm of both edge points in order to minimise the effect of differences in edge intensities. However, the NTGST does not. It utilises this difference to discriminate against noise.

The next three rows highlight the differences as a result of each transform’s respective phase weight function. Divergent symmetry is defined as a pair of points whose orientations point away from each other. This occurs when \((\theta_i, \theta_j)\) equal \((\pi, 0)\) or \((\pi, 2\pi)\). Convergent symmetry is defined as a pair of points whose orientations point towards each other. This occurs when \((\theta_i, \theta_j)\) equals \((0, \pi)\) or \((2\pi, \pi)\). Asymmetrical points have equal orientations, \(\theta_i = \theta_j\). The differences described by these representations are the
most significant in defining the transforms.

The fifth row is for the sign of the symmetry magnitude, $M_\sigma$. The GST and GAT transforms use a phase weight function that output values ranging from zero to a positive real number. The summation of such contributions will always be positive. The GAST and NTGST use bi-polar phase weight functions. However, the NTGST’s symmetry magnitude is defined as the sum of the absolute value of the contributions and hence will always be positive.

The sixth row distinguishes the nature of the orientation of each point, $\phi(p)$.

### 9.3 Contribution

Chapter 2 provided a review of symmetry in literature. It contributed an original interpretation of two methods to measure the symmetry in an image, taking the form:

\[
\text{Symmetry} = 1 - \text{Asymmetry} \tag{9.1}
\]

This applied to Huebner’s Qualitative Symmetry Transform (QST) [34] and the Discrete Symmetry Transform (DST) [20]. This interpretation lead to new definitions of asymmetry derived from the transforms. This understanding demonstrates that it is mathematically more practical to accumulate the deviations from symmetry in these simple cases.

In Chapter 4, a methodology for classifying patterns by their symmetry features was proposed. In the experiment that followed, it demonstrated that
patterns could be classified by only their symmetry features. Although it was also demonstrated that the performance was not an improvement on the traditional method. That the difference in performance was small, indicated that symmetry features are still useful for describing the patterns (at least in the case of the digits).

In Chapter 5, a method for normalising the position of patterns based on their symmetry features was proposed. This method used an affine transformation with parameters set according to the features extracted by performing the Generalised Symmetry Transform to the patterns. A specific symmetry feature was identified as significant: the point with the largest symmetry magnitude.

A similar approach was applied in Chapter 6 to the problem of orientation invariance. The orientation of symmetry, of the point with the largest symmetry magnitude, was used as a parameter in an affine transformation. By combining this mechanism and that of Chapter 5, this pre-processing step would allow a classifier to become invariant to changes in pattern position and rotation. Significant in enabling this approach was the alternative definition of symmetry orientation proposed in this Chapter.

Chapter 7 investigated the significance of asymmetry. It presented two original symmetry transforms based on the GST. These transforms differ from the GST in that they are tailored to measure asymmetry. These transforms are termed the Generalised Asymmetry Transform (GAT) and the Generalised Asymmetry and Symmetry Transform (GAST). The GAT demonstrated that patterns can be classified by their asymmetry features alone. Using both symmetry and asymmetry improved performance over us-
ing either one separately. This demonstrates the importance of asymmetry together with symmetry features for generalisation.

Chapter 8 revisited the Generalised Symmetry Transform’s definition. It identified that the Phase Weight Function was not identifying all possible symmetric orientation pairs, but rather only four particular cases. The chapter contributed a new symmetry transform for detecting reflectional symmetries in an image. It identified three symmetric functions: a new phase function (based on internal angles); a new intensity function; and a more suitable distance function. An experiment was conducted to test the effectiveness of a pattern recognition system which uses features extracted by the new transform. Finally, the effect of adding noise prior to the symmetry feature extraction on pattern recognition was investigated.

9.3.1 Published Work

Preliminary reports were published for Chapters 4 [29] and 5 [30], which are reproduced in Appendix A.

9.4 Conclusions

This section will review how successful this investigation was in achieving its aims and objectives; and review the new knowledge gained from this investigation.
9.4.1 Research Aims and Objectives

The main objective of this investigation was to improve a neural networks ability to perform generalisation and to tolerate variances introduced to the patterns.

