Hidden Markov models for robust recognition of vehicle licence plates

by

R. P. van Heerden

9351477

Submitted in partial fulfilment of the requirements for the degree
Master of Engineering (Computer Engineering)
in the
Department of Electrical, Electronic and Computer Engineering
UNIVERSITY OF PRETORIA

January 4, 2002

Advisor: Professor E.C. Botha

Summary

Keywords: Number/licence plate recognition, neural networks, Markov models, pattern recognition, motor vehicle, image analysis, Viterbi search, segmentation

In this dissertation the problem of recognising vehicle licence plates of which the symbols can not be segmented by standard image processing techniques is addressed. Most licence plate recognition systems proposed in the literature do not compensate for distorted, obscured and damaged licence plates. We implemented a novel system which uses a neural network/ hidden Markov model hybrid for licence plate recognition.

We implemented a region growing algorithm, which was shown to work well when used to extract the licence plate from a vehicle image. Our vertical edges algorithm was not as successful. We also used the region growing algorithm to separate the symbols in the licence plate. Where the region growing algorithm failed, possible symbol borders were identified by calculating local minima of a vertical projection of the region.

A multilayer perceptron neural network was used to estimate symbol probabilities of all the possible symbols in the region. The licence plate symbols were the inputs of the neural network, and were scaled to a constant size. We found that 7 × 12 gave the best character recognition rate. Out of 2117 licence plate symbols we achieved a symbol recognition rate of 99.53%.

By using the vertical projection of a licence plate image, we were able to separate the licence plate symbols out of images for which the region growing algorithm failed.

Legal licence plate sequences were used to construct a hidden Markov model containing all allowed symbol orderings. By adapting the Viterbi algorithm with sequencing constraints, the most likely licence plate symbol sequences were calculated, along with a confidence measure.

The confidence measure enabled us to use more than one licence plate and symbol segmentation technique. Our recognition rate increased dramatically when we combined the different techniques. The results obtained showed that the system developed worked well, and achieved a licence plate recognition rate of 93.7%.

Samevatting

Sleutelwoorde: Nommerplaatherkenning, neurale netwerke, Markov modelle, patroonherkenning, motorvoertuig, beeldanalise, Viterbi soektog, segmentasie

Die doel van hierdie verhandeling is om outomaties voertuig nommerplate te herken wat nie deur tradisionele beeldverwerkingsmetodes verwerk kan word nie. Die meeste nommerplaat herkenningstelsels in die literatuur werk nie vir verwronge, gedeeltelik bedekte of beskadigde nommerplate nie. Ons het 'n nuwe stelsel geïmplementeer wat 'n neurale netwerk en verskuilde Markov model kombinasie gebruik om nommerplate te herken.

'n "Beeldomgewing-groei" algoritme is gebruik om die nommerplaat vanuit die voertuig beeld te verkry. 'n "Vertikale-rant" algoritme was nie so suksesvol nie. Ons het ook die beeldomgewing-groei algoritme gebruik om die simbole van nommerplate te segmenteer. Waar die beeldomgewing-groei algoritme gefaal het, is moontlike simbool posisies verkry deur die pieke en dale van die vertikale projeksie te analiseer.

Simbool waarskynlikhede is verkry deur die nommerplaat simbole d.m.v 'n neurale netwerk te klassifiseer. Die inset van die neurale netwerk is die nommerplaat simbole wat geskaleer is na 'n konstante grootte. Ons het bepaal dat 7 × 12 die beste karakterherkenning resultate gelewer het. Uit 2117 nommerplaat simbole het ons 'n 99.53% suksessyfer behaal.

Waar die beeldomgewing-groei algoritme gefaal het, kon ons wel daarin slaag om die simbole van 'n nommerplaat beeld te segmenteer deur gebruik te maak van die projeksie van die nommerplaat. 'n Verskuilde Markov model is toe opgestel uit toelaatbare nommerplaat volgordes en die Viterbi soek algoritme is aangepas om toelaatbare volgordes te soek. Die Viterbi algoritme soek die nommerplaat volgorde met die hoogste waarskynlikheid en allokeer 'n waarskynlikheidswaarde aan die resultaat.

Die waarskynlikheidswaarde het ons in staat gestel om veelvuldige nommerplaat- en simbool-metodes te kombineer. Ons herkenningsyfer het baie verbeter toe ons verskillende metodes gekombineer het. Die resultate toon dat die stelsel goed werk met 'n suksessyfer van 93.7% op volledige nommerplate.

Acknowledgements

I would like to thank the following people for their help and support, without which this project would not have been possible:

- Prof. Liesbeth Botha, my dedicated and patient supervisor.
- Christoph Nieuwoudt for his initial guidance.
- Darryl Purnell, Justin Schoeman and Johann Holm for sharing their expert knowhow.
- Bert van Leeuwen, Henk Peens, Claude Prinsloo, Janus Brink and Warren du Plessis for their insights and encouragement.

List of abbreviations

ANN	Artificial neural network
ARTMAP	Adaptive Resonance Theory - mapping
HMM	Hidden Markov model
LCDC	Local Direction Contributivity Density
MLP	Multilayer Perceptron neural network
MM	Markov model
NN	Neural network
OCR	Optical character recognition
URL	Universal Resource Locator

Contents

1 Introduction **1**

1.1 Approach and System Overview 2

1.2 Applications of licence plate recognition systems 3

1.3 Summary of related work 4

1.3.1 Symbol segmentation 5

1.3.2 Optical character recognition 6

1.3.3 Artificial neural networks 8

1.3.4 Hidden Markov models 9

1.3.5 Vehicle licence plate recognition 10

1.4 Dissertation outline 14

1.5 Contributions of this study 15

2 Theory **16**

2.1 Image segmentation 16

2.1.1 Thresholding 16

2.1.2 Region-growing 20

2.2 Artificial Neural Networks 21

2.2.1 Multilayer preceptron Neural Network 23

2.2.2 Training of the Neural Network 25

2.2.3 Optical character recognition (feature selection) 26

2.2.4 Probability estimation 30

2.3 Hidden Markov models 31

2.3.1 Introduction 31

2.3.2 Hybrid HMM-MLP symbol segmentation and classification . . . 32

2.3.3 Viterbi search procedure 33

2.3.4 Constrained Viterbi symbol alignment 35

3 Vehicle licence plate recognition system **37**

3.1 Image capture 38

3.2 Image segmentation 39

3.3 Licence plate location 39

3.4 Symbol segmentation 45

3.4.1 Region growing separation of symbols 45

3.4.2 Vertical projection separation of symbols 46

3.5 HMM-MLP symbol segmentation and classification 51

4 Experiments **56**

4.1 Experimental procedure 56

4.2 Optimisation of licence plate segmentation 57

4.3 Symbol segmentation 62

4.4 MLP optical character recognition 64

4.4.1 Neural Network training and optimisation 64

5 Summary and conclusion **69**

5.1 Summary of results 69

5.2 Limitations and more avenues to explore 70

A Commercial vehicle license plate systems **72**

B Recursive region growing algorithm **74**

C Threshold on the vertical projection algorithm **76**

D Valleys below a threshold algorithm **79**

E Detail results of symbol segmentation experiment 82

F Optimisation of neural network input dimensions 85

G Parameters used to obtain the optimal result 91

Bibliography 94

List of Figures

1.1 System block diagram 2

2.1 Digital image mapping 17

2.2 (a) Example of a vehicle licence plate grey-scaled image, (b) its binary image calculated with the global threshold, and (c) the local threshold method 19

2.3 A fast local thresholding technique 20

2.4 An example of a segmentation error with a broken symbol 21

2.5 An example of a segmentation error with multiple symbols segmented as a single symbol 22

2.6 An example of a binary image after region growing 22

2.7 An example of the multi-layer Perceptron neural network 23

2.8 An example of the working of a node 24

2.9 Example of a sigmoid function 25

2.10 Error rate compared versus training iterations with the training and cross validation data sets 27

2.11 Block division of licence plate symbols 28

2.12 Symbols in which the feature values of the blocks are represented by their colour 28

2.13 The split-and-merge method for character normalisation 29

2.14 An example of the split-and-merge method for character normalisation for the letter "s". 29

2.15 NN output array (scaled between 0 and 1) 30

3.1 A system overview of the vehicle recognition system 38

3.2 Images with a spectrum of threshold values 40

3.3 The licence plate symbol location process 41

3.4 Possible licence plate locations 42

3.5 The location of possible licence plate borders 43

3.6 The licence plate borders of Figure 3.5 44

3.7 Dark and brightly lit licence plates which have been separated with different threshold offsets 46

3.8 Example of licence plate images which can not be separated with binarisation 47

3.9 False separation of symbols achieved with a thinning algorithm 48

3.10 Vertical projection of licence plate images of Figure 3.8 49

3.11 An example used for symbol separation with vertical projection 50

3.12 Gauteng licence plate with vertical projection 51

3.13 The hidden Markov model used by the Viterbi algorithm of Figure 3.14 and Figure 3.15. 52

3.14 Hidden Markov model neural network hybrid example 53

3.15 Hidden Markov model neural network hybrid examples for over-segmentation 55

4.1 Optimisation of local threshold block size 58

4.2 Optimisation of local threshold offset value 59

4.3 Optimisation of horizontal extra space for un-segmented symbols 60

4.4 Optimisation of horizontal extra space for un-segmeneted symbols . . . 61

4.5 Network error for the training and cross validation set against number of iterations 66

4.6 Network error for the cross validation set against the size of the training data set 67

4.7 Network error for different hidden neuron numbers 68

C.1 Threshold on the vertical projection algorithm example 78

D.1 Valleys below a threshold algorithm example 81

E.1 Results of symbol segmentation experiment 83

E.2 Results of symbol segmentation combination experiment 84

F.1 Network error for cross validation set with different initialisation weights 87

F.2 Network error versus size of horizontal dimension 88

F.3 Network error versus size of vertical dimension 88

F.4 Network error versus size of input vector 89

F.5 Network error versus the horizontal and vertical dimensions ratio . . . 89

F.6 Network error versus input dimensions 90

List of Tables

4.1 The advantage of an iterative thresholding approach 59

4.2 Summary of optimal values obtained with experiment 1 60

4.3 Comparison between licence plate segmentation techniques 62

4.4 Comparison between symbol segmentation techniques 63

4.5 Comparison between combining symbol segmentation techniques 64

F.1 The effect of the dimensions of the mesh function used for feature extraction on the cross validation error 86

F.2 The effect of the dimensions of the mesh function used for feature extraction on the cross validation error with a larger training set 90

G.1 The best local threshold parameters used for the optimal result 91

G.2 The best region growing parameters for licence plate segmentation . . . 92

G.3 The best vertical edges parameters for licence plate segmentation . . . 92

G.4 Symbol segmentation parameters 92

G.5 Neural network parameters 93

G.6 Symbol segmentation techniques used 93

Chapter 1

Introduction

A vehicle's licence plate number is unique and its purpose is to enable quick visual identification.

The visual identification of vehicle licence plates from photographs by humans is an error prone, labour intensive and time consuming process. An automated system ought to perform the recognition in a timely and simple manner.

The shape and specifications of licence plates were chosen by their designers for easy identification by the human eye. These same specifications should also simplify automatic computer recognition, but computer-based recognition has limitations. Recognition might fail when the licence plates are not strictly within the limitations set by the specifications. Distorted, obscured and damaged vehicle licence plates are currently used by the general public, for which any robust vehicle licence plate recognition system must compensate.

The speech recognition field has developed techniques to recognise distorted and incomplete data, particularly when there is a sequence of patterns involved. Hidden Markov models (HMMs) and Markov Models (MMs) are currently used by such systems. Some of the techniques developed in speech recognition can be adapted for image processing.

This forms part of the current study in vehicle licence plate recognition.

By combining traditional image processing techniques with MMs from speech recognition techniques, a robust vehicle licence plate recognition system can be developed. This type of system is implemented and investigated in this dissertation.

1.1 Approach and System Overview

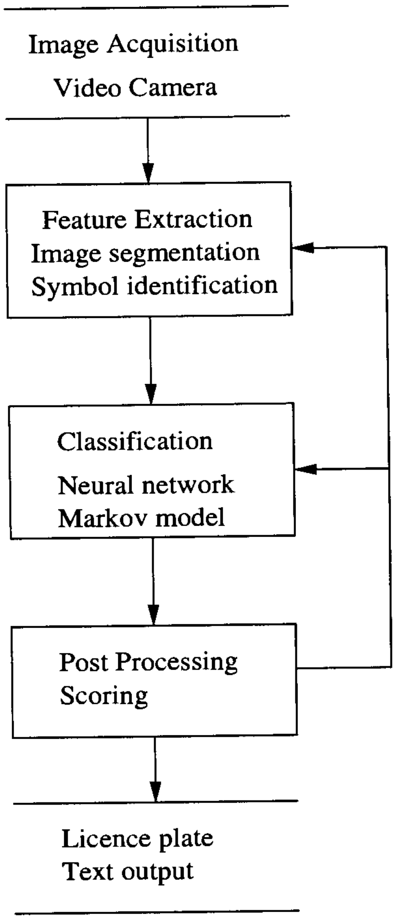


Figure 1.1: System block diagram

Figure 1.1 gives an overview of the approach used to find and identify the licence plate. A normal video camera is used to capture images containing vehicle licence plates. These images are digitised with a Matrox Meteor card in a personal computer. With feature extraction methods (thresholding and segmentation) the digitised images are prepared for classification. The classification operates only on the part of the image in which possible licence plate symbols may occur. Classification is performed with a neural network and a hidden Markov model, which scores possible licence plate sequences. The text sequence with the highest score is chosen as the output.

In this dissertation, extensive experiments were conducted to optimise the parameters used for classification, the neural network and the hidden Markov model. The influence of iterative approaches was also investigated.

1.2 Applications of licence plate recognition systems

The automatic recognition of vehicle licence plates has many possible applications. The identification of a specific vehicle can be automated with a digital system. An automatic vehicle licence plate recogniser, connected to vehicle databases can automate access control. Tollgates, pay-garages, traffic flow measurement and secure parking areas may be among the applications.

The tollgates currently in operation in South Africa require vehicles to come to a complete stop. This stoppage increases the travel time and decreases the overall productive operation of the road. An automated system which can keep track of each vehicle that passes through a specific point has obvious advantages. By adding the licence plate information to traffic data measurement, more information can be gathered. An example of this is to establish whether roads are used by local residents or as a bypass.

Pay-garages can also be fully automated with a vehicle licence plate recognition system. The dispensing of paper coupons can be taken out of the system, thus saving costs and reducing the burden on the customer to keep the coupon. Forms of fraud like the usage of fake or extra coupons can be prevented. For secure areas, a licence plate recognition system can be integrated with other security systems. In combination with access cards and face recognition, a licence plate recognition system will provide extra security.

It can be concluded that for any of the above systems to be viably implemented, a high success recognition rate is needed. The current visibility conditions of licence plates range from highly visible to obscured or damaged. Therefore a system that is robust over the whole range must be developed. This is our aim in this work.

1.3 Summary of related work

Commercial vehicle licence plate recognition systems have been available for the last few years. Since these systems are proprietary, only their operation specifications are freely available. The URLs of several of these systems are given in Appendix A. Research on vehicle licence plate recognition is rather limited and therefore we focus mainly on previous research in optical character recognition (OCR) and general pattern recognition.

A comparison between published results of different vehicle licence plate recognition systems is impractical due to several factors. The testing databases of each vehicle licence plate recognition system differ. The quality of the licence plate image influences the segmentation and the recognition results. The resolution, colour depth and lighting conditions differ for each system. These differences make the overall comparison of systems not feasible. However, the techniques used for symbol identification, recognition and syntax can still be compared and tested on a common database.

1.3.1 Symbol segmentation

The techniques to locate and separate the licence plate symbols from the background (referred to as "segmentation") were developed and adapted from techniques used to segment typed symbols in documents. The documents' symbols are dark with a light background. The background of the document is uniform, unlike the images of vehicles which have random backgrounds.

Most OCR systems scan grey-scale images, which have to be converted into binary images [1, 2]. A robust method termed local thresholding, was developed to separate characters from scene images [3, 4]. Local thresholding was also used in a system that locate bookracks in a bookrack image [5]. Trier *et al.* [4] evaluated binarisation methods and concluded that no single method performs better than the other methods, on all their images. The local thresholding technique is used to binarise our images and is further discussed in Section 2.1.

An alternative analysis technique where binarisation is bypassed is called typographic analysis [3, 6]. Typographic analysis extracts the information in the grey-scale image by computing typographic labels for each pixel. The computational intensity of typographic analysis is too high for a practical recognition system using current technology [3].

In terms of image enhancement, Jiang [7] achieved a higher recognition rate with a morphological filter which smoothed character boundaries than without the filter. Since the character boundaries in our system are not as apparent, morphological filtering will induce segmentation errors. Jiang [7] presented a variable thresholding scheme, which was implemented as decribed in Section 2.1.

Liu and Srihari [8] developed a texture feature based algorithm to binarise document images with poor contrast and/or strong noise in grey-scale histograms. Their algorithm consists of three steps: 1) Candidate thresholds are produced iteratively; 2)

Texture features associated with each candidate threshold are extracted from the run-length histogram of the accordingly binarised image; 3) The optimal threshold is selected so that the desired document texture features are preserved. The texture features include the stroke width, background noise and broken character features. These features lead Lui and Srihari's binarisation method to work better with a uniform background than a diverse background.

More robust document processing techniques than for typeset OCR were developed for handwritten recognition systems [9, 10, 11]. To handle overlapping examples of symbols, over-segmentation techniques were used [10, 12, 13, 14]. The vertical projection of the symbols are used as a guide to separate possible symbols in [10, 12, 13].

Other applications which need to locate symbols within a noisy environment are character recognition in bookshelf images [15, 5], document layout analysis [16, 17], general extraction of characters of scene images [18], finding text in live video [19, 20] and colour map text extraction [21].

Lu [22] separated broken and touching characters based on the vertical projection of the characters. Broken and touching characters are responsible for the majority of errors in automatic reading of machine-printed and handwritten text. A large number of errors with the licence plate recognition systems occur because of broken and touching characters [23, 24]. In many cases the broken and touching characters in binary images were well-formed characters in the original grey-scaled images, but which can not be separated with a single threshold value. Vertical projection technique is used in Section 3.4.1 to separate licence plate symbols.

1.3.2 Optical character recognition

Mori *et al.*[25] and Govindan and Shwaraprasad [26] present an historical overview of the development of character recognition. The goal of developing a reading machine that has the same reading capabilities of humans still remains unachieved. However, each new OCR system narrows the gap between human and machine reading.

Rodriguez *et al.* [27] developed a two-phase classifier to attain a better recognition rate with multiple fonts used with typewritten digits. The first phase carries out a multi-font classification and uses a common reference set of patterns for all forms. In the second phase, the high probability of finding just one font in one single form is the foundation for making a mono-font classification using a medium distance metric among classes in the form. The error rate was reduced by two-thirds over using just the common reference set for recognition. The fonts used for vehicle licence plates are too similar for multi-font classification.

De Beer and Botha [28] investigated several features for the recognition of handwritten digits. Five feature spaces were compared, and are listed from performing best to worst:

- LCDC feature,
- Mesh features,
- Fourier representations of characters,
- Projection features and
- Directional derivative features.

Although use of the LCDC features performed best, the performance benefit over the mesh features are not significant enough to warrant investigation. The mesh function as implemented in the system is discussed in Section 2.2.

Thinning techniques have also been used to separate symbols and to identify features for OCR systems [29]. These thinning techniques were used to generate extra training data as discussed in Chapter 4.

Muller and Herbst [30] showed that the ideas behind the eigenface technique can be extended to the OCR problem by using multiple image classes, but the recognition rate still lags behind more conventional techniques and was not investigated for the use of licence plate recognition.

Handwritten character recognition works with characters from a uniform background and uses a higher resolution. This may explain the gap in results between a vehicle licence plate system and some of the successful handwritten recognition systems.

1.3.3 Artificial neural networks

We investigated the use of artificial neural networks (ANNs) as the symbol classifier.

The ANN approach to pattern recognition is a non-algorithmic, black box strategy, which is trainable [31]. Handwritten digits have been successfully recognised with artificial neural networks [32, 33, 34, 35, 36, 14].

Lim and Chien [37] state that the recognition rate of multiple feature sets used in a single classifier is lower than multiple feature sets which have been concentrated to a single high-dimensional feature set. We believe that multiple feature sets for this licence plate recognition is an area in which more research is possible.

De Beer and Botha [28] used a multilayer perceptron classifier with one hidden layer to classify handwritten digits. Backpropagation was used for its training. We use the same training and classifier algorithms.

Martin and Pittman [38] postulate that the topology and capacity of an ANN, as measured by the number of connections in the net, have surprisingly little effect on recognition rates. They also suggest that a large and representative training sample may be the most important factor in achieving high recognition rates. The topology and training size of the neural network is discussed in Section 4.4.

Weideman *et al.* [33] found that a neural network performs comparably to the nearest neighbour classifier while being significantly more cost effective. Cho and Conner's [39] experiments showed that the performance of the neural network was slightly superior to that of the k-nearest neighbour classifier in a lumber inspection problem. Because the nearest neighbour classifier in general does not perform significantly better than the neural network classifier, we only used a neural network classifier.

1.3.4 Hidden Markov models

Hidden Markov modeling has been successfully used for speech recognition [40, 41] and has recently been applied to handwritten character recognition. One approach is to calculate the most probable syntax through a hidden Markov model. Each individual character is separated before the hidden Markov model method searches for the most likely sequence [42, 43]. We use legal licence plate sequences as the probable syntax for a hidden Markov Model.

Li *et al.* [44] recognised handwritten characters by representing each character by a sequence of symbolic strokes. The symbolic strokes are used for hidden Markov model multiple observation training or recognition. Chai and Liu [45] used a similar technique to identify handwritten characters. The number of symbols out of which a licence plate consist is much smaller than all the possibilities of handwritten characters, thus symbolic strokes are too cumbersome for a licence plate recognition system.

Arica and Vural [46] recognise digits with a hidden Markov model and use binarised normalised images, scanned in several directions as features.

Unconstrained handwritten text where syntax and character separation is needed was also addressed with hidden Markov models [47, 48, 49, 50, 51].

A neural network was used by Morgan and Bourland [52] to estimate probabilities which were integrated into a HMM based approach. The neural network translated an input sequence of acoustic vectors [53] into an output sequence which the HMM structured into sequences of speech sounds or words. The same use of a neural network to estimate probabilities is used in our system as described in Section 2.2.4.

A neural network has also been used to estimate probability density functions of handwritten symbols in hybrid hidden Markov model/ neural network systems[54, 55]. The neural network estimates the probability density functions of handwritten characters, which were subsequently used in HMMs. Neural network, hidden Markov model hybrids have also been used to classify bank cheques [56, 57]. We apply the same technique for vehicle licence plates.

1.3.5 Vehicle licence plate recognition

In the past decade, the advances in computer and image technologies made real-time, licence plate recognition systems a more affordable technology challenge. The following systems/implementations were reviewed.

The licence plate recognition system developed by Draghici [58] scanned the image horizontally looking for repeating contrast changes on a scale of 15 pixels and more. This approach to find the licence plate region uses the assumptions that the contrast between the characters and the background of the plate is sufficient, that there are at least 3-4 characters on a plate and that the characters have a minimum vertical size of 15 pixels.

Licence plate segmentation is performed using a differential gradient edge approach. This approach makes the assumption that the area of interest zone which actually contains the characters is characterised by high spatial gradient of intensity. Furthermore, it is assumed that the characters are distributed on one or more rows which are more or less horizontal and the distribution of the characters is more or less uniform.

The Draghici system performs a Sobel operation on the licence plate region with average preserving templates for vertical and horizontal edges. The result is binarised, filtered with a Gaussian filter and binarised again. The above steps are repeated with different thresholds for the binarisation step, and the average horizontal and vertical histograms are calculated. These histograms are smoothed with one-dimensional Gaussian filtering.

The estimated limits of the plate are determined by analysing the horizontal and vertical histograms. Character segmentation is performed by finding the number and location of the horizontal group(s) using binarisation and lateral histogram analysis for each group. Subsequently, the number and location of the characters which form the group are found by using lateral histogram analysis.

The proposed Barroso *et al.* [59, 60] number plate reading system searches for character-like shapes. When three or more such shapes are found in similar horizontal positions, the system looks in the neighbourhood for other similar shapes. The performance of the location process was improved with a technique based on the fact that the lines where the number plate is contained in the image have a clear "signature".

The analysis of the image lines which "cut" the number plate has been conducted both in the spatial and Fourier domain. The Fourier domain analysis has proved to be difficult. The maxima and minima of the cross section which have some predefined characteristics are searched for. Once a horizontal line that crosses the number plate has been located, that information is used to locate an area in which the licence plate is contained.

The licence plate area is calculated more accurately by using the vertical and horizontal projections of a binary version of the previous image. The threshold is dynamically chosen, based on the value of the grey level maxima and minima. The characters are segmented by searching for valleys in the vertical projection of the binary image.