With respect to improving generalisation, the investigation was not successful. The traditional method recognises unseen patterns better than using the sets of features proposed in this thesis. However, it was demonstrated that symmetry features alone could classify a set of patterns. An analysis of the failure lead to the identification of inadequacies of the symmetry transform and the proposal of a new transform. This new transform can be used in place of the GST in all existing research and future work.

More success was had with research into invariance. Specifically, the work identified that the location and orientation of the point with the greatest symmetry magnitude was a useful feature for achieving position and orientation invariance.

9.4.2 New Knowledge

The new knowledge is applicable to a recognition problem in which there are no common symmetrical feature shared by all classes, though a subset of classes may share features. Additionally, the original pattern is presented in an offline context. That is, temporal information describing how the pattern was created is not available.

The GST-produced symmetry features alone are sufficient for the classification of a set of patterns. A network trained using these features can
sufficiently perform classification on a related but previously unseen set of patterns (that is, it can generalise.) The GST’s complement is the GAT. Using GAT-produced asymmetry features in classification yields similarly sufficient performance. The GAST provides an approach to measure symmetry and asymmetry simultaneously. Classification and generalisation performance using the GAST outperforms that using either the GST or the GAT, though only slightly.

Invariance to differences in position and orientation can be achieved by performing affine transformations using symmetry features as parameters. For position, all points should be transformed such that the point with the greatest symmetry magnitude lies at the centre of the pattern. For orientation, the pattern should be rotated such that the orientation of the point with the greatest symmetry transform faces the norm. For such a transform to work, the GST must be modified to account for the circular nature of orientation variables. The mechanisms for normalising position and orientation can be combined to produce a new mechanism which is invariant to both simultaneously. The equivalent features of the GAT and GAST can also be used for this purpose.

Finally, this investigation demonstrated that the Generalised Symmetry Transform is not a true symmetry transform, but an opposing edge mid-point detector. The research demonstrated that a more accurate reflectional symmetry detector can be implemented by considering three components of symmetry: the equality of two points’ colour intensity; the equality of the distance between each of the two points and the reflection plane; and each points’ internal angles (that is, angle between the point’s orientation, and
9.5 Further Work

This thesis has investigated the normalisation of pattern position and orientation using symmetry features. This has enabled a neural network to perform pattern recognition, invariant to differences in position and orientation. Could symmetry also be used to normalise the pattern’s size? A good starting point for such an investigation is the research of multi-resolution schemes for determining the optimal value of the distance weight function’s variance, \( \sigma_{GST} \).

This thesis has focused on the investigation of reflectional symmetries. How can rotational and transformation symmetries be used to aid pattern recognition? Can we utilise symmetry in multi-dimensional patterns? Can the three symmetry functions of the Reflectional Symmetry Transform be modified to produce a rotational symmetry transform based on the same concepts?

Symmetry has been used as a pre-classification step. Traditional programming has been used to present the end product of the transform to a network. An identical value of the distance weight function’s Gaussian width parameter, \( \sigma_{GST} \) was used for detecting symmetry at every point in the image. Could this value vary depending on the position of the point under consideration? One approach could be to linearly or non-linearly vary the variable with respect to the distance from a centre point. But an alternative approach would be set these values to better suit the specific problem do-
main. The function of measuring symmetry could be implemented as one half of a network, with $\sigma_{GST}$ acting as a weight at each point. The remaining half could perform the classification. Could this complicated network structure’s parameters be evolved using a genetic algorithm? Could evolution solve the problem of assigning distance weight function parameters tailored to the classification problem? Could such a mechanism improve generalisation?

It should be clear that there are many questions still unanswered. Rela-
tively little research has been conducted into the use of symmetry for pattern recognition.
References


Appendix A

Published Literature

This appendix contains the work of this thesis which has already been published [29, 30].
Abstract—We propose a technique to classify characters by two different forms of their symmetry features. The Generalized Symmetry Transform is applied to digits from the USPS dataset. These features are then used to train Probabilistic Neural Networks and their performances are compared to the traditional method.