Setchell [61] developed a technique based on the fact that the image of a number plate will contain a significant number of vertical edges. Vertical edges are characterised by large differences in the grey-level of adjacent pixels and are clearly visible as steps in the cross-section of the plate. The parts of an image which might possibly represent a number plate are identified by taking horizontal cross-sections through the image and searching for character-like specifications.

Possible number plate clusters are identified by the above technique. The top, bottom and angle of tilt of the possible plate is calculated by sliding a horizontal bar forwards for several intervals at the top and bottom of the possible licence plate until the number of edges reaches zero. The plate height and angle of tilt are calculated by fitting a simple model of a number plate.

Coetzee *et al.* [62] used the Niblack binarization algorithm to threshold digital grey-scaled images. A rule-based algorithm detects the position and size of number plates. The characters are segmented from the threshold plate using blob-colouring, and passed as 15x15 pixel bitmaps to a neural network OCR system. The 225 inputs are reduced to 50 features, where six small networks in parallel are used.

Siah *et at.* [63] used an iterative thresholding operation and blob analysis to calculate the position of the licence plate symbols according to the procedure defined by [23, 24]. The licence plate symbols are classified with a fuzzy ARTMAP (Adaptive Resonance Theory - mapping) neural network.

Fahmy [64] used adaptive thresholding, followed by a number plate location task to scan the image looking for the area of the number plate and outputting the co-ordinates. The co-ordinates are used to identify the location of the characters included in the licence plate. The characters are scaled to fit a 16x16 pixel image and a line-thinning algorithm is applied. The 16x16 thinned pixel image is recognised by using neural network techniques.

Lotufo *et al.* [65] used an adaptive thresholding algorithm on the entire scene. A boundary following algorithm was applied to the resulting binary image to locate the licence plate by searching for closed contours that have an appropriate size and aspect. Once the number plate has been found, a second threshold is computed and applied to the number plate area to properly segment the characters from the rest of the plate background. After the characters have been detected and have had their features extracted, a character classification process is performed that is based on the Firschien and Fischler method [65]: the characters are iteratively divided into classes and subclasses, with subclasses expressing the variations within a class.

Auty *et al.* [66] constructed a lookup table that encodes the grey values associated with roadway shadows. The images were thresholded corresponding to the lookup table. Neural network techniques were used for both location and deciphering of licence plate characters. Once located, character recognition and licence plate deciphering was performed by feature matching algorithms.

Chang and Griswold [67] obtained characters by segmenting images with grey level histogramming, thresholding, shadowing and enhanced morphological filtering. The character recognition was build up by hierarchical stacking small multi-layer perceptron neural networks that work as local classifiers. The generalisation and over-training problem was reduced.

Howington [68] analysed Percetics Corporation's first licence plate recognition system. The system first determines if a retroreflective target is present and if it has the features of a licence plate. Several tests within a decision tree structure confirm if the plate is present, locate the plate, and segment the characters within the plate. A structural analysis approach was implemented which uses a decision tree to analyse geometric features of each character's contour.

Hwang *et al.* [69] locate the licence plate by utilizing the variation of grey levels of each horizontal row or vertical column of a grey scaled licence plate image. These variations form a waveform with frequency characteristics. The licence plate characters are located by searching in a vehicle image for specific waveforms. The characters are recognised by a specific four-layered back-propagation neural network and are normalised into a 14x14 pixel array. The neural network considers the whole shape and separate detailed parts of the 14x14 array as input features. This separation helps to overcome the problem of mis-recognising similar characters.

Takatoo *et al.* [70] extract the licence plate by horizontal differentiation and dilation techniques. Changing conditions and uneven brightness are compensated for with the minimum-maximum subtraction method (variable thresholding method).

1.4 Dissertation outline

In this dissertation, robust recognition of vehicle licence plates is investigated. The dissertation consists of 5 chapters.

In Chapter 1 we introduced the licence plate recognition problem and our approach to solve the problem. We surveyed the relevant literature in the fields of: symbol segmentation, optical character recognition, neural networks, hidden Markov models and vehicle licence plate recognition.

The theory applied in the licence plate recognition system of our study is presented and discussed in Chapter 2. Neural networks and hidden Markov models are explained along with the implementation of the Viterbi search procedure.

Chapter 3 describes the licence plate recognition system used in our study in detail. The complete process from raw images to licence plate sequencing is shown.

Chapter 4 presents experimental results obtained with the system. The optimisation of all the relevant variables and features is researched in detail.

Chapter 5 concludes the dissertation. The implications of the results are discussed, and suggestions for future research are given.

1.5 Contributions of this study

The contributions of the work are:

- an extensive examination of the feature extraction process and the use of neural networks for recognition of the individual licence plate characters,
- a novel implementation and evaluation of a neural network hidden Markov model hybrid for licence plate recognition,
- an adaptation of the Viterbi search procedure specifically for licence plate recognition.

Chapter 2

Theory

The theoretical aspects of the techniques used in the development of the licence plate recognition system are presented and discussed in this chapter.

2.1 Image segmentation

All the operations and techniques are conducted on grey-scale digitised images. The digital image is represented with a discrete array $f(x,y)$, $0 < x \leq X_{max}$, $0 < y \leq Y_{max}$, $0 \leq f \leq f_{max}$, and $x, y, f \epsilon \mathbb{N}$. $f(x,y)$ represents the luminance or grey-scale value of each pixel; x and y represent the horizontal and vertical position of the pixel respectively. The mapping of the digital image is shown in Figure 2.1.

2.1.1 Thresholding

The goal of thresholding is to reduce a grey-scale image to a binary image. The binary image properties are determined by the thresholding technique used. Thresholding techniques offer a simplified approach to the problem of extracting licence plate symbols

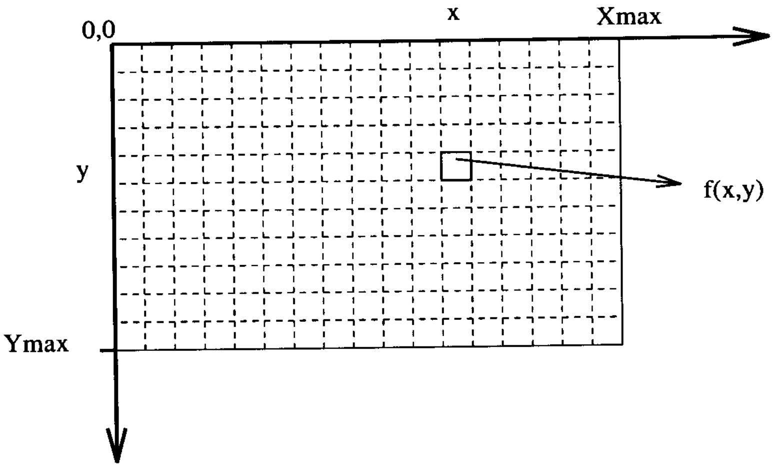


Figure 2.1: Digital image mapping

from their background. The licence plate symbols normally occupy a separable grey-level range in the grey-scale histogram of the complete image. Thresholding can be defined as a mapping of grey scale onto the binary set $\{0,1\}$, i.e.

$$B_{xy} = \begin{cases} 0 & \text{if } f_{xy} < T_{xy} \\ 1 & \text{if } f_{xy} \geqslant T_{xy} \end{cases} \tag{2.1}$$

where B_{xy} is the binary pixel value of the segmented output, f_{xy} is the grey level of a pixel and T_{xy} is the threshold value.

The quality of the binarisation step is critical for any subsequent analysis. The desired properties which will simplify the extraction of licence plate symbols out of a binary image are:

- separation of the background and the desired foreground,
- reducing the complexity of the image, and
- reducing the data-size requirement.

The available thresholding techniques [71] can be categorised as follows:

- interactive versus automatic
- global versus local
- contextual versus non-contextual
- iterative versus non-iterative
- supervised versus unsupervised.

Interactive thresholding requires a user to specify pixels belonging to the background and pixels belonging licence plate symbols. It is not a viable option for any automated system. Automated thresholding techniques usually only adapt the window size, offset values and number of labelled pixels.

A global thresholding scheme uses a single threshold to segment the entire input image, whereas a local method computes a threshold value for each individual pixel or local group of pixels. Global threshold techniques fail to separate licence plate symbols from their background if the lighting conditions differ across the licence plate. This is illustrated in Figure 2.2 (b), where the licence plate symbols can not be successfully separated with any global threshold value.

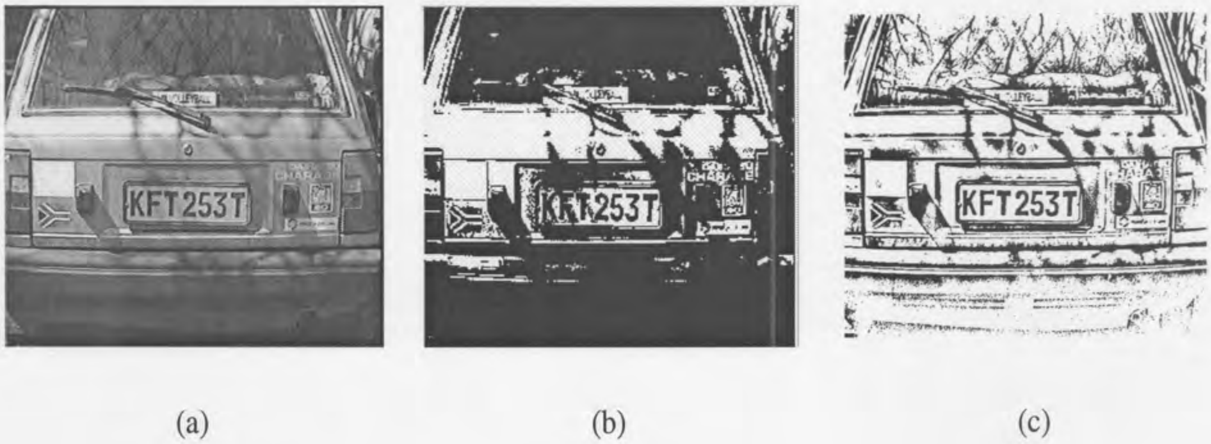
With local thresholding [3, 72, 73] the threshold value for each pixel may differ. Each pixel threshold $g(x, y)$ is calculated separately as follows:

$$B(x,y) = \begin{cases} 0 & \text{if } f(x,y) < g(x,y) + \phi \\ 1 & \text{if } f(x,y) \geqslant g(x,y) + \phi \end{cases} \tag{2.2}$$

where

$$g(x,y) = \frac{1}{N_x N_y} \sum_{i=x-\frac{N_x}{2}}^{x+\frac{N_x}{2}} \sum_{i=y-\frac{N_y}{2}}^{y+\frac{N_y}{2}} f(i,j). \tag{2.3}$$

N_x and N_y are the horizontal and vertical dimensions of the region which influences the threshold. ϕ represents a constant offset.



(a) (b) (c)

Figure 2.2: (a) Example of a vehicle licence plate grey-scaled image, (b) its binary image calculated with the global threshold, and (c) the local threshold method

A faster method to calculate a local threshold is achieved by dividing the original image into sub blocks, with each sub block's threshold calculated individually. The threshold of each pixel is then calculated as a linear interpolation of the four surrounding threshold values, as shown in Figure 2.3.

Two segmentation errors can occur namely that the threshold value can be too high which leads to broken characters as shown in Figure 2.4, or multiple characters can be segmented as a single character when the threshold is too low, as shown in Figure 2.5.

One threshold value can not successfully separate Figure 2.4 and Figure 2.5 without inducing segmentation errors. By iteratively changing the threshold value, a possible threshold value will work for Figure 2.4 and another threshold value for Figure 2.5. The iterative process is explained in Section 3.3.

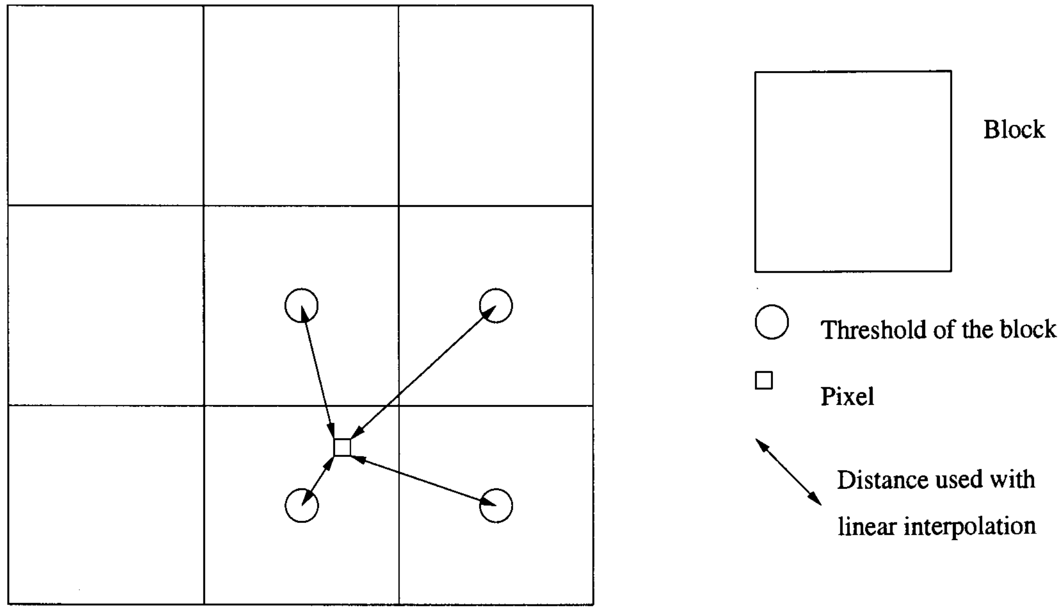


Figure 2.3: A fast local thresholding technique

2.1.2 Region-growing

Once the licence plate symbols have been thresholded, they are separated from the uniform licence plate background colour. Thus the licence plate symbols can be extracted from the image by region-growing algorithms, which separate and group the pixels of dark areas surrounded by light areas.

The region growing algorithm specify the position and size of each dark region. These regions are also known as blobs. An example of a image which has been separated with a recursive region-growing algorithm is shown in Figure 2.6. Each separate region or blob is shown in a different shade of grey. The algorithm used for region growing is shown in Appendix B.

The region in which the whole licence plate occur is identified with the dark regions' features. This process is described in Section 3.3. The region growing algorithm is also used to identify the individual symbols. The process of identifying each symbol is described in Section 3.4.1.

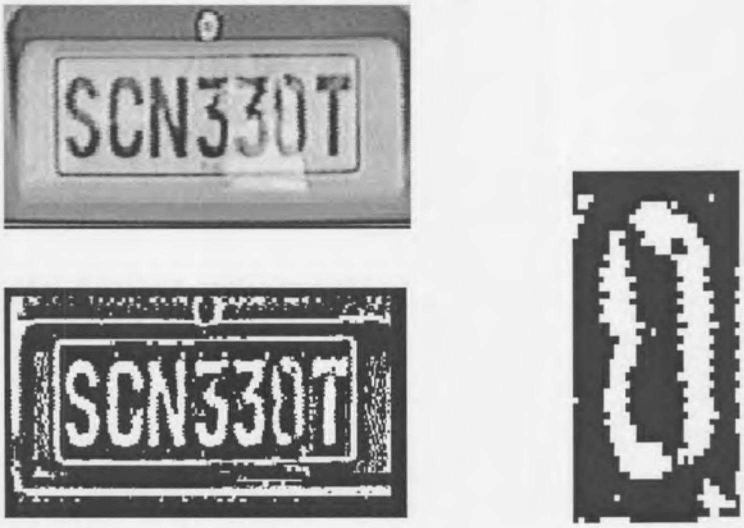


Figure 2.4: An example of a segmentation error with a broken symbol

2.2 Artificial Neural Networks

The classification process needs to convert the symbol images into computer based alpha-numeric symbols. ANNs are used in the first step of this role.

Artificial neural networks were inspired by research into the human brain [74, 31]. An ANN acts on data by detecting some kind of underlying structure, and can recognise spatial, temporal, or other relationships [75] to solve classification, prediction and function estimation problems. Problems with the following characteristics may be suited for an ANN solution [31]:

- a high-dimensional problem space,
- complex interactions between problem variables,
- an empty solution space,
- a solution space which contains a number of useful solutions,
- problems dealing with flawed, missing data.

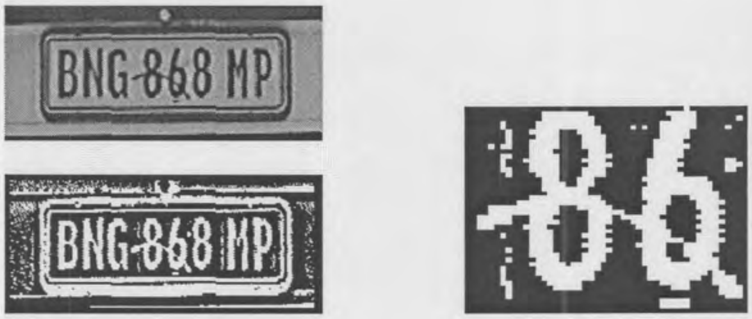


Figure 2.5: An example of a segmentation error with multiple symbols segmented as a single symbol

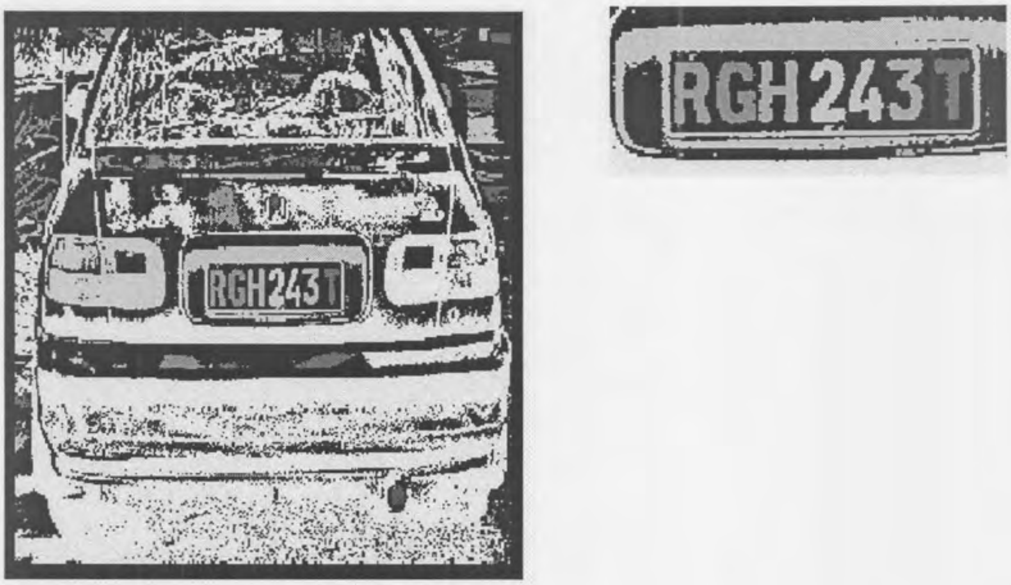


Figure 2.6: An example of a binary image after region growing

The problem of classifying licence plate symbols exhibits some of the characteristics mentioned above.

2.2.1 Multilayer preceptron Neural Network

The Multilayer Perceptron neural network (MLP) also known as the multi-layer feed-forward network, can be used to classify licence plate symbols. The MLP is composed of a hierarchy of processing units, organised in a series of two or more mutually exclusive sets of neurons or layers [31]. The first, or input serves as holding site for the values applied to the network. The last, or output layer is the point at which the final state of the network is read. Zero or more hidden layers lie between the input and output layer. Links or weights connect each node in one layer to nodes in the next layer. The output of a node, scaled by the value of a connecting weight, is fed forward to provide a portion of the activation for a unit of the next layer. Figure 2.7 illustrates the MLP.

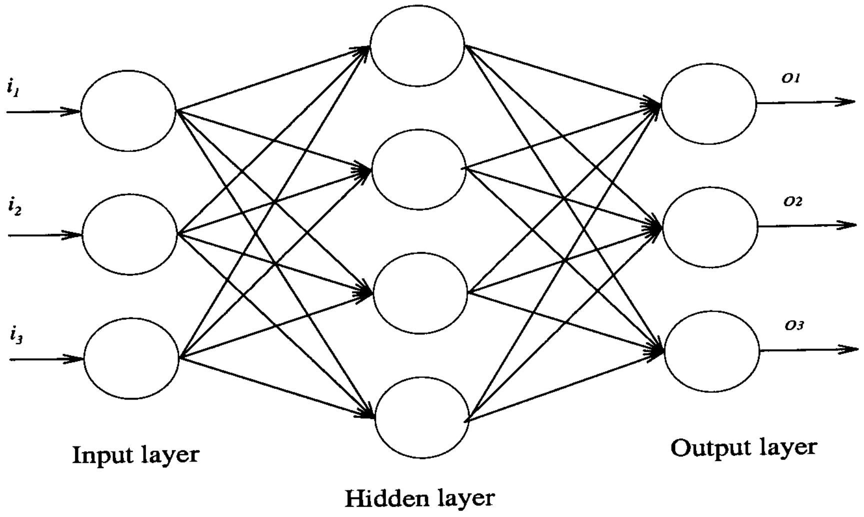


Figure 2.7: An example of the multi-layer Perceptron neural network

The input layer's role is only to 'hold' the input values, and to distribute these values to the nodes of the next layer. These input values are passed to the units of the next layer, which receive a weighed value from each of the input nodes. The sum of the weighted input values (called the activation of the node) is passed through a transfer function. Each node distributes its value to all the nodes of the next layer. These values are again individually weighted. The working of a node is illustrated in Figure 2.8.

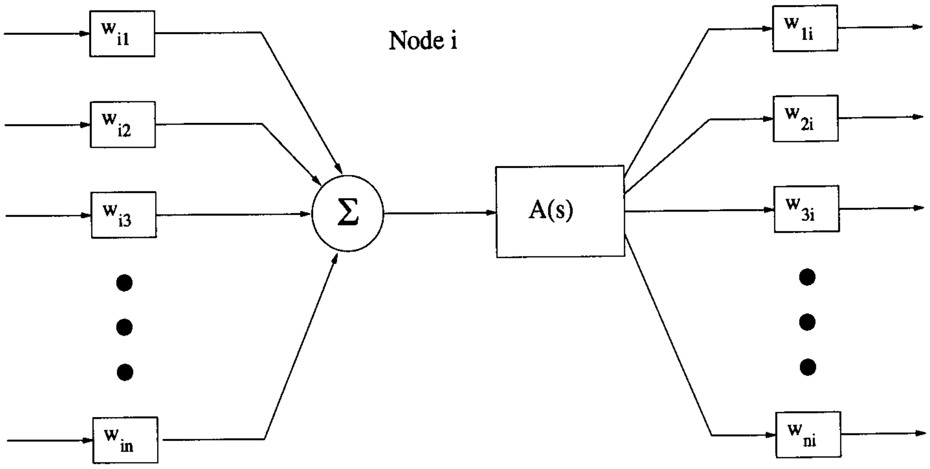


Figure 2.8: An example of the working of a node

The transfer function ($A(s)$) shown in Figure 2.8) maps the node inputs to an output value. Common transfer functions are [31]:

- linear, with a adjustable gain,
- threshold (Heaviside step function),
- linear threshold and
- sigmoid functions.

The sigmoid function, as given in Eq. (2.4) is one of the most popular functions used because of its smooth non-linear qualities and its correspondence to the average firing frequency of biological neurons [31]:

$$A(S_i) = \frac{4}{1 + e^{-\lambda S_i}}. \tag{2.4}$$

As seen in Figure 2.8 each node input has an associated weight, indicating the strength of its connection with either an external input or another node output (node of a previous layer). In Figure 2.8 w_{ij} represents the strength of the connection neuron from neuron j to neuron i. A large, positive value of w_{ij} indicates a strong input, and a large negative value is considered inhibitory [31].

In Eq. (2.4) $A(S_i)$ is the activation of a node, with S_i representing the sum of the weighted inputs and λ is an adjustable gain parameter that controls the 'steepness' of the input transition, as shown in Figure 2.9.

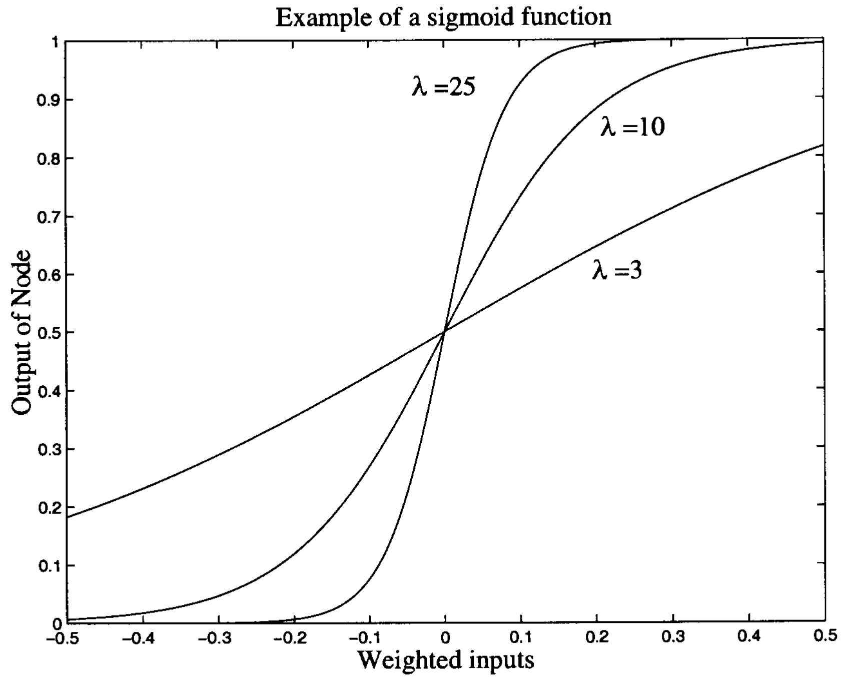


Figure 2.9: Example of a sigmoid function

The values of w_{ij} are trained off-line. It is not required for the licence plate recognition system that the network have on-line learning capabilities. The training algorithm used must ensure that the neural network (NN) has a high level of accuracy, and must be fast enough to train the network in a reasonable time.

2.2.2 Training of the Neural Network

The network training time is not a primary consideration because it occurs off-line and thus has no influence on the speed of the the final system. Only the feed-forward execution time is of importance, but thus is generally negligible. The back propagation algorithm with conjugate gradient optimisation was chosen to train the neural network

[76, 31, 77].

De Villiers and Barnard [77] concluded that three- and four-layered networks perform similarly in all performance respects. The number of input and hidden nodes is empirically calculated (see Section 4.4). Each output node of the network represents a single alpha-numeric symbol of the licence plate.

Over-training of a neural network can occur when the neural network is trained for too many iterations. In the case of overtraining, the neural network learns to classify the training examples extremely accurately but it loses its ability to generalise [61, 78]. To counter the over-training effect, a separate training set of data termed a "cross validation" set is used. After each iteration of training the error that the network makes with the cross validation set is calculated. When the network begins to overtrain, the error for the validation set will begin to rise. The optimal training point for the neural network is at the global minimum of the cross validation error set, as shown in Figure 2.10.

Over-training can be reduced by increasing the size and scope of the data set. The neural network's ability to specialise is reduced if the data set is large and representative. The influence of overtraining was tested in Chapter 4, Section 4.4.

2.2.3 Optical character recognition (feature selection)

The input features used by the neural network have a large influence on the performance of the whole system. The features used to identify the character symbols on our system must have the following properties:

- able to separate all the different classes,
- able to compensate for skew and out-of-plane rotated symbols, and
- size invariance.

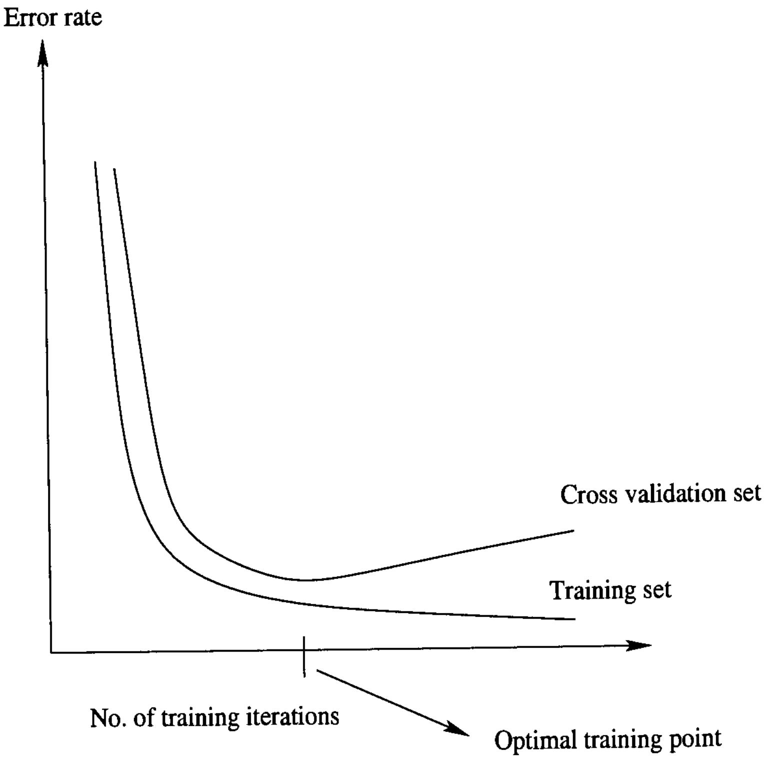


Figure 2.10: Error rate compared versus training iterations with the training and cross validation data sets

By fitting a scaled mesh function on the binary licence plate symbol, a specified number of features can be selected [23, 24, 38, 32, 36, 35]. The fraction of 'on' (black) pixels in each block represents the value of each feature. If a block is completely white then the value of that block will be 0, and the value will be 1 if a block is totally black. If half the pixels of the block are white the value of that block will be 0.5. In Figure 2.11 an example of the block division for features is shown. Each block is assigned a specific value as illustrated in Figure 2.12, where the greyness of a block is proportional to its value.

Smoothing [7] the input symbols have minimal effect on the mapped mesh function. The mapping of any figure to a mesh function can also be achieved by normalising the image to the dimension of the mesh function. Each pixel of the smaller image represents one of the features. The split-and-merge[69] method for normalising image

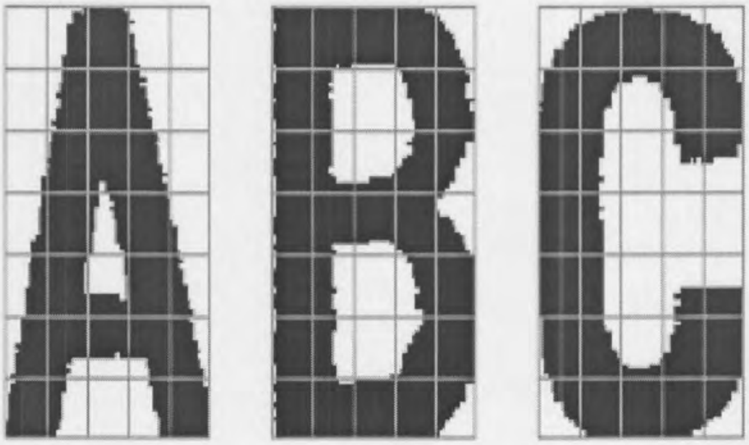


Figure 2.11: Block division of licence plate symbols

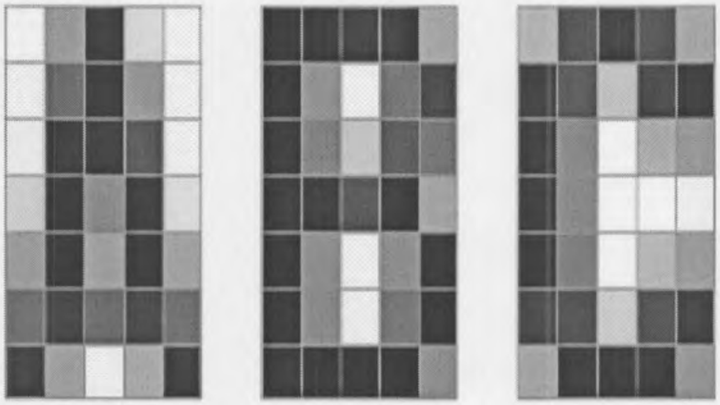


Figure 2.12: Symbols in which the feature values of the blocks are represented by their colour

dimension is shown in Figure 2.13, and now explained.

Suppose the original character size is $M \times N(3 \times 3)$ pixels, and we want to normalise it to $P \times Q(4 \times 2)$ pixels. In the split process, the $M \times N(3 \times 3)$ array will be copied by a factor P for the row and factor Q for the column. So the $M \times N(3 \times 3)$ array becomes a $MP \times NQ(12 \times 6)$ array. In the merge process, every $M \times N(3 \times 3)$ area in the $MP \times NQ(12 \times 6)$ array will be averaged to a single cell. So the $MP \times NQ(12 \times 6)$ array is normalised to a $P \times Q(4 \times 2)$ array. An example is shown in Figure 2.14 .

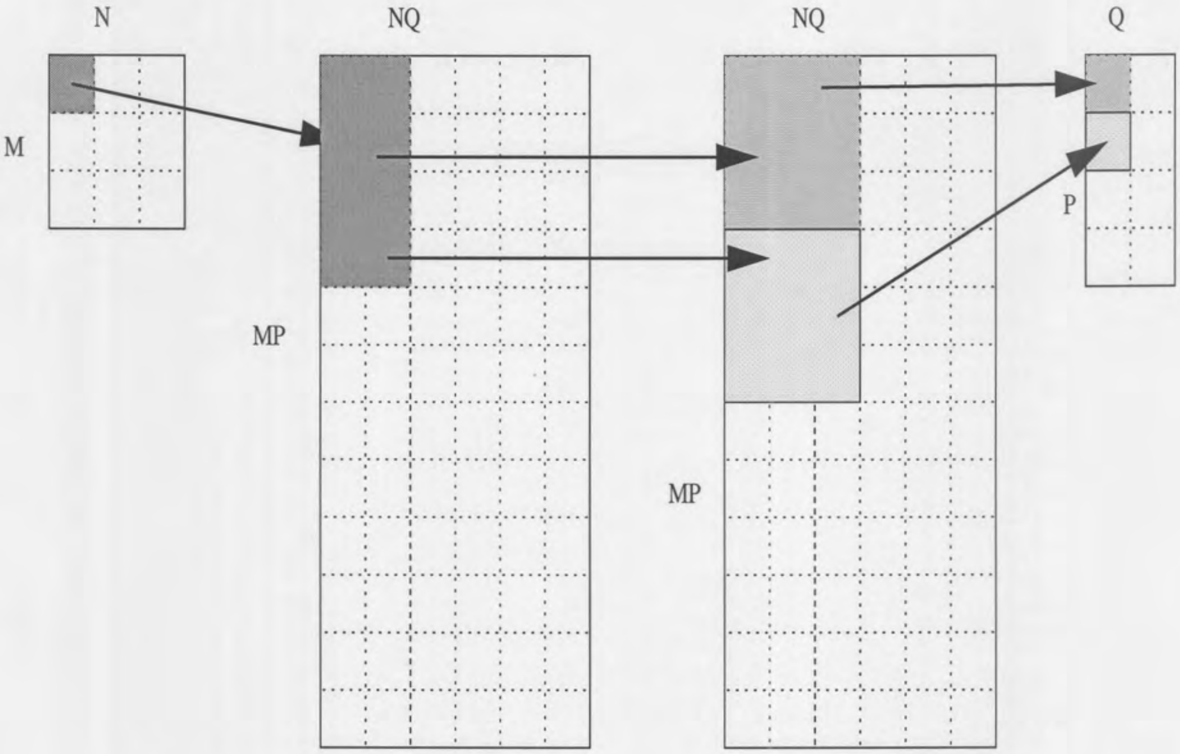


Figure 2.13: The split-and-merge method for character normalisation

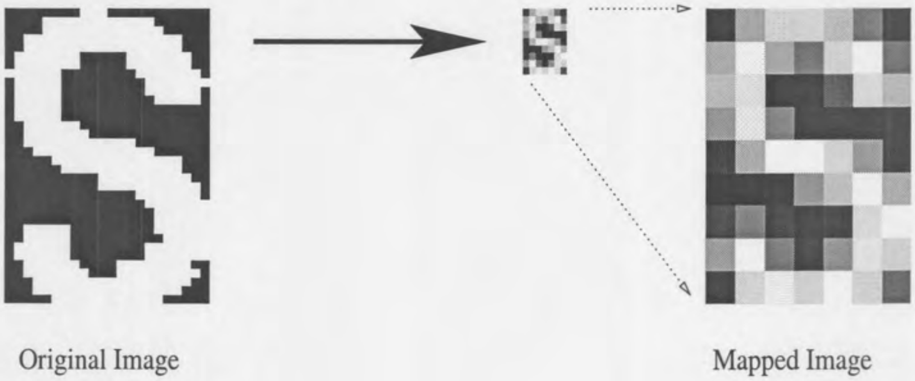


Figure 2.14: An example of the split-and-merge method for character normalisation for the letter "s".

The split and merge technique was used to generate an input vector for the neural network.

2.2.4 Probability estimation

The postprocessing stage of the system uses the probability estimations to score possible licence plate sequences.

The neural network outputs an array of values. Each value maps to a single class of character. The output nodes of the neural network restrict the possible outputs within the range (0 and 1) as determined by the transfer function. For each output set, each class of character will have its own single output value. In Figure 2.15 the output values of a "B" character is shown. The output values which map to "B" and "8" characters are the highest. Visually the difference between those two characters is less than between any of the other characters.

The neural network outputs are directly used as symbol probabilities, as supported by the literature [79]. Thus we estimate probabilities with a neural network.

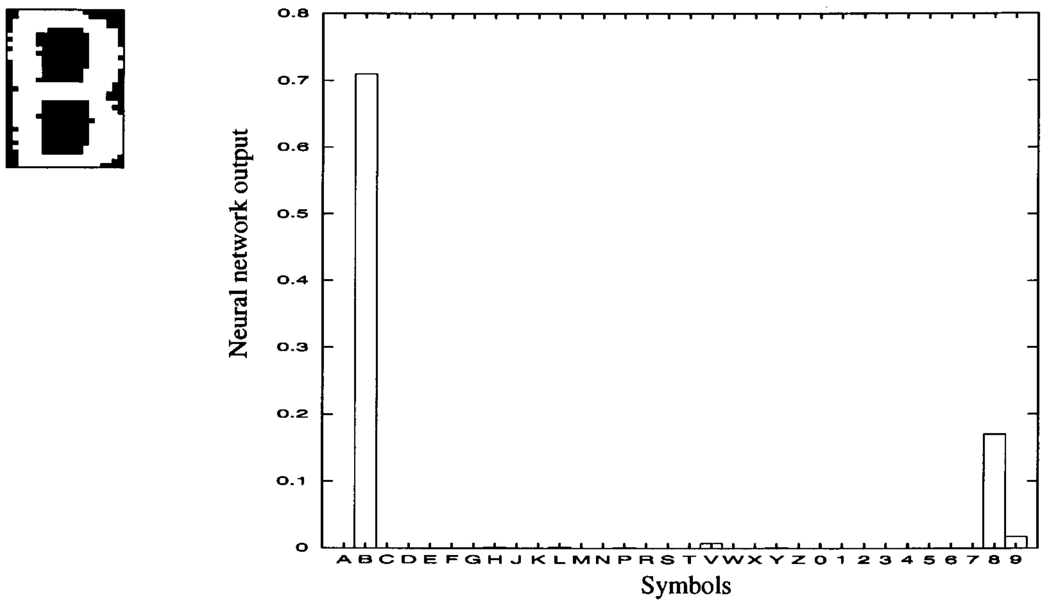


Figure 2.15: NN output array (scaled between 0 and 1)

2.3 Hidden Markov models

2.3.1 Introduction

Hidden Markov modelling has been used with great success in speech recognition [40, 41]. We seek to apply some of the same concepts to licence plate recognition.

An HMM is characterised by the following:

- N, the number of states in the model,
- M, the number of distinct observation symbols per state,
- the state transition probability distribution $A = (a_{ij})$,
- the observation symbol probability distribution in state B_j, and
- the initial state distribution π.

We denote the individual states as

$$S = s_1, s_2, \cdots, s_N, \tag{2.5}$$

the individual symbols as

$$V = v_1, v_2, \cdots, v_M, \tag{2.6}$$

the state of the HMM at time t as q_t and the state transition probability distribution $A = \{a_{ij}\}$ as

$$a_{ij} = P[q_{t+1} = s_j | q_t = s_i] \qquad 1 \leq i, j \leq N. \tag{2.7}$$

The observation symbol probability distribution in state j, $B_j = \{b_j(k)\}$, is given by

$$b_j(k) = P[v_k \quad at \quad t | q_t = s_j], \qquad 1 \leq j \leq N, 1 \leq k \leq M \tag{2.8}$$

and the initial state distribution $\pi = \{\pi_i\}$ where

$$\pi_i = P[q_1 = s_i], \qquad 1 \leq i \leq N. \tag{2.9}$$

Given the appropriate values of N, M, A, B and π, the HMM can be used as a generator to give an observation sequence,

$$M = O_1 O_2 \cdots O_T \tag{2.10}$$

where each observation O_t is one of the symbols from V and T is the number of observations in the sequence.

Three basic problems must be solved for HMMs to be useful in the application of a system:

- How to calculate the probability of the observation sequence, given the observation sequence O and the model (A, B, π) ? (Recognition)
- How to choose a corresponding state sequence $Q = q_1 q_2 \cdots q_T$ which is optimal, given the observation sequence $O_1 O_2 \cdots O_T$ and the model (A, B, π) ? (Recognition)
- How to adjust the model parameters (A, B, π) ? (Training)

2.3.2 Hybrid HMM-MLP symbol segmentation and classification

The problem of calculating the probability of an observation sequence can also be viewed as the problem of scoring how a given model matches a given observation sequence[41, 40].

To exhaustively search for an optimal state sequence is impractical [80]. Fast search techniques can be orders of magnitude more efficient in processing and memory requirements [80]. The Viterbi algorithm [81] is an efficient algorithm for finding an optimal solution. The Viterbi algorithm is described in Section 2.3.3.

The HMM model parameters (A, B, π) were all set to 1 for legal transitions (A), valid symbols in (B) and initial states. The rest were set to 0. Thus (A), (B) and (π), are not truly probabilities, but the system will still work because all the "on" values are set to 1. The data set was not large enough to calculate meaningful values.

In Section 2.2.4 it will be explained that symbol probabilities can be estimated with an NN. NN and HMM combinations have been used for speech recognition and handwriting recognition [55, 56, 53]. In such systems and in ours the NN output values are used directly as symbol probabilities.

2.3.3 Viterbi search procedure

The Viterbi search algorithm is a formal technique for finding the single best state sequence [80, 81, 43, 45] given an observation sequence and an HMM. The Viterbi algorithm used with the hidden Markov model can be summarised as follows [81]:

Initialisation:

$$\delta_1(i) = \pi_i b_i(o_1) \qquad 1 \leq i \leq N \tag{2.11}$$

where π_i represents the initial probability for state i, $b_i(o_1)$ represents the probability of observing o_1 in state i, δ_i is the best score along a single path, ending in state i, and N is the number of states in the HMM.

Recursion:

$$\begin{aligned} For \qquad & 2 \leq t \leq T, 1 \leq j \leq N \\ \delta_t(j) = & \max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}] b_j(o_t) \end{aligned} \tag{2.12}$$

$$\phi_t(j) = \arg\max_{1\leq i\leq N}[\delta_{t-1}(i)a_{ij}] \tag{2.13}$$

where a_{ij} represents the transition probability from state i to state j and ϕ keeps track of the optimal state sequence.

Termination:

$$\delta^* = \max_{1\leq i\leq N} [\delta_T(i)X_i] \tag{2.14}$$

$$q_T = \arg\max_{1\leq i\leq N}[\delta_t(i)X_i] \tag{2.15}$$

where X_i denotes the probability that state i is a final state in the hidden Markov chain and δ^* denotes the probability associated with the most likely sequence. The most likely state sequence is given by $\mathbf{q}$, which ranges from q_1 for the first state up to q_T for the final state.

Backtracking:

$$t = T-1, T-2, \ldots, 1$$

$$q_t = \phi_{t+1}(q_{t+1}). \tag{2.16}$$

We implement an HMM by using left-to-right sequenced frames of licence plate symbols. A pseudo probability vector of each licence plate symbol is calculated with a neural network. These probability vectors form the observations of each state.

All legal licence plate sequences form the states of the HMM. The possible subset of legal observations (complying with legal licence plate sequences) for each state form the distinct observation symbols for each state.

The state transition probabilities are set to 1 if it is a legal transition, and 0 if it is not a legal transition. In Section 3.5, a detailed example is shown.

2.3.4 Constrained Viterbi symbol alignment

The Viterbi algorithm needed to be adapted for over-segmentation of licence plate symbols. The over-segmentation process will be explained in Section 3.5 (Figure 3.15). The Viterbi algorithm was constrained by the following rules:

- each symbol is one or two frames wide,
- each symbol is directly followed by the next symbol, and
- the frames of viable symbols will not overlap.

These rules are explained in more detail in Section 3.4.2. An example of over-segmented licence plate is shown in Section 3.5. The Viterbi algorithm now becomes:

Initialisation:

$$\delta_i(1w) = \pi_i b_i(o_{1w}) \qquad 1 \le i \le N, 1 \le w \le 2 \tag{2.17}$$

where π_i represents the initial probability for state i, $b_i(o_{tw})$ represents the probability of observing o_{tw} (image area starting at frame t with a width of w frames) in state i, δ_{itw} is the best score along a single path, ending in state i, at frame $t+w$ and N is the number of states in the HMM.

Recursion:

$$For \qquad 2 \le t \le T, 1 \le w \le 2, 1 \le j \le N$$

$$\delta_j(t,w) = \max_{\substack{1 \le i \le N \\ 1 \le w' \le 2}} [\delta_i(t-w',w')a_{ij}]b_j(o_{tw}) \tag{2.18}$$

$$\phi_t(j) = \arg\max_i [\delta_i(t-w',w')a_{ij}] \tag{2.19}$$

$$\varphi_t(j) = \arg\max_{w'} [\delta_i(t-w',w')a_{ij}] \tag{2.20}$$

where a_{ij} represents the transition probability from state i to state j and ϕ and φ keep track of the optimal state sequence and symbol width sequence respectively.

Termination:

$$\delta^* = \max_{\substack{1 \leq j \leq N \\ 1 \leq w \leq 2}} [\delta_{j(T-w)w} X_j] \tag{2.21}$$

$$q_T = \arg\max_j [\delta_j(T-w, w) X_j] \tag{2.22}$$

$$\varphi^* = \arg\max_w [\delta_j(T-w, w) X_j] \tag{2.23}$$

where X_j denotes the probability that state j is a final state in the hidden Markov chain, δ^* denotes the probability associated with the most likely sequence and φ^* denotes the last frame width of the sequence. The most likely state sequence is given by $\mathbf{q}$, which ranges from q_1 for the first state up to q_T for the final state.

Backtracking:

$$q_{T-\varphi^*} = \phi_T(q_T) \tag{2.24}$$

$$q_{t-\varphi_t(q_t)} = \phi_t(q_t) \qquad 1 \leq t \leq T - \varphi^*. \tag{2.25}$$

Chapter 3

Vehicle licence plate recognition system

The vehicle licence plate recognition system consists of the following main modules, which will be described in detail in this chapter.

- image capture,
- image segmentation,
- licence plate location,
- symbol segmentation,
- HMM-MLP symbol segmentation and classification.

Figure 3.1 shows an overview of the system. The optimal solution of image segmentation licence plate location and symbol-segmentation may differ from one image to the next, which lends itself to an iterative approach.

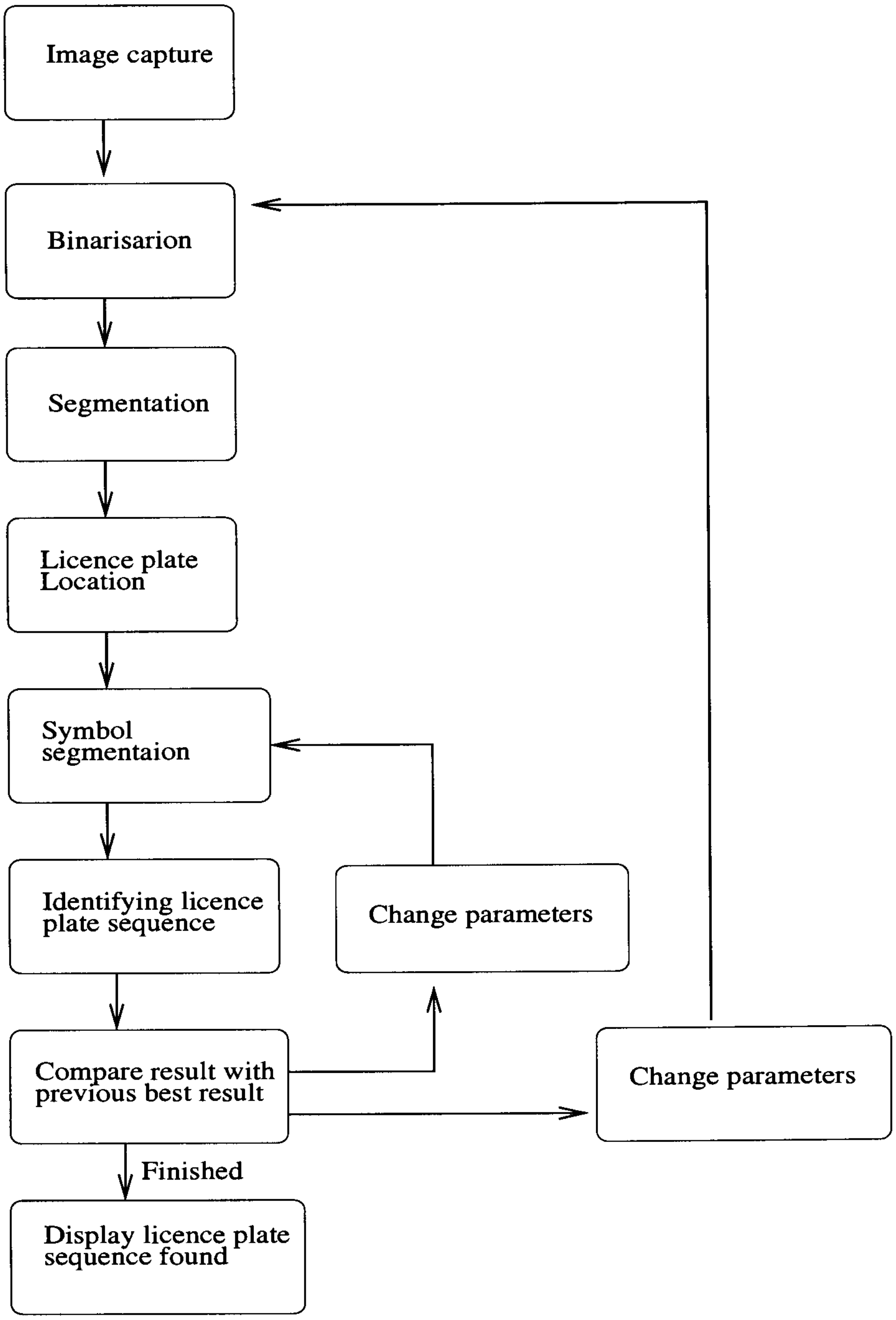


Figure 3.1: A system overview of the vehicle recognition system

3.1 Image capture

A standard portable video and a security camera was used to collect grey-scaled images. The minimum input resolution is 512 × 480 pixels, and the maximum resolution is

768 × 576 pixels.