I. INTRODUCTION

Symmetry is a property of many of the objects in the world around us, often a by-product of simplicity in design and manufacturing. Symmetry is not widely used to assist Artificial Neural Networks (ANNs) to perform classification. The performance of Optical Character Recognition (OCR) systems has not yet reached that of humans. We believe that symmetry features used by Neural Networks will improve recognition.

The aim of this work is to test the hypotheses that symmetry features of patterns can be used for classification; and that classification of a pattern’s symmetry features can improve generalization. To this end, we trained with a large set of handwritten characters collected from real-world samples. We partitioned the set into two groups: a training set and a test set. The performance on the test set provides an insight into the net’s ability to generalize - i.e. correctly classify new, unseen patterns which vary from those in the training set.

II. PROBABILISTIC NEURAL NETWORKS

Probabilistic Neural Networks (PNNs) have been shown to outperform the traditional Multi-Layer Perceptron (MLP) [2], [3] classifier in Optical Character Recognition (OCR) tasks [7]. Their main advantage is that their training is trivial and near instantaneous [1], [5], requiring only a single pass. Training is based on estimating probability distribution functions using Parzen windows. They can support very complex decision surfaces which approach the Bayes optimal [6].

Probabilistic Neural Networks have three layers of neurons (not including an input layer): the pattern layer; the summation layer; and the output layer. They are connected in a feed-forward manner. The structure is depicted in Figure 1.

A. Pattern Units

The pattern units are similar to the semi-linear neurons of an MLP except that they use an exponential activation-output function:

\[ g(Z_i) = e^{\frac{Z_i - 1}{\sigma^2}} \]  

(1)

A smoothing parameter, \( \sigma \), identical for all the units in the layer, controls the exponential scale factor. The activation, \( Z_i \), is defined as the sum of the weighted (\( W \)) inputs (\( X \)):

\[ Z_i = X \cdot W_i \]  

(2)

For each training pattern there is one pattern unit. The training pattern input is used as the weight for that pattern unit. Whilst this layer will be large for any non-trivial classification task, training is completed within a single pass. Recognition is slower, but training will be quicker than an MLP used for the same task.

After training, the output of these units represents how similar the new pattern is to the training patterns. The unit with the highest output represents the closest matching training pattern.

B. Summation Units

The summation units simply sum the output of pattern units belonging to the same class:

\[ f_A(X) = \sum_{i \in A} Z_i \]  

(3)

C. Output Units

The output units have two inputs and a binary output. The first input is from the summation unit of patterns belonging to that class. The second input is from a summation unit of non-class members. This input is inverted (multiplied by negative
one). Hence, the activation for the output unit is equivalent to:

\[ a = f_A(X) - f_B(X) \]

(4)

The activation-output function is a hard limiter:

\[ y = \begin{cases} 
+1 & \text{if } a \geq 0 \\
-1 & \text{if } a < 0 
\end{cases} \]

(5)

D. PNN Smoothing Parameter, \( \sigma \)

A PNN has only one parameter (other than the training patterns): The smoothing parameter, \( \sigma \), affects the network’s Probability Distribution Function (PDF) estimate. As \( \sigma \to 0 \), the network acts as a nearest-neighbour classifier. As \( \sigma \to \infty \), the network creates a non-linear decision boundary approaching the Bayes optimal. In practice, it is not difficult to find the optimal value for \( \sigma \) and the value can be modified after training of the weights is complete.

III. THE GENERALIZED SYMMETRY TRANSFORM

Reisfeld et al [4] proposed a method of measuring symmetry called the Generalized Symmetry Transform. The measure compares pairs of edge point intensities and orientations. The GST produces a detailed map of the symmetry magnitudes (as a continuous measure) and symmetry orientations, rather than produce an average symmetry position.

Identifying the location of just one reflection plane, that acts globally on an object, is of limited use to classification. Two or more classes of characters are likely to share such a reflection plane. A classifier would be unable to distinguish between these classes. The GST was selected for this experiment because it identifies symmetries which are present in only a subsection of the object. We call this reflective relationship between a subset of pixels local symmetry.