The angle between the camera and the licence plate affects the licence plate location and the character recognition rate. For the best results, the angle between the licence plate and the camera need to be as close to 90^o as possible.

3.2 Image segmentation

The threshold process is applied iteratively and includes the whole spectrum of usable threshold values. For each threshold value used, the system returns an overall recognition score. Thus a spectrum of theshold values can be tried. The licence plate returned with the highest score is then retained, thus using the optimal threshold value.

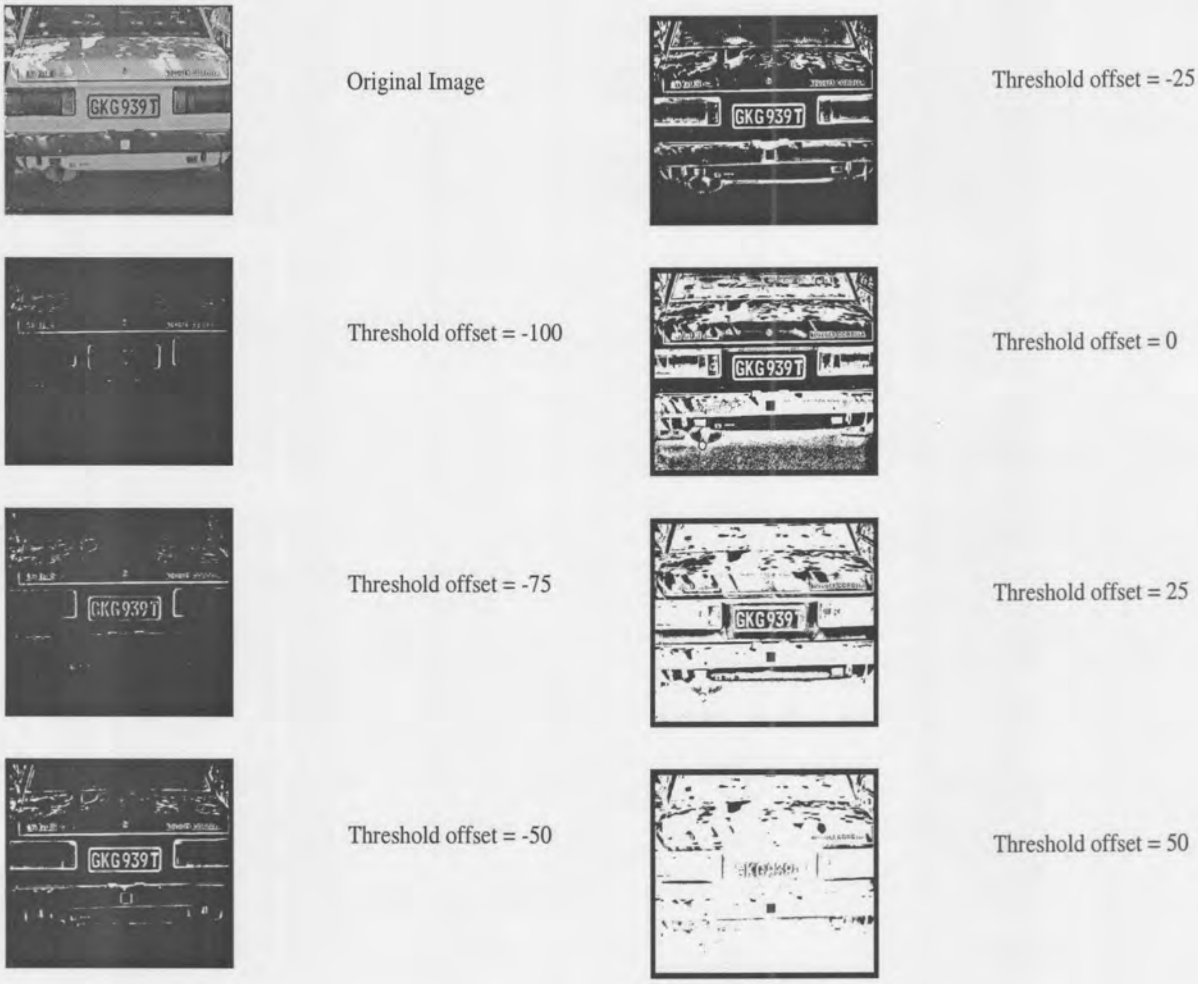
The iterative process eliminates the need to find the optimal threshold. The Niblack Binarisation algorithm threshold value used by Coetzee et al.[62] will fall within the range of threshold values chosen. In Figure 3.2, images with a spectrum of threshold values are shown.

3.3 Licence plate location

After the region growing algorithm, each region's features were used to extract the position of the licence plate symbols. The regions (blobs) are areas of touching pixels which have the same value.

The features used in locating the licence plate in the binary image were the position, dimensions and size of the blobs. This process is illustrated in Figure 3.3.

The following steps were used to identify the licence plate symbols:



Original Image

Threshold offset = -25

Threshold offset = -100

Threshold offset = 0

Threshold offset = -75

Threshold offset = 25

Threshold offset = -50

Threshold offset = 50

Figure 3.2: Images with a spectrum of threshold values

- the number of pixels in each region must fall within certain specifications,
- the height and width must fall within certain specifications,
- the width versus height ratio must fall within certain specifications,
- at least three other regions must be of similar height and of a similar vertical value.

A new algorithm based on the fact that a licence plate contains a significant number of vertical edges [61], was also tested. Vertical edges are characterised by large differences in grey-level of adjacent pixels.

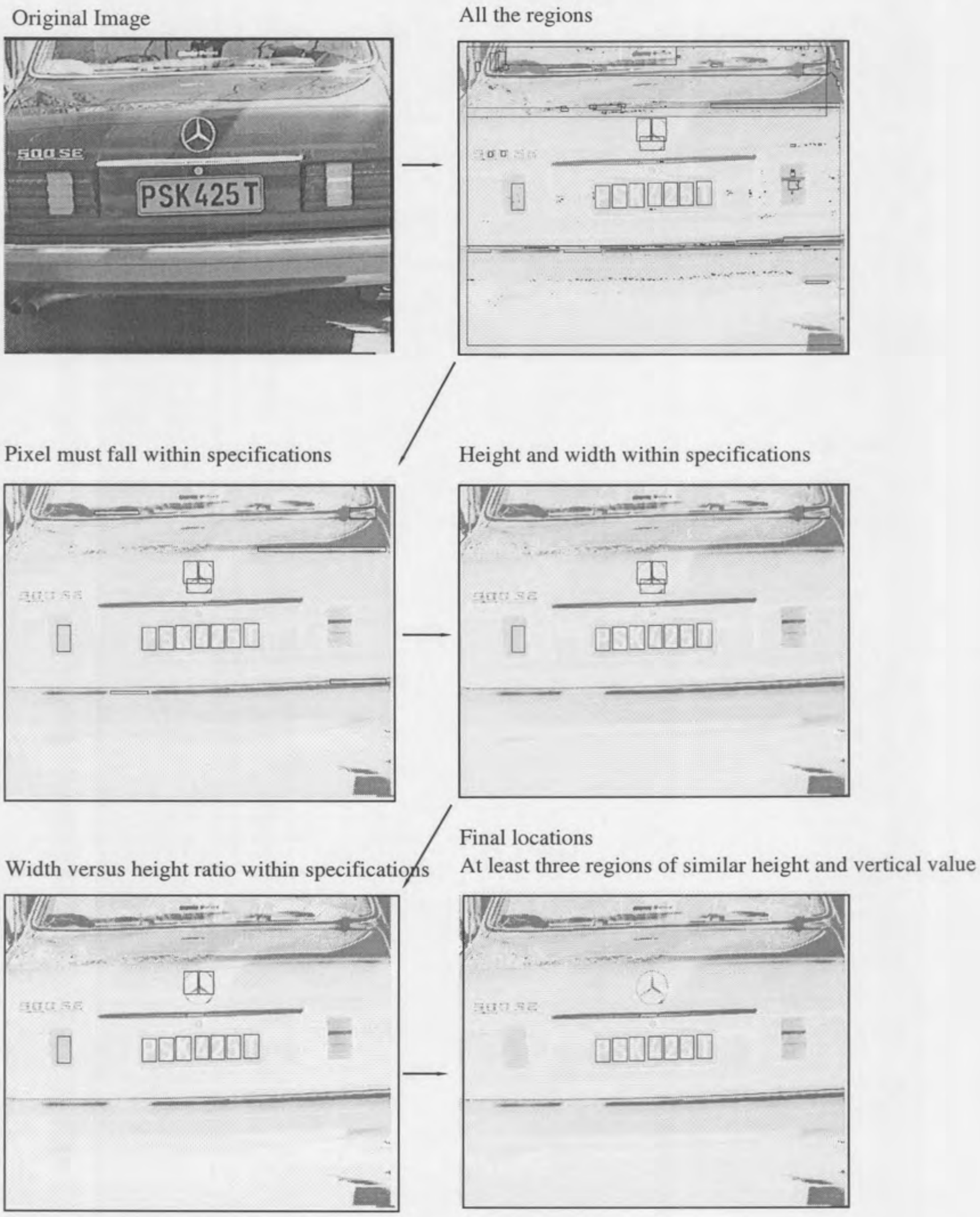

Original Image
All the regions
Pixel must fall within specifications
Height and width within specifications
Width versus height ratio within specifications
Final locations
At least three regions of similar height and vertical value
500 SE
PSK 425 T

Figure 3.3: The licence plate symbol location process

The new algorithm works as follows:

- take a horizontal cross-section through the image every n^{th} line of the image, where n must be less than the height of the smallest character expected,

- for each pixel in the current cross-section, if the difference in grey-level between the pixel and its neighborhood is greater than some threshold T, the pixel is marked as a vertical edge,
- cluster all vertical edge points into groups so that the horizontal distance between vertical edge points in the same point is less than the width of the largest character expected,
- merge clusters which are vertically adjacent into a single cluster.

An example of a number of clusters is shown in Figure 3.4, with the possible licence plate locations indicated with lines.

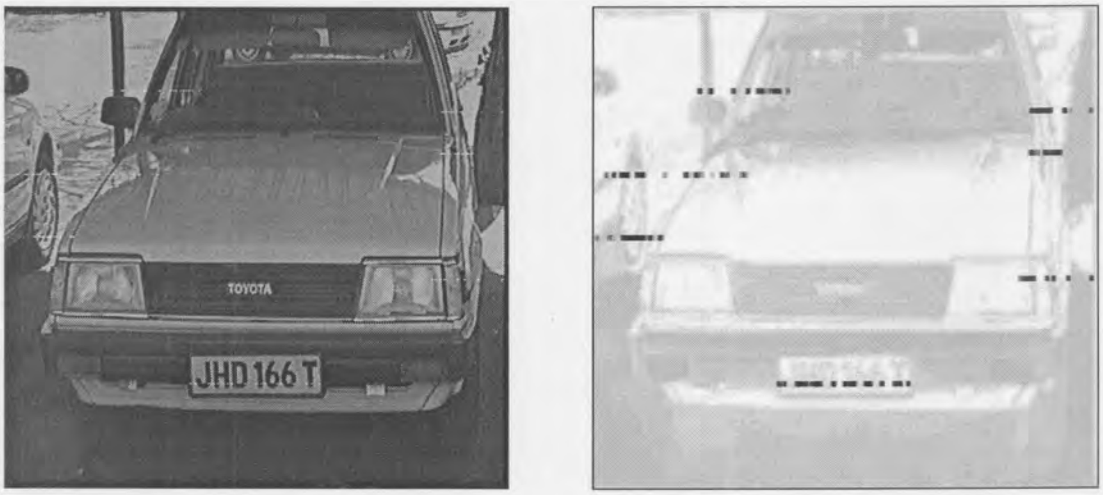


Figure 3.4: Possible licence plate locations

Executing this algorithm yields a number of clusters, where each cluster represents left and right extrema of a possible licence plate. The bottom and top and angle of the licence plate are found as follows:

- take the horizontal line between the left and right extremes of the possible licence plate and divide it into a number of intervals,

- for each interval find the estimate for the position of the top and bottom of the plate as follows:
    - place a horizontal bar over the current interval of the horizontal line and count how many vertical edges there are in the image through the bar,
    - slide the bar upwards, one pixel at a time and count the number of edges at each point,
    - when the bar passes over the number plate symbol, the number of edges will be zero,
    - the bar is slid no further than the largest character expected,
    - repeat for the bottom part of the licence plate symbols.

In Figure 3.5 and Figure 3.6 the possible licence plate locations are shown with black lines, while the top and bottom edges of the licence plates are shown with grey lines. The grey lines indicate where the licence plate symbols end. Only at a real licence plate will these end points form two lines as shown in Figure 3.6.

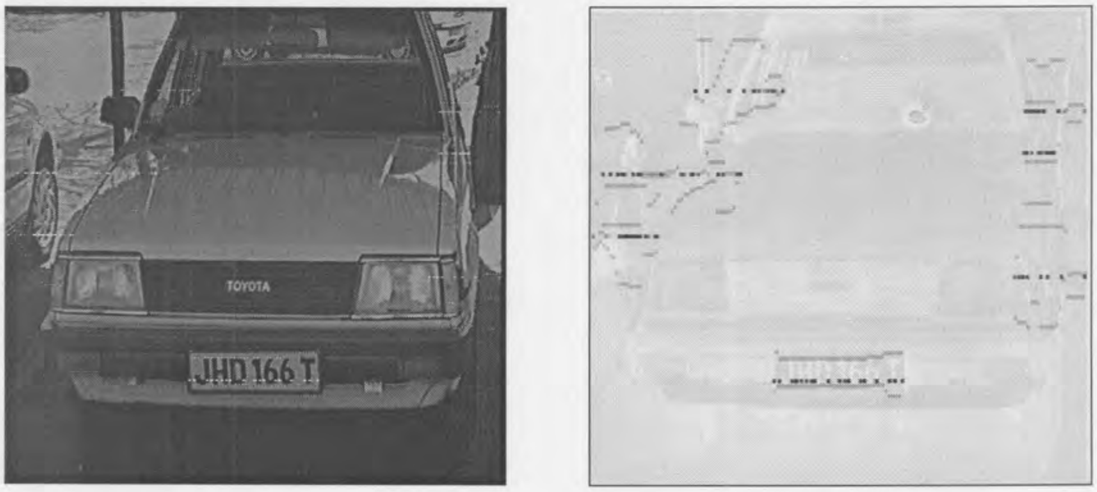


Figure 3.5: The location of possible licence plate borders

Clusters which do not represent licence plates can be removed as follows:

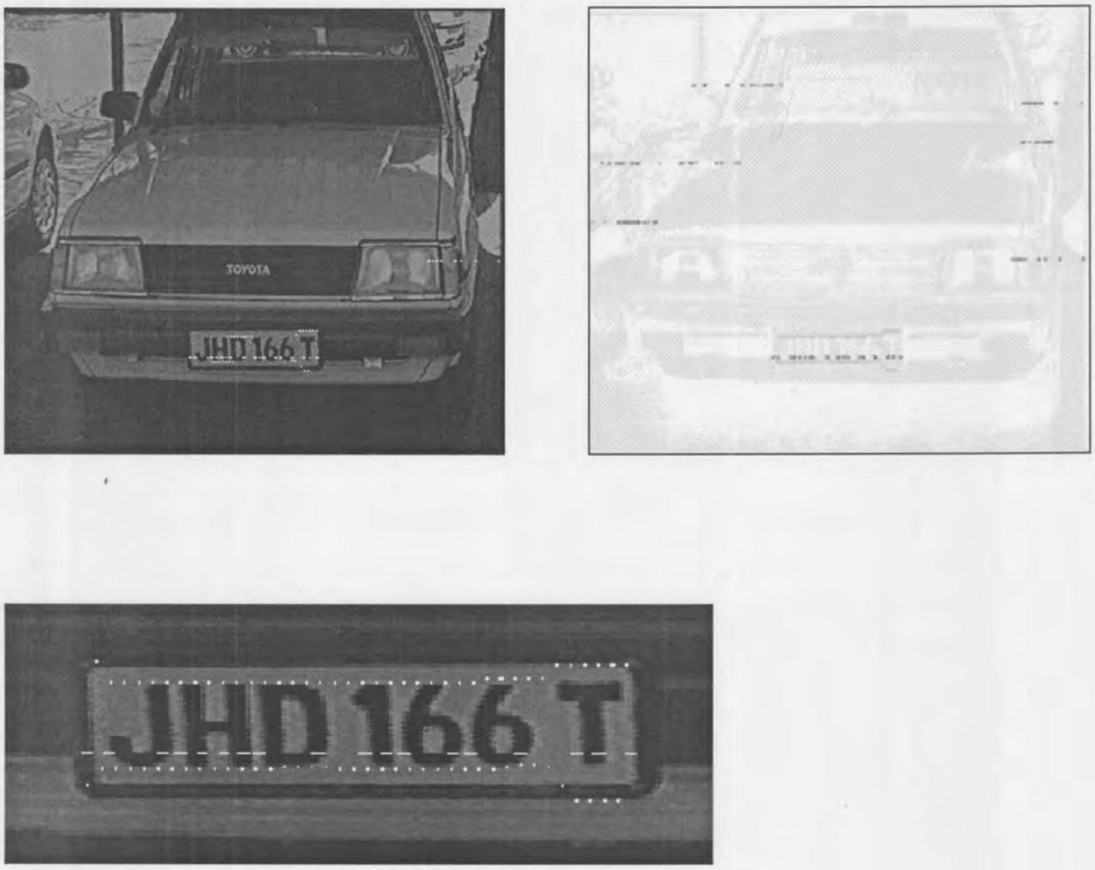


Figure 3.6: The licence plate borders of Figure 3.5

- more than half of the top symbols borders are in a row,
- more than half of the bottom symbols borders are in a row,
- the angles of the top and bottom row are the same and within certain limits,
- the vertical height of the symbol(distance between top and bottom border) is within certain limits.

The left and right borders detected by the above method will normally be too wide. The edge between the licence plate and the rest of the car normally forms the left and right border as seen in Figure 3.6. This algoritm thus produces the region where licence plate symbols are.

If the licence plate location algorithms are used on an image which does not contain a vehicle licence plate, both algorithms will exit because it will not find a region which comply to all the rules.

3.4 Symbol segmentation

After the licence plate region has been identified, each symbol has to be separated from its surrounding symbols.

The following techniques were used to identify possible symbol regions

- uniquely identified regions, obtained with the region growing algorithm,
- using the vertical projection of the licence plate to separate the symbols.

3.4.1 Region growing separation of symbols

For good quality images, the separation can be achieved by only using a region growing algorithm [23, 63]. Each region is identified as a unique symbol. Two kinds of errors frequently occur: More than one symbol is grouped as one or one symbol is split into more than one region. These errors occur when the licence plate is dark or too brightly illuminated. By adapting the threshold offset [63], dark and lightly lit licence plates symbols can be separated.

As shown Figure 3.7, a single threshold offset was not able to fully separate each licence plate into its symbols. The unique regions/blobs were given different colours for illustrative purposes only. By iteratively cycling through a number of threshold offset values, all of the symbols were separated.

In the final system, a two stage approach was used to find the location of the licence

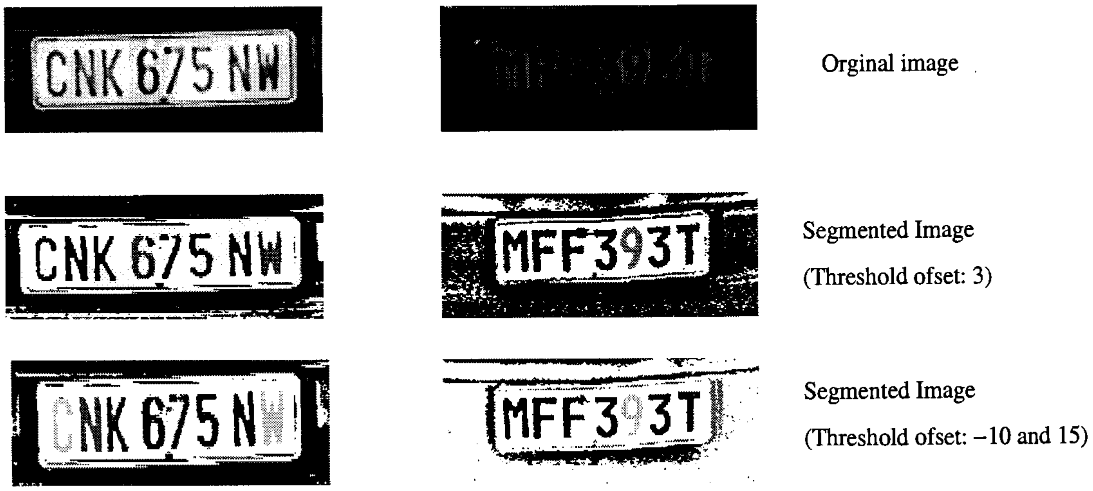


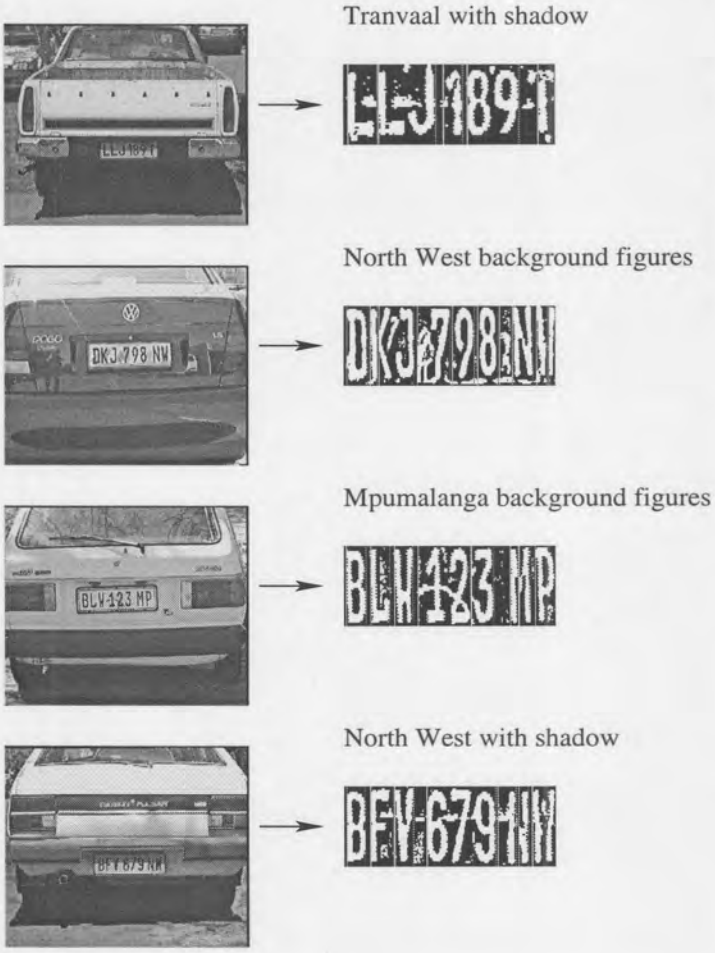
Figure 3.7: Dark and brightly lit licence plates which have been separated with different threshold offsets

plate. The first(rough) approach needed only to identify the licence plate region. Due to the possibility that some of the symbols were not separated from the background extra horizontal and vertical space is allocated for.

The second stage finds the exact borders of the licence plate symbols. The smaller image identified by the first stage is used in the second stage to identify the exact position of the licence plate symbols. A comparison between a two and a single stage recognition system will be discussed in Section 4.2.

3.4.2 Vertical projection separation of symbols

Some of the new licence plate (Mpumalanga , North West, Eastern Cape and Northern Cape Provinces) symbols can not be separated only with region growing and variable threshold offsets, as the licence plate contain dark background figures which will result in multiple symbols segmented as single entities. Physical objects clinging to the plate or damage can also result in multiple symbols segmented as single entities. Examples of licence plates symbols which can not be separated are shown in Figure 3.8.



Tranvaal with shadow

North West background figures

Mpumalanga background figures

North West with shadow

Figure 3.8: Example of licence plate images which can not be separated with binarisation

All of the symbol overflow in Figure 3.8 occur in the horizontal spaces between the symbols. Overflow between licence plate symbols and regions above and below the licence plate is rare, and can mostly be solved with variable threshold offsets.

Thinning algorithms such as morphological erosion [29, 82, 83] can separate symbols joined by overflow, but thin symbols have the problem of being falsely separated. False separation with a thinning algorithm is shown in Figure 3.9. The 05 symbols were correctly separated by the erosion function, but the *M* symbol was split incorrectly into two separate symbols.

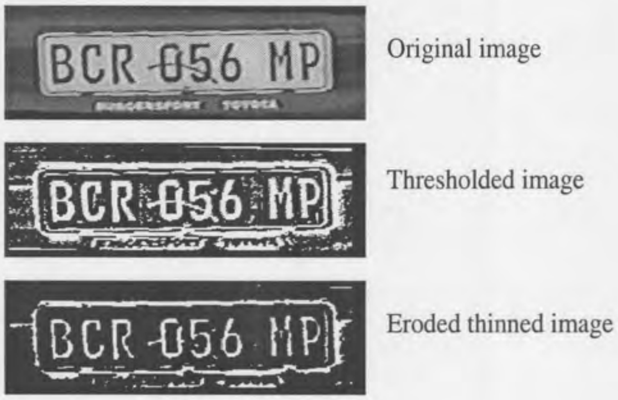


Figure 3.9: False separation of symbols achieved with a thinning algorithm

The horizontal overflow between the symbols can not be solved with variable threshold offsets. The overflow structure is usually only a few pixels high, fewer than the height of the symbols. If the overflow height is small, the symbols can be separated by using the vertical projection of the licence plate. Figure 3.10 displays the vertical projection of the licence plate images of Figure 3.8.

The licence plate symbols can be separated from each other using the vertical projection as follows:

- each symbol lies above a threshold on the vertical projection (see Figure 3.12),
- each symbol lies between two major valleys of the projection,
- each symbol lies between a positive and a negative derative of the vertical projection,
- combination of above techniques.

Noise on the licence plate symbols can lead to false symbol borders. The noise is inherent in the picture or is generated from the binarisation step. By using a simple digital low-pass filter the noise in the vertical projection is reduced. The following

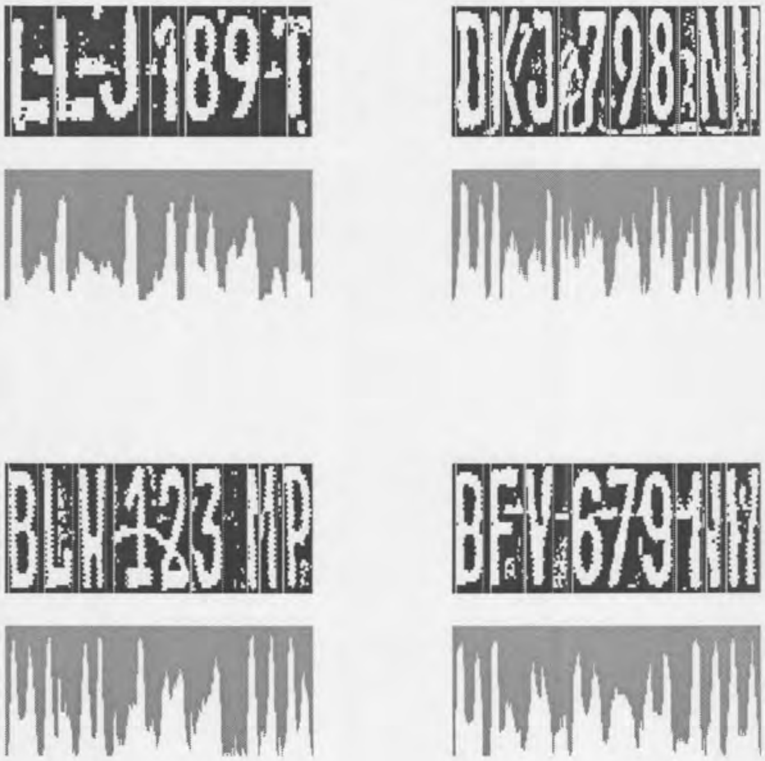


Figure 3.10: Vertical projection of licence plate images of Figure 3.8

averaging (low-pass) filter is used:

$$X_i = (X_{i-2} + X_{i-1} + X_i + X_{i+1} + X_{i+2}) * 0.2 \tag{3.1}$$

In Figure 3.11 an example of symbol separation applying the above rules on the vertical projection is shown. The white vertical lines show the start position of a symbol and the black vertical line show the end position of a symbol.

With or without the filter it may not always be possible to separate all the symbols without splitting some of the symbols. Extremely thin symbols can be implicitly removed as noise. Some licence plate symbols can never be separated because the overflowing/merged structures are thicker than one or more of the other characters. In Figure 3.8 the *BLW*123*MP* licence plate is such an example. The *M* and *W* symbols will always split be into two symbols each before the 1 and 2 symbols are separated.

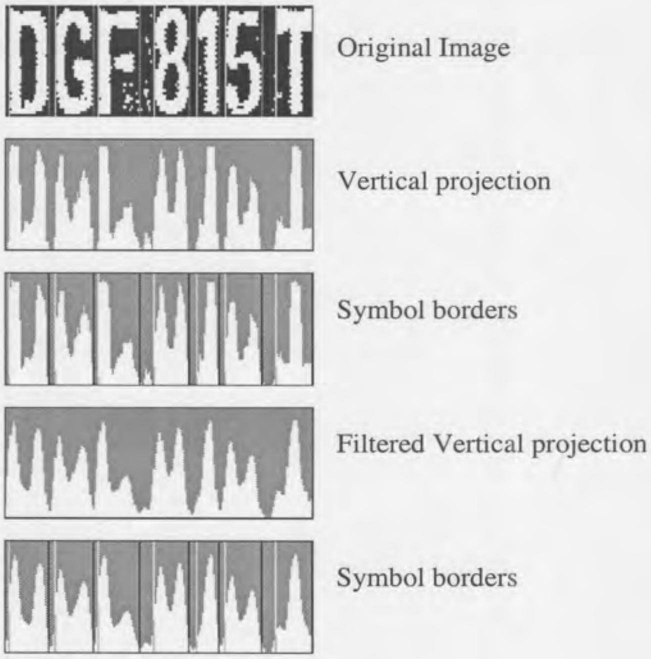


Figure 3.11: An example used for symbol separation with vertical projection

By over-segmenting [84] [14], and combining, licence plate symbols as seen in Figure 3.8 can be successfully segmented. For the majority of the over-segmentation cases it can be assumed that a symbol is split into one or two segments. These segments are called frames in Section 3.5. Extremely thin frames are removed and assumed to have originated from noise.

It is assumed that each symbol is directly followed by the next symbol. The province logo of Gauteng licence plates is removed by only selecting the symbols with a vertical projection above a threshold with the vertical projection. An example of a Gauteng licence plate is shown in Figure 3.12.

Over-segmented images are used with the constrained Viterbi symbol alignment algorithm described in Section 2.3.4 to search for the most probable licence plate symbol sequence. This technique is discussed further in Section 3.5.

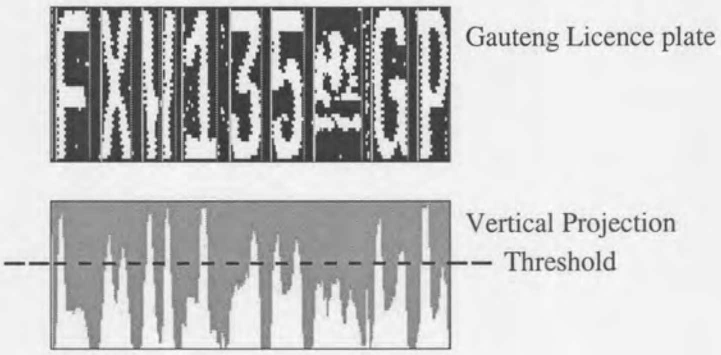


Figure 3.12: Gauteng licence plate with vertical projection

3.5 HMM-MLP symbol segmentation and classification

To explain the HMM-MLP segmentation and classification algorithm, a step by step evaluation of two examples is given.

The first example evaluates the Viterbi algorithm with a simplified HMM. The HMM used can only deal with old Transvaal, Gauteng, North West and Mpumalanga licence plates. All the legal transition values are 1 (a_{ij}), with the rest set to 0. The legal transitions are shown in Figure 3.13.

In Figure 3.13 the legal observations are $\alpha, \eta, T, N, W, G, P, M$. These symbols represent the following licence plate symbols:

- $\alpha = [A, B, C, D, E, F, G, H, J, K, L, M, N, P, R, S, T, V, W, X, Y, Z]$
- $\eta = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]$
- $T = [T]$
- $N = [N]$
- $W = [W]$

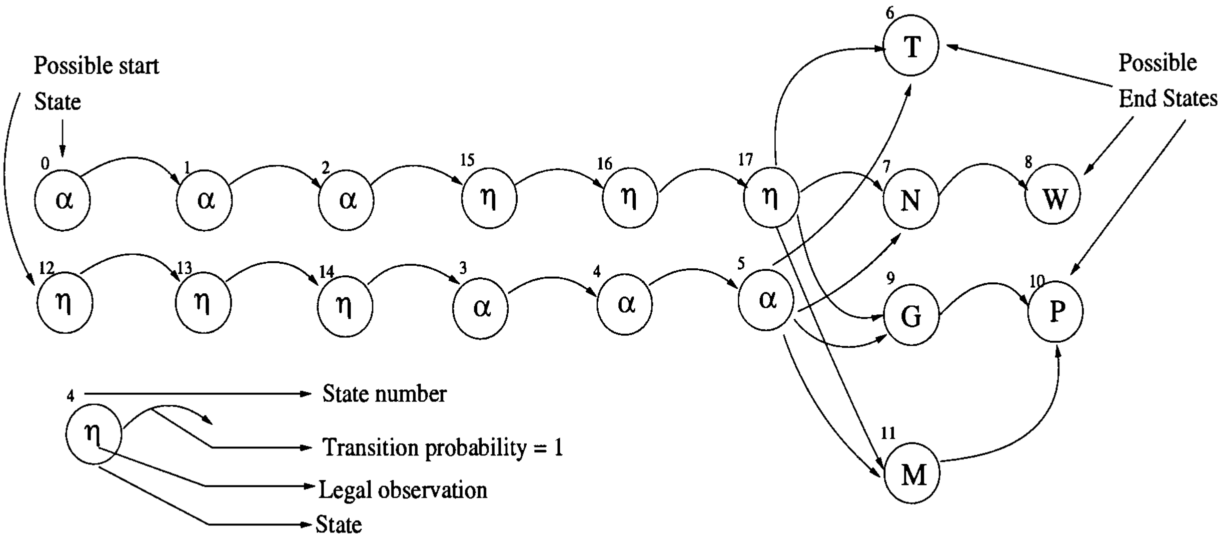


Figure 3.13: The hidden Markov model used by the Viterbi algorithm of Figure 3.14 and Figure 3.15.

- $G = [G]$
- $P = [P]$
- $M = [M]$

All the transition probabilities are equal to 0 or 1. Only valid transitions have a value of 1, thus ensuring that only a legal sequence is followed. A possible optimisation of the HMM is to include a new class similar to α but without the vowels. The advantage gained by the recognition, may be negated by the increased complexity of the HMM.

To illustrate the HMM-NN hybrid, the following example is shown in Figure 3.14.

The first frame symbol (B), probability output vector as calculated by the NN is shown in the top part of Figure 3.14. Next to the probability output vector the observations maximums of that (B) symbol is shown. In Figure 3.13 only two legal observations exist for the first symbol, the α and η observations.

In the bottom part of Figure 3.14 for each symbol, the legal observation probabilities

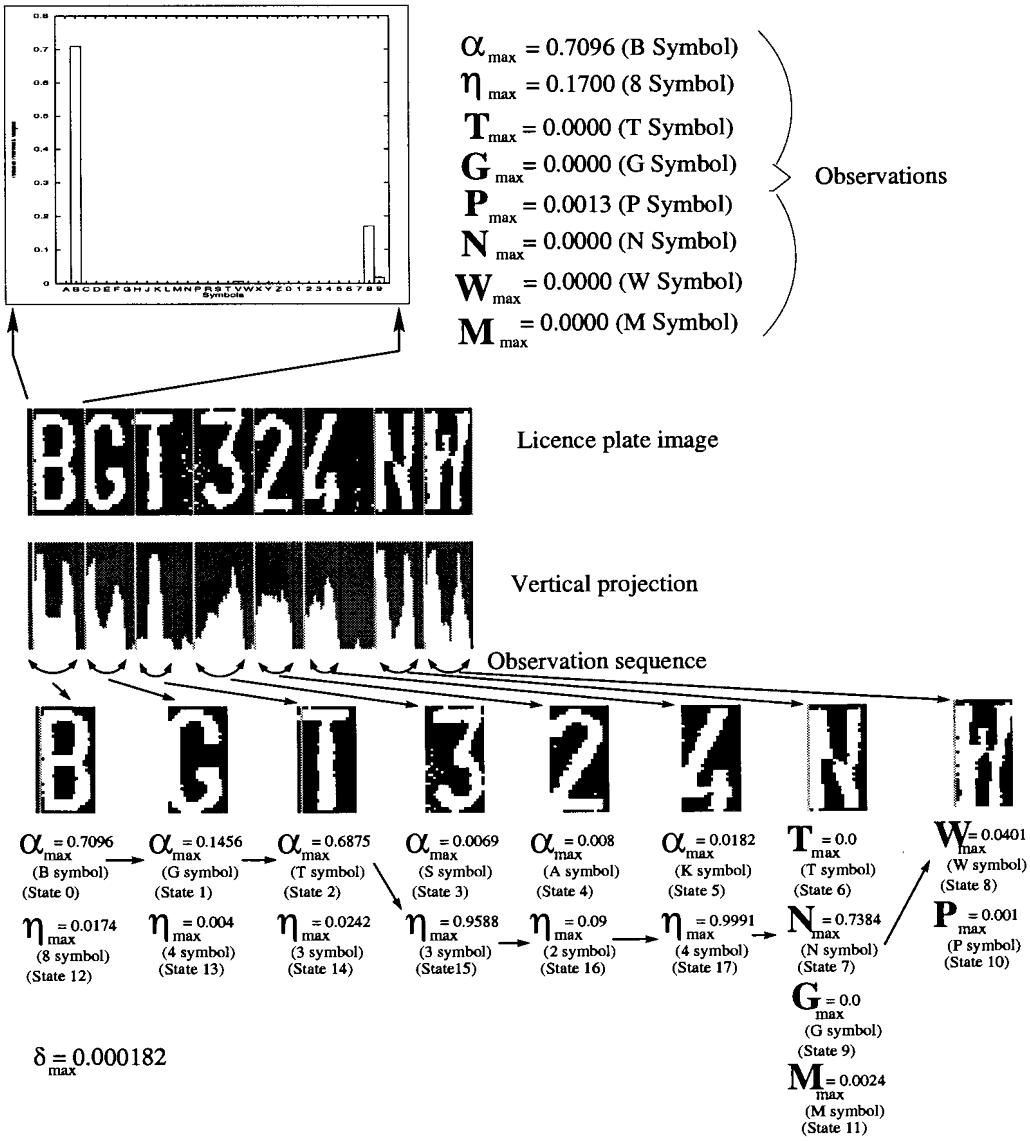

α max = 0.7096 (B Symbol)
η max = 0.1700 (8 Symbol)
T max = 0.0000 (T Symbol)
G max = 0.0000 (G Symbol)
P max = 0.0013 (P Symbol)
N max = 0.0000 (N Symbol)
W max = 0.0000 (W Symbol)
M max = 0.0000 (M Symbol)
Observations
Symbols
Licence plate image
Vertical projection
Observation sequence
α max = 0.7096 (B symbol) (State 0)
α max = 0.1456 (G symbol) (State 1)
α max = 0.6875 (T symbol) (State 2)
α max = 0.0069 (S symbol) (State 3)
α max = 0.008 (A symbol) (State 4)
α max = 0.0182 (K symbol) (State 5)
T max = 0.0 (T symbol) (State 6)
W max = 0.0401 (W symbol) (State 8)
η max = 0.0174 (8 symbol) (State 12)
η max = 0.004 (4 symbol) (State 13)
η max = 0.0242 (3 symbol) (State 14)
η max = 0.9588 (3 symbol) (State15)
η max = 0.09 (2 symbol) (State 16)
η max = 0.9991 (4 symbol) (State 17)
N max = 0.7384 (N symbol) (State 7)
P max = 0.001 (P symbol) (State 10)
G max = 0.0 (G symbol) (State 9)
M max = 0.0024 (M symbol) (State 11)
δ max = 0.000182

Figure 3.14: Hidden Markov model neural network hybrid example

are shown. Each symbol represents a state as shown in Figure 3.13. The first 6 symbols' legal observations are only α or η. The (N) symbol is the seventh symbol and thus its state may be: T, N, G or M. The last symbol (W) only observations are W or P. The values of the observations are calculated by the NN. Each symbol which represents a state was passed to the NN, and out of the NN output values (also known as the probability vector) each possible observation maximum was calculated. The path which leads to the maximum value is shown in the bottom of Figure 3.14. For the licence plate image of Figure 3.14, $BGT324NW$ is the most likely legal sequence. The most likely sequence is calculated with the Viterbi search algorithm, described in Section 2.3.3.

To illustrate the over-segmentation technique, another example is shown in Figure 3.15. Two possible legal sequences are shown. The first sequence ($BGN868MP$) has a score of 0.0535, compared to the second sequence ($BNG587GP$) with a score of 0.0000174.

The number of possible sequences is much larger with over-segmentation. The NN uses features from one and two frames as seen in Figure 3.12. The sequence of the states must still be linear, thus no overlapping or exclusion of frames.

The Viterbi search algorithm can produce false results if the over-segmented or under-segmented frames have high output symbol probabilities ($\alpha_{max} = 0.46$ and $\eta_{max} = 0.63$). In the example in Figure 3.15 the 86 and the first half of the M symbols have high output symbol probabilities. If the second part of the M symbol output symbol probability was higher, a false sequence could have been the most probable and the system will fail to correctly recognise the licence plate.

The ability of the Viterbi search algorithm to assign a score to a specific sequence, enables the system developer to change any parameter or symbol segmentation technique, and then compare the result with the previous highest score.

In conclusion, multiple methods with multiple small optimisations to recognise a licence plate are possible. These methods do not achieve the success with all possible licence plate images. By giving a score for each method the best result can be obtained by using multiple recognition methods.

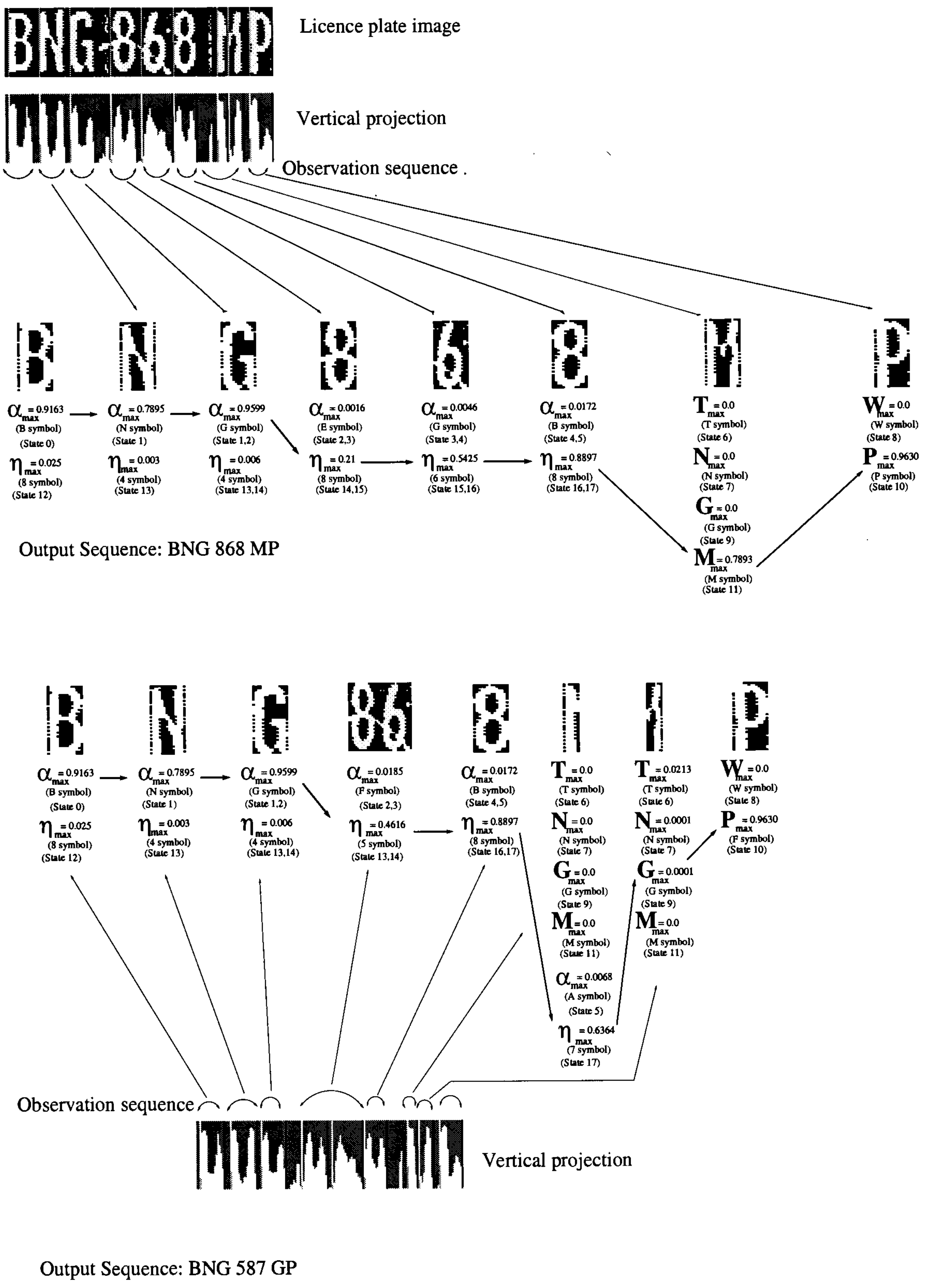


Figure 3.15: Hidden Markov model neural network hybrid examples for over-segmentation

Chapter 4

Experiments

4.1 Experimental procedure

The experiments were performed off-line on the images captured with standard commercial video and security cameras. The video camera stored the images to tape, which was digitised using a Matrox Meteor card. The data set includes slow moving and stationary vehicles. The digitised images were processed on a Intel-based PC running Linux.

The range of lighting conditions varied between sunshine, over-cast to light rain. Special attention was given to finding examples of damaged, darkened, lightly lit or dirty licence plates. [1]

Colour and high definition images were converted to 8 bit grey-scaled images. The resolution ranged between 512 × 480 pixels to 768 × 576 pixels. The size of the licence plate symbols differed from a maximum of from 35 × 48 pixels to a minimum of 6 × 31 pixels.

[1] The images are available at: http://eerc.up.ac.za/~renier/M

All experiments were performed on a test data set of 318 images. These images were not used for any neural network training.

4.2 Optimisation of licence plate segmentation

The goal of the following experiments was to optimise all the variables for the location of the licence plate.

The location of the licence plate entailed a two-step-approach. First a rough area in which the licence plate could be, was obtained (licence plate segmentation). The two schemes discussed in Section 3.3 were tested. The second step identified the precise location of the licence plate (symbols segmentation).

For the first step many variables, as subsequently described, were empirically tested, to find the best value. Although dependencies may exist, the optimisation problem will become too large if all possible solutions were tested for.

Experiment 1: Optimisation of licence plate segmentation iteration

The optimal value of the following variables were obtained experimentally:

- size of local threshold blocks,
- local threshold offset (See ϕ of equation 2.2),
- the horizontal extra space for un-segmented symbols,
- the vertical extra space for un-segmented symbols.

All the system parameters are kept constant, except the variable being tested. In Figure 4.1 to Figure 4.4 the results of the experiments are shown. Each variable seems to have a minimum error area. These minima were used for the rest of the experiments. The experiments were performed on the test data set of 318 images.