The symmetry magnitude is calculated from an approximation of the pixel greyscale intensity gradients. We can approximate the gradient using the Sobel edge operator convolution masks:

\[ \Delta p_k = \left( \frac{\delta}{\delta x} p_k, \frac{\delta}{\delta y} p_k \right) \]

(6)

A vector, \( v_k = (r_k, \theta_k) \) is associated with each pixel \( k \) in the input image consisting of a magnitude of the intensity gradient, \( r_k \), and the angle between the normal of the gradient and the horizontal axis, \( \theta_k \).

\[ r_k = \log(1 + \|\Delta p_k\|) \]

(7)

\[ \theta_k = \arctan \left( \frac{\delta_y p_k}{\delta_x p_k} \right) \]

(8)

Each pair of pixels contributes to the symmetry magnitude at a pixel mid-way between them. Hence, the set \( \Gamma(p) \) consists of all the pixel-pairs that contribute to the magnitude at a point \( p \):

\[ \Gamma(p) = \left\{ (i, j) \mid \frac{p_i + p_j}{2} = p \right\} \]

(9)

A Gaussian-based distance weight function favours contributions from pixel-pairs that are closer together than those further apart:

\[ D_\mu(i, j) = \frac{1}{\sqrt{2\pi \mu}} e^{-\frac{\|p_i + p_j\|^2}{2\mu^2}} \]

(10)

The shape of the gaussian is controlled by \( \mu \). A phase weight function favours opposing gradient orientations (rather than aligned orientations) from a pair of pixels:

\[ P(i, j) = \left[ 1 - \cos(\theta_i + \theta_j - 2\alpha_{ij}) \right] \times \left[ 1 - \cos(\theta_i - \theta_j) \right] \]

(11)

Where \( \alpha_{ij} \) is the angle between a line connection points \( i \) and \( j \), and the horizontal axis. Figure 3 shows the effect of the two edge orientations on the symmetry measure. Whilst the other components of the symmetry contribution calculate a significance (equivalent to weighting), the phase function is the continuous measure of symmetry. Two orientations are symmetrical if \( \theta_i - \theta_j = \pm \pi \). Two orientations are not symmetrical if \( \theta_i - \theta_j = 0 \).

The total contribution of two pixel points is a function of

![Fig. 2. Point pairs contribute to a symmetry magnitude and direction at a point mid-way between them.](image2)

![Fig. 3. The phase function of the Generalized Symmetry Transform, for input edge orientations from two points, in the range 0 to 2\( \pi \) radians.](image3)
the distance and phase weight functions:
\[ C(i,j) = D\mu(i,j)P(i,j)r_ir_j \] (12)

The symmetry magnitude at a given point \( p \) is the sum of the contributions over a set of pixels pairs, \( \Gamma(p) \):
\[ M\mu(p) = \sum_{(i,j) \in \Gamma(p)} C(i,j) \] (13)

The contribution has a direction, which is calculated from the two contribution gradient orientations:
\[ \phi(i,j) = \frac{\theta_i + \theta_j}{2} \] (14)

The direction of the symmetry at a specific point \( p \) is \( \phi(p) = \phi(i,j) \) such that \( C(i,j) \) is the maximum in the set \( \Gamma(p) \).

IV. EXPERIMENT

The experiment used the United States Postal Service (USPS) data set. The set contains 9292 patterns each representing a handwritten digit (the numerals zero to nine inclusive). A 16-by-16 grayscale image describes each digit. We trained with networks with 7291 patterns. The networks' performance was measured by presenting 2001 previously unseen patterns.

One network trained with the original patterns without any additional processing. A second network trained on the network symmetry magnitude map, \( M\mu \), and a third trained on the symmetry orientation map, \( \phi \).

We normalised the magnitude maps into the range –1 to 1. The GST represents no symmetry with a zero value, but these do not propagate well through weighted-sum connectionist systems.

We varied the Probabilistic Neural Networks’ smoothing parameter and measured the effect on the classification performance to find the optimal performance.

V. RESULTS

All three networks classify 100% of the training patterns. The optimal performances were: 95.17% at \( \sigma = 0.5 \) using the original data as input; 87.2% at \( \sigma = 0.1 \) for the symmetry magnitude map; and 72.2% at \( \sigma = 0.7 \) for the symmetry orientation map. The relationship between the smoothing operator and the recognition rate for all three networks is shown in Figure 4 and listed in Table I.

VI. DISCUSSION

Both symmetry feature based networks classified 100% of the training data. Hence, handwritten characters can be classified by their symmetry features. However, the network using the unprocessed USPS data set outperforms the networks using the symmetry magnitude and orientation maps in terms of generalization. The smoothing operator value indicates that all sets of data are better suited to nearest-neighbour classification, suggesting no clear separation of classes within the decision space.

Only one form of symmetry was used in this experiment: reflectional (or mirror) symmetry. It is too early to discount symmetry’s potential significance. Research into the use of rotational (or combinations of) symmetry should be conducted. Normalisation techniques applied to the magnitude map or orientation map prior to classification may improve performance.

ACKNOWLEDGMENT

This research is funded by the Engineering and Physical Sciences Research Council (EPSRC).

REFERENCES

Abstract—We propose an effective method to achieve position invariance in the application of Optical Character Recognition (OCR). We normalise the position of all inputs based on their symmetry features. The Generalized Symmetry Transform (GST) is used to determine the symmetry features prior to classification by a Probabilistic Neural Network (PNN). We used the United States Postal Service (USPS) data set to measure performance.

I. INTRODUCTION

Despite being a common feature of man-made objects, little research has been conducted on the usefulness of symmetry in pattern classification. Handwritten, offline optical character recognition performance has not yet reached that of humans. For this reason, we have chosen to investigate how symmetry features can be used by connectionist systems.

The research used the Generalized Symmetry Transform proposed by Reisfeld et al [5] to produce a continuous measure of reflectional symmetry of the patterns to be classified. An object has reflectional symmetry if it is invariant to reflection about a given plane or axis, called the reflection plane. We chose to classify the patterns using a Probabilistic Neural Network based on research comparing its effectiveness to other connectionist systems [8].

In this paper we discuss our use of symmetry features to create a classifier that was invariant to changes in position. Our approach was to normalise patterns prior to presenting them to the network.

II. PROBABILISTIC NEURAL NETWORKS

Probabilistic Neural Networks (PNNs) have been shown to outperform the traditional Multi-Layer Perceptron (MLP) [2], [4] classifier in Optical Character Recognition (OCR) tasks [8]. Their main advantage is that their training is trivial and near instantaneous [1], [6], requiring only a single pass. Training is based on estimating probability distribution functions utilising Parzen windows. They can support very complex decision surfaces which approach the Bayes optimal [7].

Probabilistic Neural Networks have three layers of neurons (not including an input layer): the pattern layer; the summation layer; and the output layer. They are connected in a feed-forward manner. The structure is depicted in Figure 1.

The pattern units are similar to the semi-linear neurons of an MLP except that they use an exponential activation-output function:

\[
g(Z_i) = e^{\frac{Z_i}{\sigma^2}}
\]

A smoothing parameter, \(\sigma\), identical for all the units in the layer, controls the exponential scale factor. The activation, \(Z_i\), is defined as the sum of the weighted (W) inputs (X):

\[
Z_i = X \cdot W_i
\]

For each training pattern there is one pattern unit. The training pattern input is used as the weight for that pattern unit. Whilst this layer will be large for any non-trivial classification task, training consists of a single pass. Recognition is slower, but training will be quicker than an MLP used for the same task.

After training, the output of these units represents how similar the new pattern is to the training patterns. The unit with the highest output represents the closest matching training pattern.

The summation units simply sum the output of pattern units belonging to the same class:

\[
f_A(X) = \sum_{i \in A} Z_i
\]

The output units have two inputs and a binary output. The first input is from the summation unit of patterns belonging to that class. The second input is from a summation unit of non-class members. This input is inverted (multiplied by negative one). Hence, the activation for the output unit is equivalent to:

\[
a = f_A(X) - f_B(X)
\]
The activation-output function is a hard limiter:
\[y = \begin{cases} 
+1 & \text{if } a \geq 0 \\
-1 & \text{if } a < 0 
\end{cases} \quad (5)\]

III. THE GENERALIZED SYMMETRY TRANSFORM

Reisfeld et al [5] proposed a method of measuring symmetry called the Generalized Symmetry Transform. The measure compares pairs of edge point intensities and orientations. The GST produces a detailed map of the symmetry magnitudes (as a continuous measure) and symmetry orientations, rather than produce an average symmetry position.