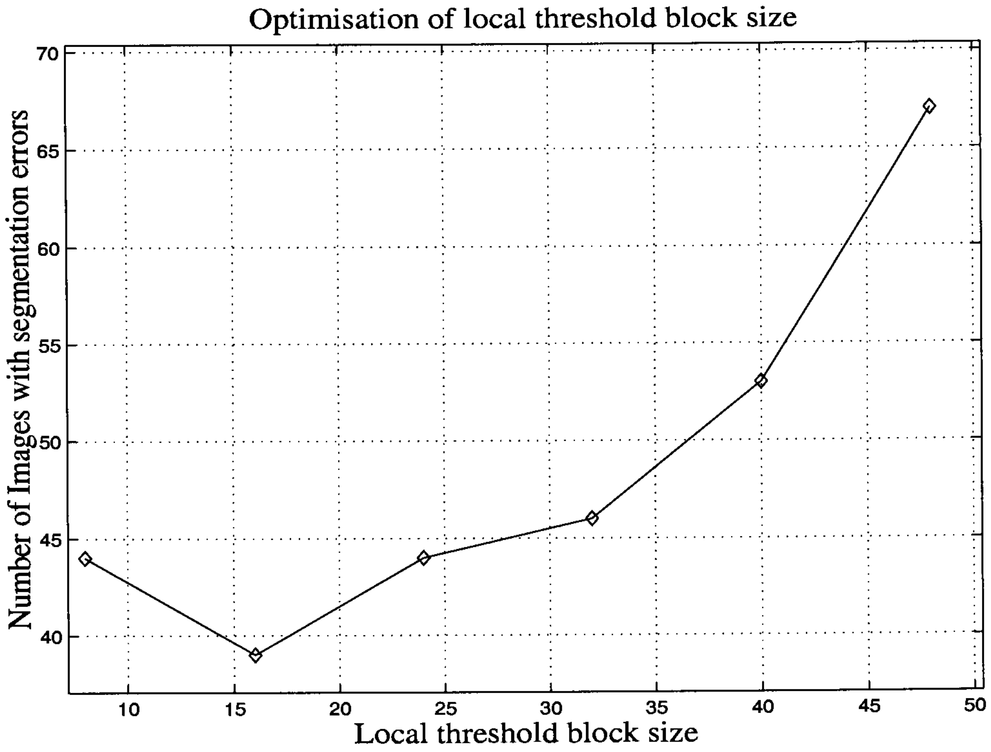


Figure 4.1: Optimisation of local threshold block size

As seen in Figure 4.2, the number of segmentation errors for a zero offset is much higher than the rest. With a zero offset a high level of noise is introduced, which increases the segmentation errors.

An iterative approach which used all the different threshold offset values and by using the score generated by the Viterbi search algorithm, achieved a much higher success rate than any specific offset value (see Table 4.1). For each threshold offset value, a possible licence plate sequence and score was generated. The sequence with the best score was kept.

The following segmentation optimisation variables were not obtained through experimentation, but chosen explicitly:

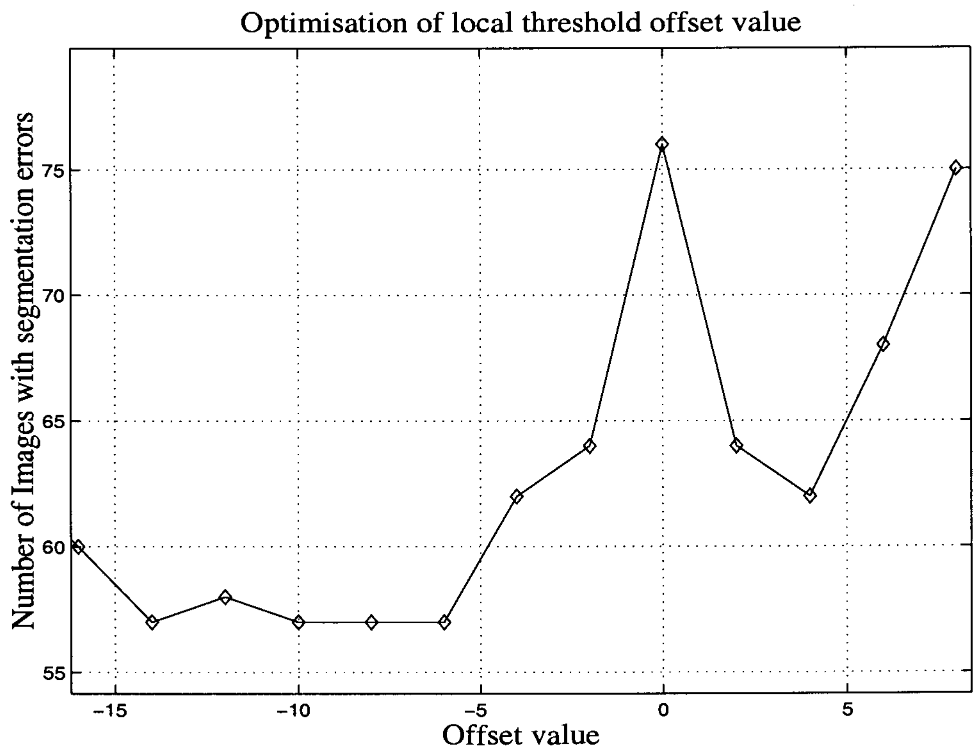


Figure 4.2: Optimisation of local threshold offset value

Threshold offset	Segmentation Errors
-8 (Best threshold offset)	57
Combination all threshold offsets	22

Table 4.1: The advantage of an iterative thresholding approach

- minimum distance between symbols: 5 pixels,
- maximum distance between symbols: 100 pixels,
- minimum pixel size of symbols: 200 pixels,
- maximum pixel size of symbols: 900 pixels,
- minimum vertical size of a symbol: 20 pixels,
- maximum vertical size of a symbol: 150 pixels,

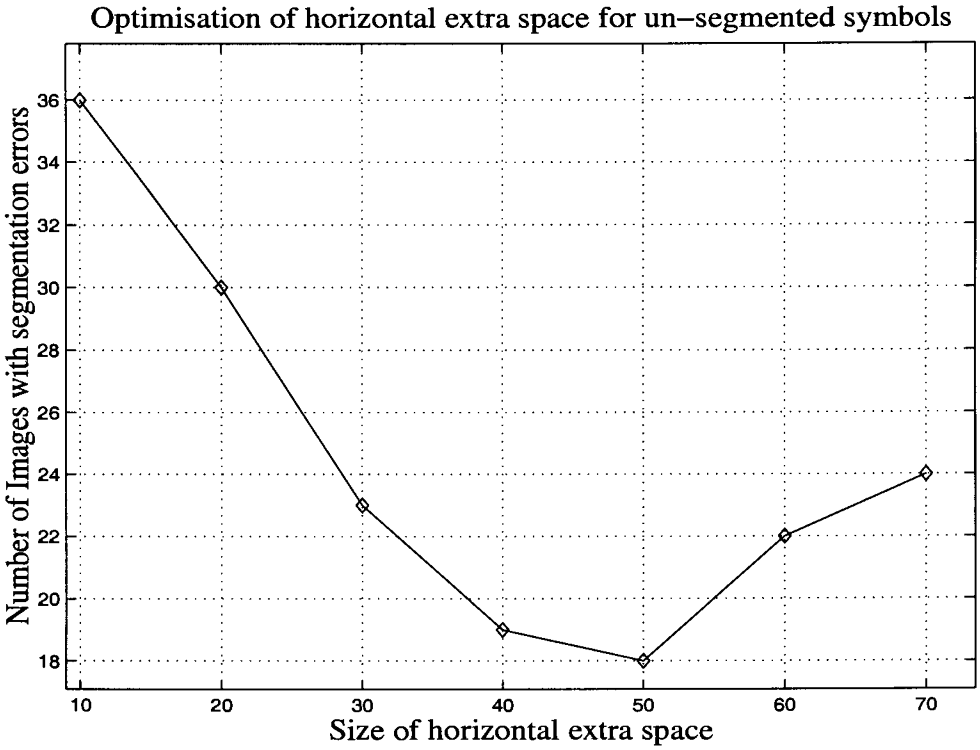


Figure 4.3: Optimisation of horizontal extra space for un-segmented symbols

- minimum symbol vertical to horizontal ratio: 1.2,
- the minimum number of symbols in a line: 2 .

Parameter	Value
Local theshold block size	16 pixels
Local theshold offset value	-8
Horizontal extra space for un-segmented symbols	50 pixels
Verical extra space for un-segmented symbols	40 pixels

Table 4.2: Summary of optimal values obtained with experiment 1

The physical properties of the licence plate images constrained the variables mentioned above. Obtaining optimal values experimentally would be meaningless, because the aim

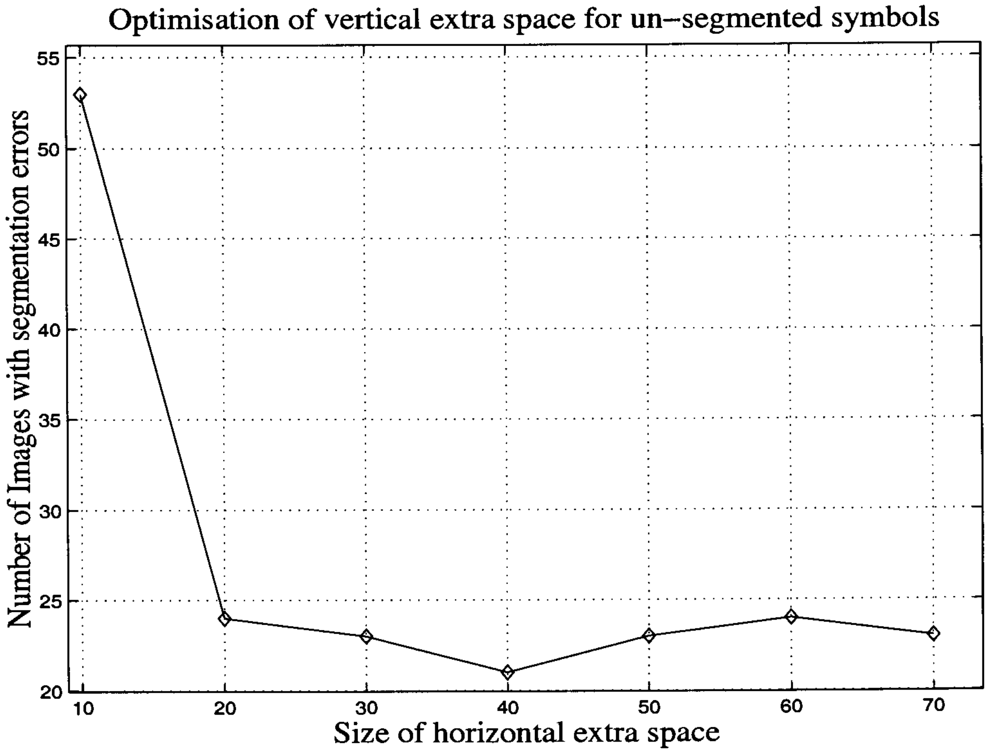


Figure 4.4: Optimisation of horizontal extra space for un-segmeneted symbols

was to recognise all the licence plates in the test set. The optimal values obtained with the experiments are shown in Table 4.2.

Experiment 2: Evaluating the effect of a second stage segmentation

We found that by using a two stage segmentation, a performance drop of 3.77 % in locating the licence plate occurred, but the second stage induced a 33 % decrease in computation time.

The new algorithm based on vertical edges (Section 3.3) was implemented without the optimisation of the variables associated with it. Out of the 318 images, 120 segmentation errors occurred compared to the 22 segmentation errors with the normal search procedure. By combining the two search algorithms the number of segmentation errors were reduced to 19. The comparison between licence plate segmentation techniques is shown in Table 4.3.

Licence plate Segmentation Technique	**Percentage segmentation errors**
Region growing	6.91 %
Vertical edges	37.8 %
Combination of both techniques	5.97%

Table 4.3: Comparison between licence plate segmentation techniques

4.3 Symbol segmentation

Experiment 3: Comparison of symbol segmentation techniques

The goal of this experiment was to compare the success rate of different symbol segmentation techniques. All the techniques used the same symbol segmentation and neural network parameters. The following techniques were compared:

- Simple segmentation with single frame Viterbi,
- Threshold on the vertical projection with single frame Viterbi,
- Threshold on the vertical projection with filtered data and single frame Viterbi,
- Threshold on the vertical projection with double frame Viterbi,
- Symbols lies between two major valleys of the projection with filtered data and double frame Viterbi.
- Symbols lies between two major valleys of the projection and with single frame Viterbi

- Symbols lies between two major valleys of the projection and with filtered data and single frame Viterbi

Experiment 3 results are shown in Table 4.4.

Segmentation Technique	**Percentage Recognised**	**Average Score**
Simple segmentation	82.70 %	0.323
Vertical projection with single frame Viterbi	70.56 %	0.241
Vertical projection with filtered data and single frame Viterbi	71.01 %	0.169
Vertical projection with double frame Viterbi	74.21 %	0.273
Symbols between valleys with filtered data and double frame Viterbi	28.80 %	0.137
Symbols between valleys and single frame Viterbi	0 %	0.0
Symbols between valleys with filtered data and single frame Viterbi	1.26 %	0.0

Table 4.4: Comparison between symbol segmentation techniques

The "Vertical projection" and "Symbols between valleys" algorithms are discussed in Appendix C and Appendix D respectively.

From Table 4.4 the "Symbol between valleys technique with single frame Viterbi" fails completely, for example symbols such as "X" and "H" are split into two symbols. By using double frame Viterbi, where one or two frames constitute a symbol, the "Symbol between valleys technique" achieves a reasonable result.

The best results were achieved with simple segmentation. The extra complexity of the other methods do more harm than good, it fails more than it improves. Filtering the vertical projection data only slightly improves the result.

Each recognition technique score the probability of success. These scores are given in the third column of Table 4.4. By using the scores, all the different techniques can be used and the highest score answer as the final result. Table 4.5 illustrates the results if a combination of the symbol segmentation techniques are used.

Segmentation Technique	**Percentage Recognised**	**Average Score**
Simple segmentation	82.70 %	0.323
Combination of all techniques	83.65 %	0.375
Combination of all techniques except the double Viterbi techniques	86.48 %	0.353

Table 4.5: Comparison between combining symbol segmentation techniques

By combining the symbol segmentation techniques, the average score and the recognition rate improves. The techniques which use double frame Viterbi unfortunately fail to recognise some licence plates while still giving its attempt a high score. Thus the highest recognition is achieved by a combination of techniques excluding the double Viterbi methods. The highest average score thus do not map to the highest recognition rate. More detail of the recognition rate and scores can be seen in Appendix E.

4.4 MLP optical character recognition

4.4.1 Neural Network training and optimisation

To find the global minimum for all the possible NN variables would have been too cumbersome and time consuming. Only local minima for each variable was sought. The final system was implemented with variable values attained by each experiment. This may not represent the global minimum.

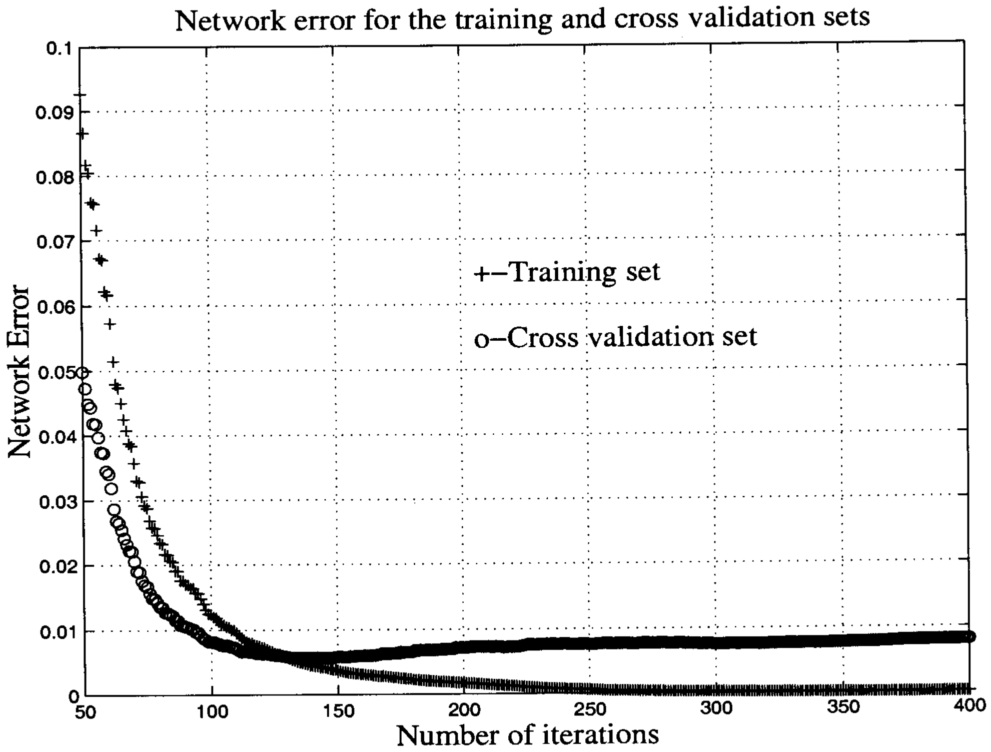


Figure 4.5: Network error for the training and cross validation set against number of iterations

Experiment 6: Optimisation of neural network features

The scaled mesh function size was empirically obtained, by selecting the dimensions with the smallest error with the cross validation set. The dimensions size of 7×12 gave the lowest network error. The details of the experiments to obtain the dimensions can be found in Appendix F.

Experiment 7: Optimisation of hidden neuron number

The optimal number of hidden neurons was 25, empirically obtained for the smallest cross-validation error. The optimal number of hidden neurons is shown in Figure 4.7.

The influence of the number of hidden neurons only has a significant influence if the

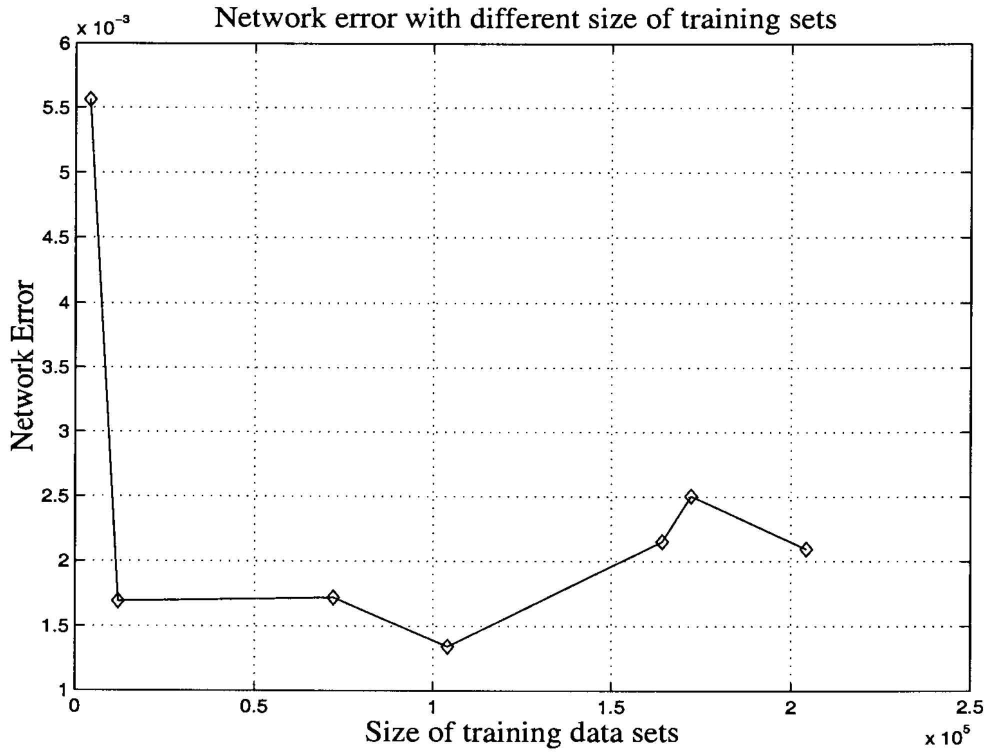


Figure 4.6: Network error for the cross validation set against the size of the training data set

number of hidden neurons is very small. For large numbers the error is only slightly higher than the optimal. De Beer and Botha [28] used 25 hidden neurons.

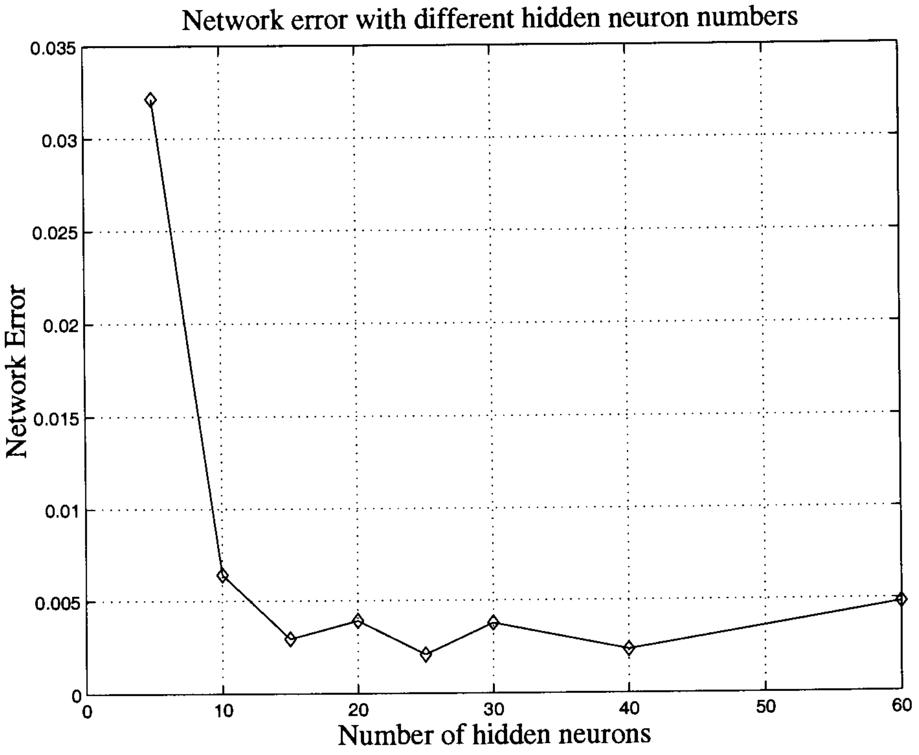


Figure 4.7: Network error for different hidden neuron numbers

Chapter 5

Summary and conclusion

This dissertation presented a study of robust recognition of vehicle license plates utilising hidden Markov models. A system was developed which requires a digital image as input, to produce a licence plate symbol sequence. The system consists of an image capture, image segmentation, licence plate location, symbol segmentation and HMM-MLP symbol segmentation and classification components. In the development of the system, a novel implementation of the Viterbi algorithm constrained for our purposes was implemented. The Viterbi algorithm scored the recognition attempt. This enabled us to use multiple threshold values and using the answer with the highest score. We were also able to combine different licence and symbol segmentation techniques.

5.1 Summary of results

In Experiment 1 we optimised the variables used in the licence plate segmentation process. In Experiment 2 we evaluated the effect of a second stage in the licence plate segmentation process. In Experiment 3 we compared the symbol segmentation techniques. In Experiment 4 to Experiment 7 we optimised the neural network.

Because we could score any attempt to recognise a licence plate, we were able to use multiple threshold values for the licence plate and symbol segmentation process. As seen with Experiment 1 and 2, the number of segmentation errors was dramatically reduced by combining different techniques and using the best score.

The best symbol segmentation was achieved with simple segmentation, as seen in Experiment 2 (Table 4.4). The double frame Viterbi techniques gave false high scores to symbols which were split. This lowered their recognition rate and explained why the best results were achieved without the double frame Viterbi techniques (see Table 4.5).

The optimal neural network input mesh size was experimentally obtained as 7 × 12. The overall licence plate recognition results were more dependent on the segmentation of the symbols than the neural network outputs. With the dimensions (7 × 12), a 99.53% recognition rate was achieved (recognising symbols which have already been separated from the vehicle image).

In conclusion the optimal result achieved (the correct recognition of the licence plate sequence) was 93.7%. This was achieved with a test set of 318 images, i.e. 298 licence plates sequences were recognised correctly. The parameters used to obtain the optimal results and error analysis are shown in Appendix G, Table G.1 to Table G.6.

5.2 Limitations and more avenues to explore

The current system's execution time is too long for a real-world application. This can be solved by optimising the code and compromising recognition for speed by removing some of the subsystems.

The "vertical edge" licence plate segmentation technique developed by Scetchell[61] was not optimized. His symbol segmentation technique was not explored.

The influence and effect of using multiple fonts for the neural network and the use of multiple feature sets as inputs for the neural network can also be investigated in future work.

In the Markov model, only one value was used for an allowed transition. If more data is available, the transition values can be optimised according to the statistics of licence plate sequences.

Finally, in calculating the final score, post processing of the value received may enhance the result.