Identifying the location of just one reflection plane, that acts globally on an object, is of limited use to classification. Two or more classes of characters are likely to share such a reflection plane. A classifier would be unable to distinguish between these classes. The GST was selected for this experiment because it identifies symmetries which are present in only a subsection of the object. We call this reflective relationship a continuous measure) and symmetry orientations, rather than the symmetry contribution calculate a significance (equivalent to weighting), the phase function is the continuous measure of symmetry. Two orientations are symmetrical if \(\theta_i - \theta_j = \pm \pi\). Two orientations are not symmetrical if \(\theta_i - \theta_j = 0\).

The symmetry magnitude at a given point \(p\) is the sum of the contributions over a set of pixels pairs, \(\Gamma(p)\):

\[C(i, j) = D(\mu(i, j))P(i, j) \| r_i \| \| r_j \| \]

\[M_\mu(p) = \sum_{(i, j) \in \Gamma(p)} C(i, j) \quad (13)\]

IV. POSITION NORMALISATION

The basic idea was to reposition the contents of the pattern such that the point with the greatest symmetry magnitude was at the centre of the pattern. Where multiple symmetry axes exist, this should occur where the axes intersect. Point \(S = (s_x, s_y)\) is the point where \(S = \text{max}(M_s)\). We then apply the affine transformation:

\[
\begin{bmatrix}
x' \\
y'
\end{bmatrix} = \begin{bmatrix}
1 & 0 & 1 \\
0 & 1 & 0
\end{bmatrix} \begin{bmatrix}
x \\
y
\end{bmatrix} - \begin{bmatrix}
x_s & y_s \end{bmatrix}
\]

\[\Gamma(p) = \left\{(i, j) \mid \frac{p_i + p_j}{2} = p \right\} \quad (9)\]

A Gaussian-based distance weight function favours contributions from pixel-pairs that are closer together than those further apart:

\[D(\mu(i, j)) = \frac{1}{\sqrt{2 \pi \mu}} e^{\left(-\left|\frac{p_i + p_j}{2}\right|^2 \right)} \quad (10)\]

The shape of the Gaussian is controlled by \(\mu\). A phase weight function favours opposing gradient orientations (rather than aligned orientations) from a pair of pixels:

\[P(i, j) = [1 - \cos(\theta_i + \theta_j - 2\alpha_{ij})] \times [1 - \cos(\theta_i - \theta_j)] \quad (11)\]

Where \(\alpha_{ij}\) is the angle between a line connection points \(i\) and \(j\), and the horizontal axis. Whilst the other components of the symmetry contribution calculate a significance (equivalent to weighting), the phase function is the continuous measure of symmetry. Two orientations are symmetrical if \(\theta_i - \theta_j = \pm \pi\). Two orientations are not symmetrical if \(\theta_i - \theta_j = 0\).

The total contribution of two pixel points is a function of the distance and phase weight functions:

\[C(i, j) = D(\mu(i, j))P(i, j) \| r_i \| \| r_j \| \]

\[M_\mu(p) = \sum_{(i, j) \in \Gamma(p)} C(i, j) \quad (13)\]

IV. POSITION NORMALISATION

The basic idea was to reposition the contents of the pattern such that the point with the greatest symmetry magnitude was at the centre of the pattern. Where multiple symmetry axes exist, this should occur where the axes intersect. Point \(S = (s_x, s_y)\) is the point where \(S = \text{max}(M_s)\). We then apply the affine transformation:

\[
\begin{bmatrix}
x' \\
y'
\end{bmatrix} = \begin{bmatrix}
1 & 0 & 1 \\
0 & 1 & 0
\end{bmatrix} \begin{bmatrix}
x \\
y
\end{bmatrix} - \begin{bmatrix}
x_s & y_s \end{bmatrix}
\]

The point, \(C = (C_x, C_y)\), is the point at the centre of the image. The affine transformation was applied to the original pattern and not the symmetry magnitude map, \(M_s\). The position normalised pattern was then presented to a neural network for classification.

V. EXPERIMENT

The experiment used the United States Postal Service (USPS) data set. The set contained 9292 patterns each representing a handwritten digit (the numerals zero to nine inclusive.) We trained the networks with 7291 patterns. The network performance was measured by presenting 2001 previously unseen patterns. Based on previous research [3], the
TABLE I
RECOGNITION RATES FOR THE TWO NETWORKS, LISTED BY DATA SETS, AGAINST THE POSITION OFFSET DISTANCE.

<table>
<thead>
<tr>
<th>Distance</th>
<th>Recognition Rate, %</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Control Training Set</td>
</tr>
<tr>
<td>0</td>
<td>100.00</td>
</tr>
<tr>
<td>1</td>
<td>30.82</td>
</tr>
<tr>
<td>2</td>
<td>16.28</td>
</tr>
<tr>
<td>3</td>
<td>12.30</td>
</tr>
<tr>
<td>4</td>
<td>12.81</td>
</tr>
<tr>
<td>5</td>
<td>16.86</td>
</tr>
<tr>
<td>6</td>
<td>19.87</td>
</tr>
<tr>
<td>7</td>
<td>18.46</td>
</tr>
<tr>
<td>8</td>
<td>14.88</td>
</tr>
<tr>
<td>9</td>
<td>15.64</td>
</tr>
<tr>
<td>10</td>
<td>16.13</td>
</tr>
<tr>
<td>11</td>
<td>15.84</td>
</tr>
<tr>
<td>12</td>
<td>14.10</td>
</tr>
<tr>
<td>13</td>
<td>11.70</td>
</tr>
<tr>
<td>14</td>
<td>10.49</td>
</tr>
<tr>
<td>15</td>
<td>10.04</td>
</tr>
<tr>
<td>16</td>
<td>10.03</td>
</tr>
</tbody>
</table>

smoothing parameter $\sigma$ was set to a value of 0.4. This had achieved a recognition rate of 95.12%.

A 16x16 greyscale image described each digit in the original data set. To test the classifier’s position invariance, the input pattern’s dimensionality was 32x32 pixels. The original 16x16 pixel image was placed centrally onto the new input pattern. In effect, this added a padding of 8 pixels around the original data in which we could vary the position of the object. One network trained using this data. A second network trained on the output of the mechanism described in Section IV.

Position invariance was evaluated using fifteen additional sets of control and position normalised, training and test sets. The data sets were produced by repositioning the contents of the image in both the x- and y- dimensions.

VI. RESULTS

The optimal performance of the control classifier was 92.12%. The optimal performance of the position normalised classifier was 94.67%. The rate at which the performance decreases with respect to the position offset distance is depicted in Figure 3 and tabulated in Table I.

VII. DISCUSSION

Increasing the dimensionality of the input space did not affect the performance of the classifier. The control network’s test set performance was equal to that in previous research. Normalising the position of the training data reduced the performance slightly, but not significant at only a percentage-point difference of 0.45. The control network was not tolerant of position variances. Offsetting the pattern by only one pixel in both dimensions reduced the performance to 30.8%.

Using symmetry based position normalisation, the performance remained unchanged up to a distance of three pixels and then dropped slightly for three pixels. After six pixels, the rate of recognition-rate-loss with respect to position offset distance increased. We had anticipated the significant drop in performance to occur at $P = 8$, when there is no ‘padding’ around two edges of the original data. However, this occurred one pixel early at $P = 7$. It occurred one pixel earlier than predicted because of the nature of the gradient approximation method for the GST. We used convolution on only the visible portions of the pattern. The remaining reduction in the recognition rate was the result of important data no longer being visible - i.e. outside the boundaries of the image. This continued to worsen until $P = 16$, when none of the original data was presented to the network - only the background signal.

The results showed that an affine transformation centring a pattern on the point with the highest symmetry magnitude was effective as a position invariant mechanism for classification. The method presented here could be used as a pre-processing step for classification by any type of neural network.

ACKNOWLEDGMENT

This research was funded by the Engineering and Physical Sciences Research Council (EPSRC).

REFERENCES