Appendix A

Commercial vehicle license plate systems

CARINA - Software Product for Automatic Number Plate Recognition - Budapest
http://www.arhungary.hu

VECON: Vehicle and Container Number Plate Recognition System - Hong Kong
http://www.asiavision.com.hk/

Vehicle License Plate Recognition System - IMPS (TM) - Optasia Systems Pty Ltd
http://www.singaporegateway.com/optasia/imps.htm

Computer system for car license plate recognition - Rossi-MegaCar - Savvinskaya nab. - Moscow
http://www.rossi.ru/mega/MEGA_E.HTM#Rossi_MegaCar

Automated Car Number Plate Recognition - FALCON - Livingston - EH54 6NG
http://www.ednet.co.uk/ euroquest/falcon.htm

License Plate Recognition (LPR) - Hi-Tech Solutions Israel *http://www.htsol.com/Products/SeeCar.html*

License Plate Reader - Perceptics Corporation - Tennessee - USA
http://www.perceptics.com/

RACAL'S LICENCE PLATE RECOGNITION SYSTEM - United Kingdom
http://home.racal.com/news/news39.htm

LICENSE PLATE CAPTURE SYSTEMS BY PEARPOINT - Engineered for High-Tech ITS Applications - USA
http://www.pearpoint.com/traffic_tech.html

Zamir Recognition Systems Ltd. - Manachat Technology - Park Building 1- Jerusalem - Israel
http://www.zamir.co.il/

CarFlow - MegaPixel Ltd. - Moscow - Russia
http://www.photocop.com/products.htm#MegaPixel

Jet ANPR Car Number Plate Recognition - Maple House - High Street - Potters Bar - United Kingdom
http://www.citysync.co.uk/pagedef.htm

AutoVu Technologies Inc. - Montreal - Quebec - Canada
http://www.autovu.com/index2.html

Appendix B

Recursive region growing algorithm

The algorithm used for the recursive region growing is as follows:

Main function

- Repeat for all pixels
  - If pixel is a foreground pixel
    - Create a new node in the linked-list
    - Call the recursive function

Recursive function

- Change pixel value to intermediate value
- If the pixel to the left is a foreground pixel call the Recursive function
- If the pixel to the right is a foreground pixel call the Recursive function
- If the pixel to the top is a foreground pixel call the Recursive function
- If the pixel to the bottom is a foreground pixel call the Recursive function

- Add pixel position, to the linked-list node

The linked-list consists of nodes. Each node contains the vertical and horizontal position of all the pixels of one foreground object.

Appendix C

Threshold on the vertical projection algorithm

The algorithm used for the segmentation of symbols by thresholding the vertical projection follows:

Main function:

Repeat for all the pixels of the vertical projection

- Remove all the values below a set vertical threshold.

Repeat for all the pixels of the vertical threshold:
Start from the left:

- If the vertical projection changes from zero to a positive value, mark the start position of a symbol,
- If the vertical projection changes from a positive value to zero, mark the end position of a symbol.

Repeat for all the symbols found with the vertical threshold

- Remove symbol positions if the average vertical projection value is below a set threshold, (d) and (e) in Figure C.1
- Remove symbol positions if the horizontal width of the symbol is below a set threshold (d) and (e) in Figure C.1.

The threshold on the vertical projection algorithm is shown in Figure C.1.

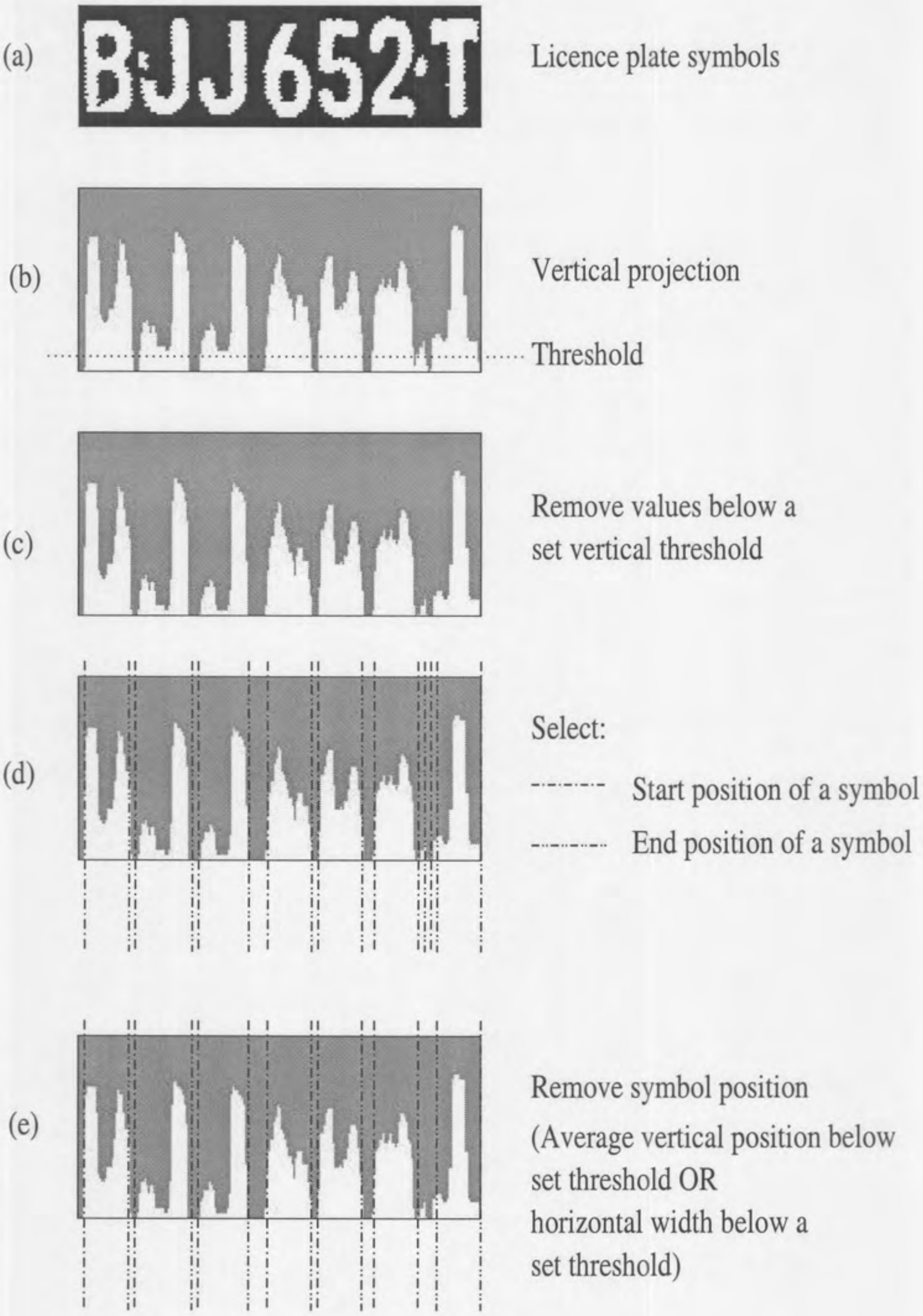


Figure C.1: Threshold on the vertical projection algorithm example

Appendix D

Valleys below a threshold algorithm

This algorithm segments the licence plate by looking for valleys between the vertical projection of the symbols: In order to prevent splitting of symbols, vertical projected values above a certain threshold are set to the maximum projection value.

The algorithm follows:

Repeat for all the pixels of the vertical projection

- Set all the values above a set vertical threshold to the maximum value, (c) in Figure D.1.

Repeat for all the pixels of the vertical projection:
Start from the left:

- If the vertical projection is at a valley, set the end position of a symbol and the start of the next symbol, (d) in Figure D.1 (At the projection value where the previous value is lower than the current value).

Repeat for all the symbols found (areas between a start and an end position)

Start from the left:

- Shrink each symbol horizontal position, to include only the symbol information, and not the zero values (e) in Figure D.1.

The valleys below a threshold algorithm is shown in Figure D.1.

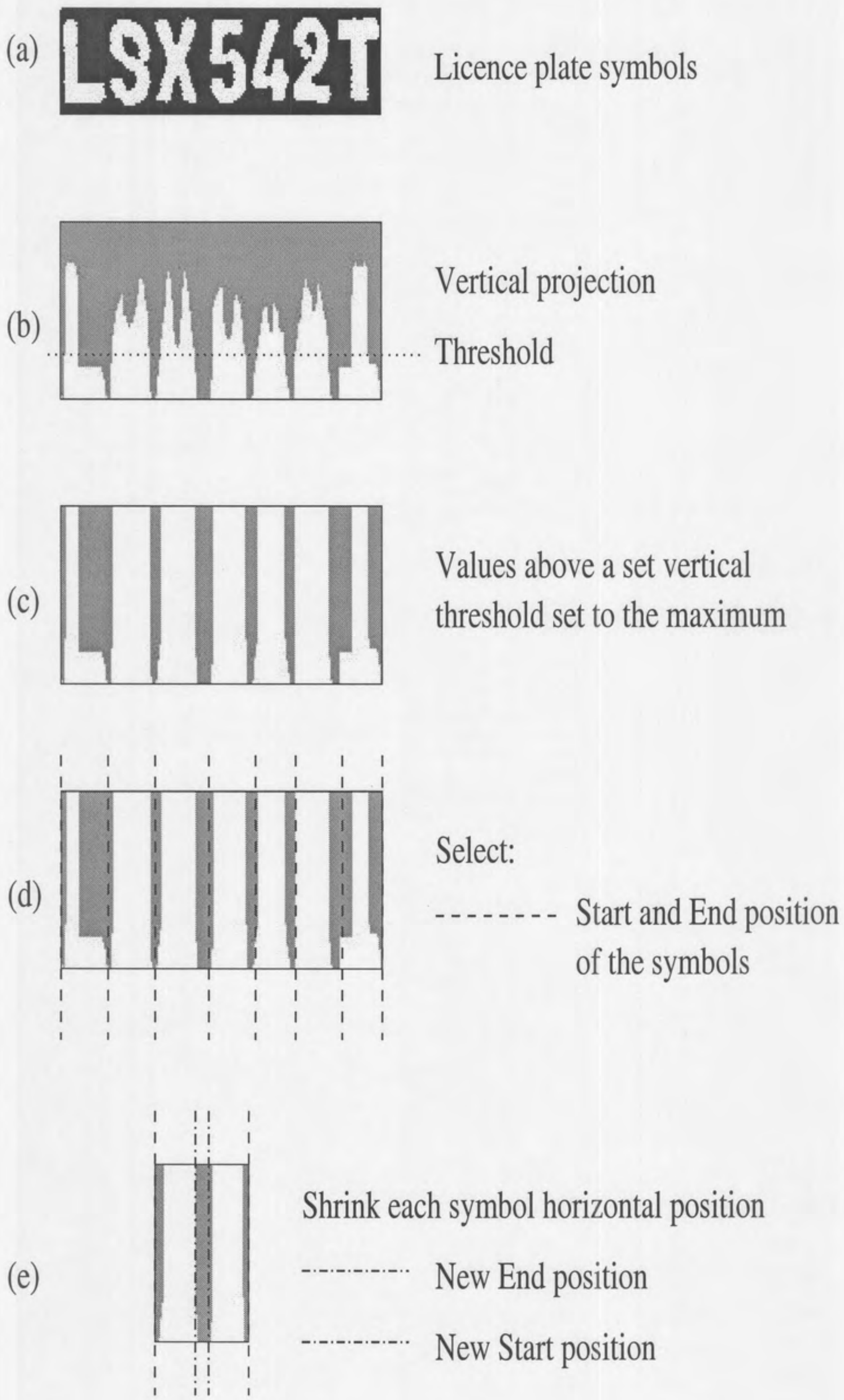

(a)
LSX542T
Licence plate symbols
(b)
Vertical projection
Threshold
(c)
Values above a set vertical threshold set to the maximum
(d)
Select:
Start and End position of the symbols
(e)
Shrink each symbol horizontal position
New End position
New Start position

Figure D.1: Valleys below a threshold algorithm example

Appendix E

Detail results of symbol segmentation experiment

In this appendix, more detail of experiment 3 is shown. In Figure E.1. The different scores for recognised and failed to recognise licence plates are shown. The techniques used were:

a Simple segmentation

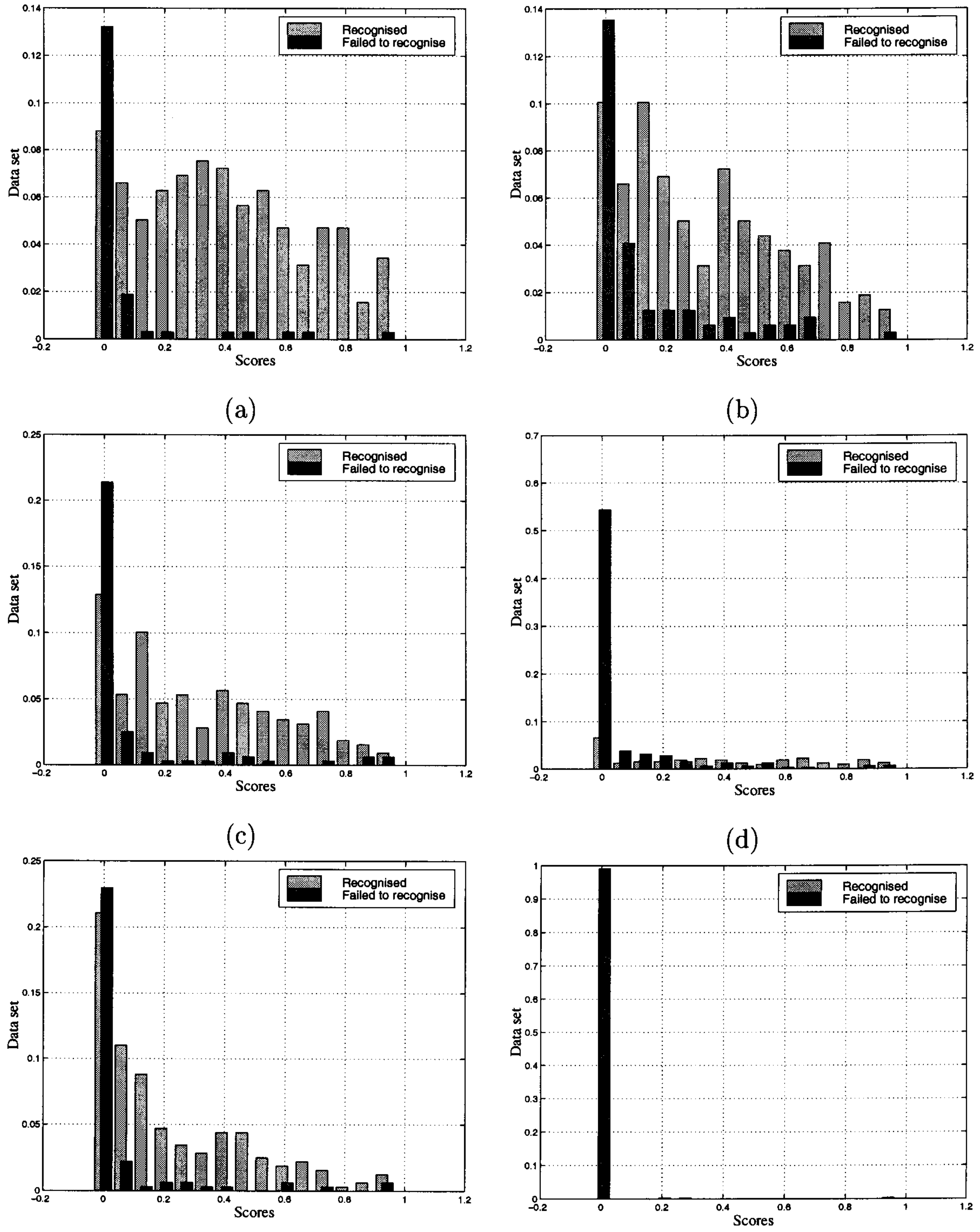
b Vertical projection with double frame Viterbi

c Vertical projection with single frame Viterbi

d Symbols between valleys with filtered data and double frame Viterbi

e Vertical projection with filtered data and single frame Viterbi

f Symbols between valleys with filtered data and single frame Viterbi



(a)

(b)

(c)

(d)

(e)

(f)

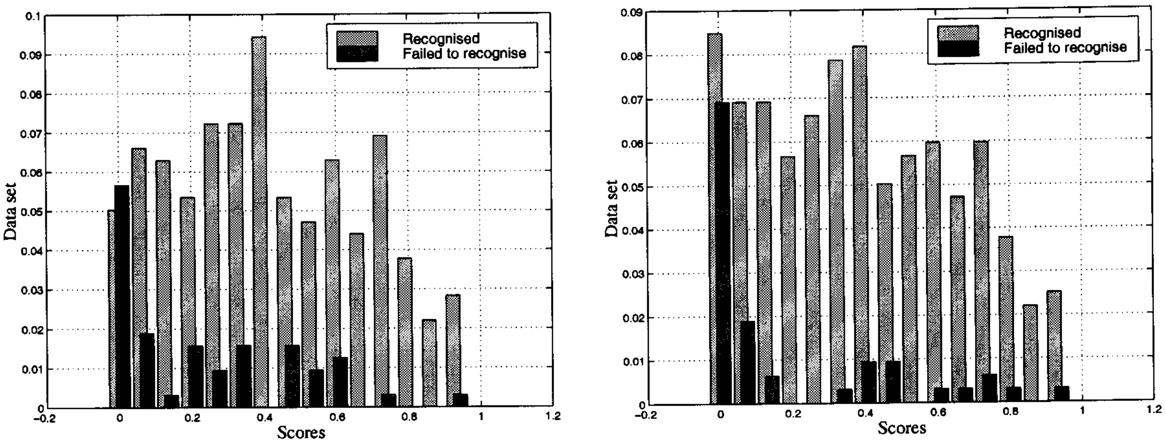
Figure E.1: Results of symbol segmentation experiment

By combining the different symbol segmentation techniques, a higher recognition rate was achieved. The highest recognition rate was achieved when the double Viterbi techniques were not included.

The results of symbol segmentation combination experiment are shown in Figure E.2. The different methods are:

a Combination of all techniques

b Combination of all techniques except the double Viterbi techniques



(a) (b)

Figure E.2: Results of symbol segmentation combination experiment

Appendix F

Optimisation of neural network input dimensions

Investigation of the initial weight influence

Four different initialisation weights were used in Figure F.1. The paths of the network error in Figure F.1, differs only by a small value. The minimum error falls between 1.1 and 1.4. The trend of the network errors are nearly the same. The influence of initialisation weights is not significant enough to optimise their values.

Scaled mesh function dimensions

In Table F.1 the cross validation error vs. the scaled mesh function size is shown.

The experiment was performed with a training set of 12015 images and a cross-validation set of 720 images. Each experiment was repeated three times, each time with different initialisation weights. The best result is shown. The results do not show any consistent trend, with neighbours differing vastly in values. Neither odd nor even values perform better than the other. Even 7 × 8 and 8 × 7 results are similar where it

Scaled mesh Dimensions	Validation Error	Scaled mesh Dimensions	Validation Error
5 × 7	0.0027	7 × 12	0.0016
5 × 8	0.0017	7 × 13	0.0033
5 × 9	0.0025	7 × 14	0.0018
5 × 10	0.0025	7 × 15	0.0019
5 × 11	0.0027	8 × 7	0.0029
5 × 12	0.0020	8 × 8	0.0023
5 × 13	0.0019	8 × 9	0.0021
5 × 14	0.0026	8 × 10	0.0021
5 × 15	0.0018	8 × 11	0.0017
6 × 7	0.0021	8 × 12	0.0017
6 × 8	0.0022	8 × 13	0.0023
6 × 9	0.0021	8 × 14	0.0022
6 × 10	0.0020	8 × 15	0.0018
6 × 11	0.0025	9 × 7	0.0025
6 × 12	0.0018	9 × 8	0.0021
6 × 13	0.0025	9 × 9	0.0024
6 × 14	0.0018	9 × 10	0.0022
6 × 15	0.0023	9 × 11	0.0017
7 × 7	0.0029	9 × 12	0.0025
7 × 8	0.0023	9 × 13	0.0027
7 × 9	0.0018	9 × 14	0.0029
7 × 10	0.0018	9 × 15	0.0021
7 × 11	0.0018		

Table F.1: The effect of the dimensions of the mesh function used for feature extraction on the cross validation error

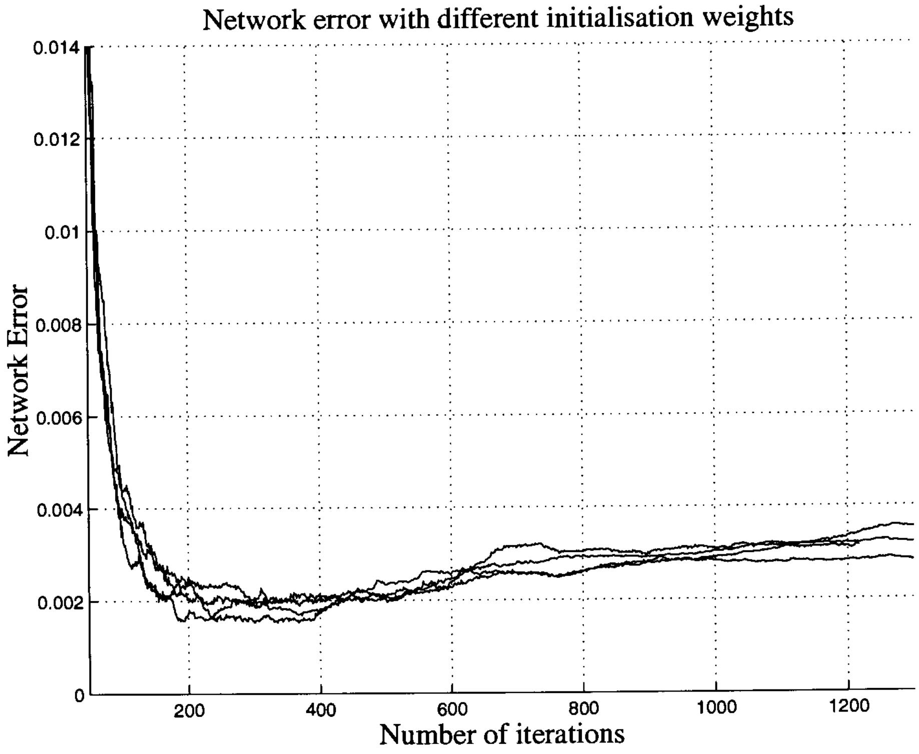


Figure F.1: Network error for cross validation set with different initialisation weights

would have been expected that 7 × 8 performs better. The best and the worst scores, 7 × 12 and 7 × 13 are right next to each other. The lack of a consistent trend can be a result of the small size of training and cross validation set, or be implicit in the problem.

The following figures, (Figure F.2 to Figure F.6), further searches for a trend in by looking at the effect of the input dimensions size, ratio and combination. No trend is visible in any of the figures.

For the best five mesh sizes, the experiment was repeated with 102123 images, and the results are shown in table F.2.

The results in Table F.1 and Table F.2 indicate that the number of training and cross validation images are to small to determine the optimal size for the mesh function.

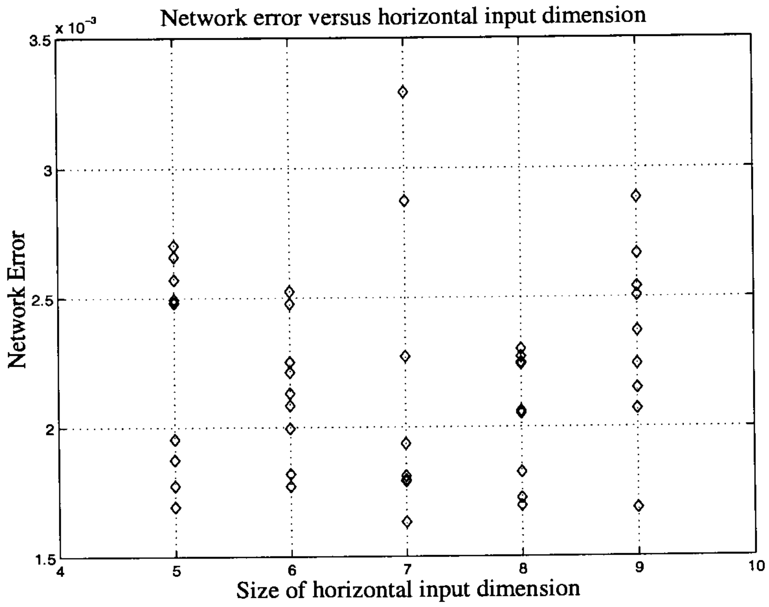


Figure F.2: Network error versus size of horizontal dimension

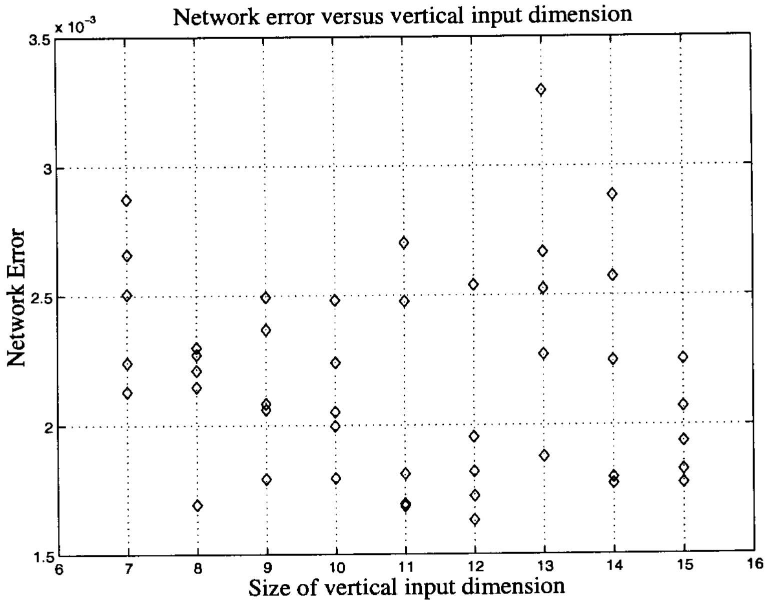


Figure F.3: Network error versus size of vertical dimension

With the dimensions (7 × 12), a 99.53% recognition rate was achieved on the test data set of 2117 licence plate symbols (recognising symbols already separated from the vehicle image).

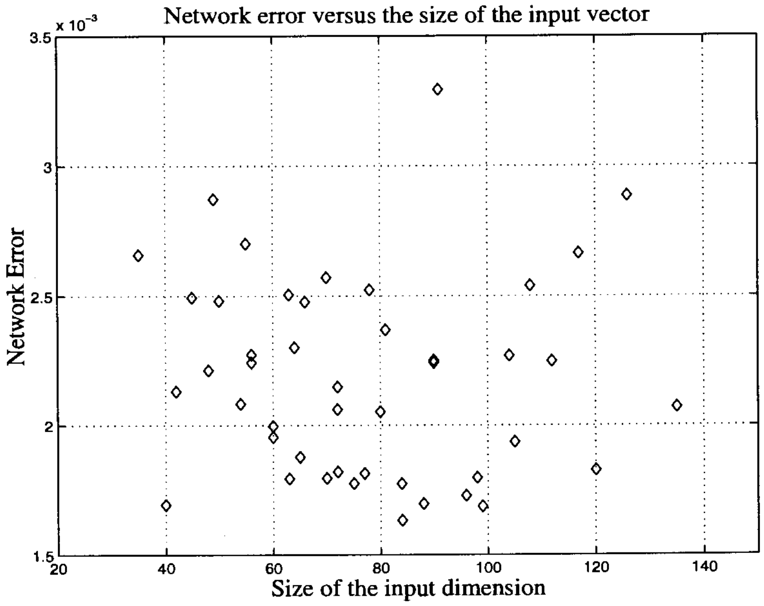


Figure F.4: Network error versus size of input vector

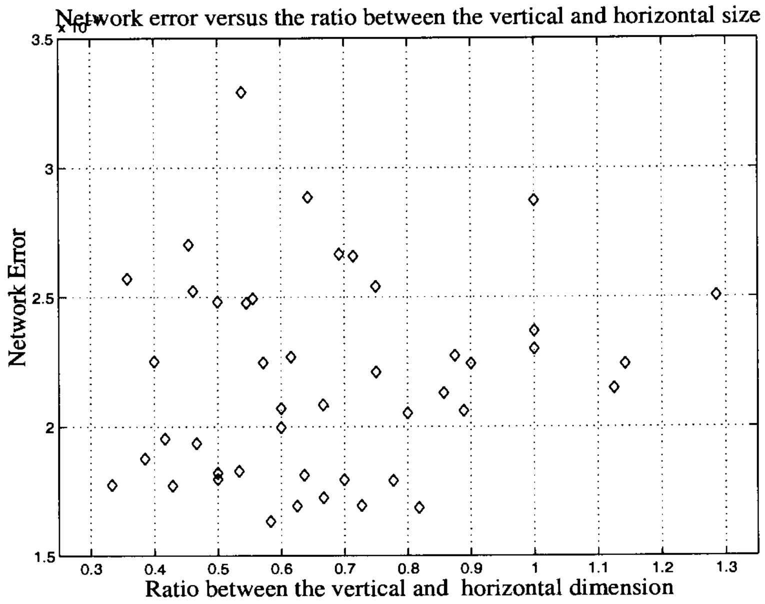


Figure F.5: Network error versus the horizontal and vertical dimensions ratio

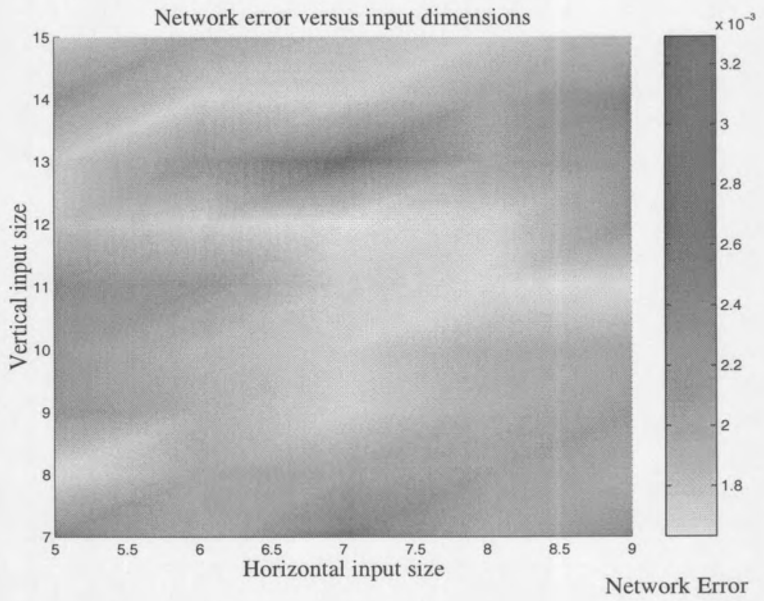


Figure F.6: Network error versus input dimensions

Scaled mesh Dimensions	Validation Error	Old Validation Error
7 × 12	0.0011	0.0016
9 × 11	0.0018	0.0017
5 × 8	0.0013	0.0017
8 × 11	0.0012	0.0017
8 × 12	0.0014	0.0017

Table F.2: The effect of the dimensions of the mesh function used for feature extraction on the cross validation error with a larger training set

Appendix G

Parameters used to obtain the optimal result

The parameters used to obtain the best result is shown in Table G.1 to Table G.6.

With these parameters the licence plate recognition rate was 93.7%. Out of 318 test vehicle images, 298 were recognised correctly. Of the remaining 20 vehicle images, 13 (65%) were segmentation errors while 7 (35%) were character recognition errors.

All the transition probabilities of the Markov model were set to 1, as the limited size of the data set prevented any realistic optimisation of the transition probabilities.

Parameter	Value
Local threshold block size	16 pixels
Local threshold offset value	-16 to 9

Table G.1: The best local threshold parameters used for the optimal result

Parameter	**Value**
Minimum distance between symbols	5 pixels
Maximum distance between symbols	100 pixels
Minimum number of pixels for a symbol	200 pixels
Maximum number of pixels for a symbol	900 pixels
Maximum vertical size of a symbol	150 pixels
Minimum symbol vertical to horizontal ratio	1.2
Minumum number of symbols in a line	2

Table G.2: The best region growing parameters for licence plate segmentation

Parameter	**Value**
Vertical distance between lines	25 pixels
Threshold between horizontal pixels	0.2
Maximum horizontal size for threshold	4 pixels
Maximum size between vertical edges	35 pixels
Minimum edges in a row	30 edges
Minimum size of horizontal row	60 pixels
Interval size	5 pixels
Maximum size of a symbol	60 pixels

Table G.3: The best vertical edges parameters for licence plate segmentation

Parameter	**Value**
Symbol segmentation offset value	-5 to 4
Vertical projection threshold	0.04 to 0.1
Minimum vertical average	25 % of symbol height
Minimum horizontal size of symbol	5 % of licence plate width

Table G.4: Symbol segmentation parameters

Parameter	**Value**
Mesh size	7 × 12
Number of hidden neurons	25
Network layers	3

Table G.5: Neural network parameters

Parameter	**Value**
Simple segmentation with single frame Viterbi	Used
Threshold on the vertical projection with single frame Viterbi	Used
Threshold on the vertical projection with filtered data and single frame Viterbi	Used
Threshold on the vertical projection with double frame Viterbi	Not Used
Symbols lie between two major valleys of the projection with filtered data and double frame Viterbi	Not Used
Symbols lie between two major valleys of the projection and with single frame Viterbi	Not Used
Symbols lie between two major valleys of the projection and with filtered data and single frame Viterbi	Not Used

Table G.6: Symbol segmentation techniques used

Bibliography

[1] S. Kahan, T. Pavlidis, and H. S. Baird, "On the recognition of printed characters of any font and size," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-9, pp. 274–288, March 1987.

[2] H. I. Avi-Itzhak, T. A. Diep, and H. Garland, "High accuracy optical character recognition using neural networks with centroid dithering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, pp. 218–224, February 1995.

[3] J. Ohya, A. Shio, and S. Akamatsu, "Recognizing characters in scene images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, pp. 214–220, February 1994.

[4] O. D. Trier, T. Taxt, and A. K. Jain, "Recognition of digits in hydrographic maps: Binary versus topographic analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 399–404, April 1997.

[5] Y. Cheng, H. Huang, and W. Hsu, "Locating bookbacks in a bookrack image," in *International Conference on Pattern Recognition*, (Brisbane, Australia), pp. 1822–1824, ICPR, August 1998.

[6] S. Lee and Y. J. Kim, "Direct extraction of topographic features for gray scale character recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, pp. 724–729, July 1995.

[7] W. W. C. Jiang, "Thresholding and enhancement of text images for character recognition," in *The 1995 International Conference on Acoustics, Speech and Signal Processing*, (Detroit, Michigan USA), pp. 2395–2397, ICASSP, May 1995.

[8] Y. Liu and S. Srihari, "Ducument image binarization based on texture features," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 540–544, May 1997.

[9] G. Kim and V. Govindaraju, "A lexicon driven approach to handwritten word recognition for real-time applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 366–379, April 1997.

[10] A. W. Senior and A. J. Robinson, "An off-line cursive handwriting recognition system," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, pp. 309–321, March 1998.

[11] M. Mohamed and P. Gader, "Handwritten word recognition using segmentation-free hidden Markov modeling and segmentation-based dynamic programming techniques," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, pp. 549–554, May 1996.

[12] R. G. Casey and E. Lecolinet, "A survey of methods and strategies in character segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, pp. 690–706, July 1996.

[13] C. Rodriquez, J. Muguerza, M. Navarro, A. Zarate, J. I. Martin, and J. M. Perez, "Segmentation of low-quality typewritten digits," in *International Conference on Pattern Recognition*, (Brisbane, Australia), pp. 1106–1109, ICPR, August 1998.

[14] J. Mao, P. Sinha, and K. Mohiuddin, "A system for cursive handwritten address recognition," in *International Conference on Pattern Recognition*, (Brisbane, Australia), pp. 1285–1287, ICPR, August 1998.

[15] M. Sawaki, H. Murase, and N. Hagita, "Character recognition in bookshelf images by automatic template selection," in *International Conference on Pattern Recognition*, (Brisbane, Australia), pp. 1117–1120, ICPR, August 1998.

[16] A. Simon, J. Pret, and A. P. Johnson, "A fast algorithm for bottom-up document layout analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 273–277, March 1997.

[17] A. K. Jain and B. Yu, "Document representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, pp. 294–307, March 1998.

[18] L. Gu, N. Tanaka, and R. Haralick, "Robust extraction of characters from color scene image using mathematical morphology," in *International Conference on Pattern Recognition*, (Brisbane, Australia), pp. 1002–1004, ICPR, August 1998.

[19] U. Gargi, S. Antani, and R. Kasturi, "Indexing text events in digital video databases," in *International Conference on Pattern Recognition*, (Brisbane, Australia), pp. 916–918, ICPR, August 1998.

[20] J. Shim, C. Dorai, and R. Bolle, "Automatic text extraction from video for content-based annotation and retrieval," in *International Conference on Pattern Recognition*, (Brisbane, Australia), pp. 916–918, ICPR, August 1998.

[21] P. Tofani and R. Kasturi, "Segmentation of text from color map images," in *International Conference on Pattern Recognition*, (Brisbane, Australia), pp. 945–947, ICPR, August 1998.

[22] Y. Lu, "Machine printed chracter segmentation - an overview," *Pattern Recognition*, vol. 28, no. 1, pp. 67–80, 1995.

[23] C. Nieuwoudt and R. van Heerden, "Automatic number plate segmentation and recognition," in *Seventh annual South African workshop on Pattern Recognition*, (University of Cape Town), pp. 88–93, IAPR, Nov 1996.

[24] R. van Heerden and C. Nieuwoudt, "A connectionist approach to vehicle licence plate recognition," in *Eight annual South African workshop on Pattern Recognition*, (Rhodes University), pp. 75–79, IAPR, Nov 1997.

[25] S. Mori, C. Suen, and K. Yamamoto, "Historical review of ocr research and development," *Proceesings of the IEEE*, vol. 80, pp. 1029–1058, July 1992.

[26] V. Govindan and A. Shivaprasad, "Character recognition - a review," *Pattern Recognition*, vol. 23, no. 7, pp. 671–683, 1990.

[27] C. Rodriguez, J. Muguerza, M. Navarro, J. Martin, and J. Perez, "A two-stage classifier for broken and blurred digits in forms," in *International Conference on Pattern Recognition*, (Brisbane, Australia), pp. 1101–1105, ICPR, August 1998.

[28] M. C. de Beer and E. C. Botha, "Comparison and combination of five feature spaces for recognition of handwritten digits," in *Proceedings of the Third South African Workshop of Pattern Recognition*, (Pretoria University), pp. 194–198, November 1992.

[29] L. Lam and C. Y. Suen, "An evaluation of parallel thinning algorithms for character recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, pp. 915–919, September 1995.

[30] N. Muller and B. Herbst, "The use of eigenpictures for optical character recognition," in *International Conference on Pattern Recognition*, (Brisbane, Australia), pp. 1124–1126, ICPR, August 1998.

[31] R. Schalkoff, *Pattern Recognition Statistical, Structural and Neural approaches.* New York: John Wiley & Sons, Inc., 1992.

[32] S. Knerr, L. Personnaz, and G. Dryfus, "Handwritten digit recognition by neural networks with single-layer training," *IEEE Transactions on Neural Networks*, vol. 3, pp. 962–968, November 1992.

[33] W. E. Weideman, M. T. Manry, and H. C. Yau, "A comparison of a nearest neighbor classifier and a neural network for numeric handprinted character recognition,"

in *International Joint Conferance on Neural Networks*, (Seattle, WA, USA), pp. I 117–120, IJCNN, June 1989.

[34] E. Sackinger, Y. LeCun, B. E. Boser, J. Bromley, and L. D. Jackel, "Application of the ANNA neural network chip to high-speed character recognition," *IEEE Transactions on Neural Networks*, vol. 3, pp. 498–505, May 1992.

[35] F. Kimura, S. Inoue, T. Wakabayashi, S. Tsuruoka, and Y. Miyake, "Handwritten numeral recognition using autoassociative neural networks," in *International Conference on Pattern Recognition*, (Brisbane, Australia), pp. 166–171, ICPR, August 1998.

[36] S. Lee, "Off-line recognition of totally unconstrained handwritten numerals using multilayer cluster neural network," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, pp. 648–652, June 1996.

[37] K. Lim and S. Chien, "Neural network based feature space generation for multiple databases of handwritten numerals," in *International Conference on Pattern Recognition*, (Brisbane, Australia), pp. 375–377, ICPR, August 1998.

[38] G. L. Martin and J. A. Pittman, "Recognizing hand-printed letters and digits using backpropation learning," *Neural Computation*, no. 3, pp. 259–267, 1991.

[39] T. Cho and R. W. Conners, "A neural network approach to machine vision systems for automated industrial inspection," in *International Joint Conferance on Neural Networks*, (Seattle, WA, USA), pp. I 205–209, IJCNN, July 1991.

[40] L. R. Rabiner and B. H. Juang, "An introduction to hidden Markov models," *IEEE ASSP Magazine*, pp. 4–7, January 1988.

[41] L. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, pp. 257–286, February 1989.

[42] M. Koschinski, H. Winkler, and M. Lang, "Segmentation and recognition of symbols within handwritten mathematical expressions," in *The 1995 International*

Conference on Acoustics, Speech and Signal Processing, (Detroit, Michigan USA), pp. 2439–2442, ICASSP, May 1995.

[43] A. Kosmala and G. Rigoll, "On-line handwritten formula using statistical methods," in *International Conference on Pattern Recognition*, (Brisbane, Australia), pp. 1306–1308, ICPR, August 1998.

[44] X. Li, R. Plamondon, and M. Parizeau, "Model-based on-line handwritten digit recognition," in *International Conference on Pattern Recognition*, (Brisbane, Australia), pp. 1134–1136, ICPR, August 1998.

[45] J. Chai and Z. Liu, "Integration of structural and statistical information for unconstrained handwritten numeral recognition," in *International Conference on Pattern Recognition*, (Brisbane, Australia), pp. 378–380, ICPR, August 1998.

[46] N. Arica and F. T. Y. Vural, "A new scheme for off-line handwritten connected digit recognition," in *International Conference on Pattern Recognition*, (Brisbane, Australia), pp. 1127–1129, ICPR, August 1998.

[47] K. Nathan, H. S. M. Beigi, J. Subrahmonia, G. J. Clary, and H. Maruyama, "Realtime on-line unconstrained handwriting recognition using statistical methods," in *The 1995 International Conference on Acoustics, Speech and Signal Processing*, (Detroit, Michigan USA), pp. 2619–2622, ICASSP, May 1995.

[48] S. Procter, J. Illingworth, and A. J. Elms, "The recognition of handwritten digit strings of unknown length using hidden Markov models," in *International Conference on Pattern Recognition*, (Brisbane, Australia), pp. 1515–15–17, ICPR, August 1998.

[49] B. Kee and J. H. Kim, "Ligature modeling for online cursive script recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 623–633, June 1997.

[50] E. J. Bellegarda, J. R. Bellegarda, D. Nahamoo, and K. S. Nathan, "A discrete paramter HMM approach to on-line handwritten recognition," in *The 1995 Inter-*

national Conference on Acoustics, Speech and Signal Processing, (Detroit, Michigan USA), pp. 2631–2634, ICASSP, May 1995.

[51] M. Chen and A. Kundu, "Multi level HMM for handwritten word recognition," in *The 1995 International Conference on Acoustics, Speech and Signal Processing*, (Detroit, Michigan USA), pp. 2623–2625, ICASSP, May 1995.

[52] N. Morgan and H. Bourlard, "Continuous speech recognition," *IEEE Signal Processing Magazine*, pp. 25–42, May 1995.

[53] Y. Bengio, R. D. Mori, G. Flammia, and R. Kompe, "Global optimization of a neural network-hidden Markov model hybrid," *IEEE Transactions on Neural Networks*, vol. 3, pp. 252–259, March 1992.

[54] G. Rigoll, A. Kosmala, and D. Willett, "A new hybrid approach to large vocabulary cursive handwritten recognition," in *International Conference on Pattern Recognition*, (Brisbane, Australia), pp. 1512–1513, ICPR, August 1998.

[55] S. Knerr and E. Augustin, "A neural network-hidden Markov model hybrid for cursive word recognition," in *International Conference on Pattern Recognition*, (Brisbane, Australia), pp. 1518–1520, ICPR, August 1998.

[56] D. Guillevic and C. Suen, "HMM-KNN word recognition engine for bank cheque processing," in *International Conference on Pattern Recognition*, (Brisbane, Australia), pp. 1526–1529, ICPR, August 1998.

[57] J. Dolfing, E. Aarts, and J. van Oosterhout, "On-line signature verification with hidden Markov models," in *International Conference on Pattern Recognition*, (Brisbane, Australia), pp. 1309–1312, ICPR, August 1998.

[58] S. Draghici, "A neural network based artificial vision for licence plate recognition," *International Journal of Neural Systems*, vol. 8, no. 1, pp. 113–126, 1997.

[59] J. Barroso, A. Rafeal, E. L. Dagless, and J. Bulas-Cruz, "Number plate reading using computer vision," in *International Symposium on Industrial Electronics*, (Universidade do Minho), ISIE, 1997.

[60] J. Barroso, E. L. Dagless, and J. Bulas-Cruz, "Real-time number plate reading," in *Workshop on Algorithms and Architectures for Real-time Control*, (Vilamoura, Portugal), IFAC, April 1997.

[61] C. Setchell, *Applications of Computer Vision to Road-traffic Monitoring.* PhD thesis, Department of Computer Science, University of Bristol, September 1997.

[62] C. Coetzee, C. Botha, and D. Weber, "PC based number plate recognition system," in *IEEE International Symposium on Industrial Elextronics*, (Pretoria, South Africa), pp. 605–610, ISIE, July 1998.

[63] Y. K. Siah, T. Y. Haur, M. Khalid, and T. Ahmad, "Vehicle licence plate recognition by fuzzy Art-map neural network," (Faculty of Electronic Engineering, University Teknologi Malaysia, Jalan Semarak, Kuala Lumpur), Centre for Artificial Intelligence and Robotics.

[64] M. M. M. Fahmy, "Computer vision application to automatic number-plate recognition," (Croydon, England, Automotive Technology and Automation), pp. 625–632, International Symposium on Automotive Technology and Automation, 1993.

[65] R. A. Lotufo, A. Morgan, A. S. Johnson, and T. B. Thomas, "A transputer based automatic number-plate recognition system," in *The Second International Conference on Applications of Transputers*, (University of Southampton, Southampton, UK.), pp. 196–202, TA90, July 1990.

[66] G. Auty, P. Corke, P. Dunn, M. Jensen, and I. Macintyre, "An image acquisition system for traffic moniroring appliactions," in *Cameras and systems for Electronic Photography and Science Imagery*, pp. 118–133, SPIE, 1995.

[67] J. Chang and N. C. Griswold, "A hierarchical multilayer perceptron neural network for the recognition of the automobile license plate number," (Department of Electrical Engineering, Texas AM University, College station, TX 77840), pp. 138–144, SPIE, 1996.

[68] L. C. Howington, "Automated licence plate reading," *Advanced imaging*, vol. 4, pp. 46–50, September 1989.

[69] C. Hwang, S. Shu, W. Chen, Y. Chen, and K. Wen, "A pc-based car licence plate reader," pp. 272–283, SPIE, 1992.

[70] M. Takatoo, M. Kanasaki, T. Mishima, T. Shibata, and H. Ota, "Gray scale image processing technology applied to vehicle licence number recognition system," in *Industrial applications of machine vision and machine intelligence*, pp. 76–79, 1987.

[71] T. Taxt, P. J. Flynn, and A. K. Jain, "Segmentation of document images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, pp. 1322–1329, December 1989.

[72] X. Zhao and S. H. Ong, "Adaptive local thresholding with fuzzy-validity-guided spatial partitioning," in *International Conference on Pattern Recognition*, (Brisbane, Australia), pp. 988–990, ICPR, August 1998.

[73] O. D. Trier and T. Taxt, "Evaluation of binarization methods for document images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, pp. 312–315, March 1995.

[74] D. Hammerstom, "Neural networks at work," *IEEE Spectrum*, pp. 26–32, June 1993.

[75] D. Hammerstom, "Working with neural networks," *IEEE Spectrum*, pp. 46–53, July 1993.

[76] J. Vermaak, "Connectionist models for short-term load forecasting," Master's thesis, University of Pretoria, Pretoria, South Africa, 1996.

[77] J. de Villiers and E. Barnard, "Backpropgation Neural Nets with one and two hidden layers," *IEEE Transactions on Neural Networks*, vol. 4, pp. 136–141, January 1992.

[78] D. da Silva Goncalves and E. Barnard, "Best size for a neural network," Master's thesis, Department of Electronic Engineering, University of Pretoria, Pretoria, South Africa, September 1994.

[79] C. Bishop, *Neural networks for patten recognition.* London: Oxford University Press, 1995.

[80] H. F. Siverman and D. P. Morgan, "The application of dynamic programming to connected speech recognition," *IEEE ASSP Magazine*, pp. 30–32, July 1990.

[81] L. Rabiner and B. Juang, *Fundamentals of Speech Recognition.* Englewood Cliffs, New Jersey 07632: Prentice Hall Processing series, 1993.

[82] O. Shiku, K. Kwasue, and A. Nakamura, "A method for character string extraction using local nad global segment crowdedness," in *International Conference on Pattern Recognition*, (Brisbane, Australia), pp. 1077–1080, ICPR, August 1998.

[83] W. Pratt, *Digital Image Processing.* Mountain View, California: Sun Microsystems, Imc, 1991.

[84] M. Koga, T. Kagehiro, H. Sako, and H. Fujisawa, "Segmentation of Japanese handwritten characters using peripheral feature analysis," in *International Conference on Pattern Recognition*, (Brisbane, Australia), pp. 1137–1141, ICPR, August 1998.

[85] A. K. Jain and D. Zongker, "Representation and recognition of handwritten digitis using deformable templates," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 1386–1391, December 1997.