

Computer Input of Morse Codes Using Finger Gesture Recognition

Ricky Li

A thesis submitted to the Auckland University of Technology
in partial fulfilment of the requirements for the degree of
Master of Computer and Information Sciences (MCIS)

2017

School of Engineering, Computer and Mathematical Sciences

Abstract

The Morse code is one of the earliest means of telecommunications; however, it is rarely used nowadays due to viral mobile communications. Although a person can tap Morse codes using his fingers easily, perhaps nobody is aware of this kind of finger gestures anymore. In this thesis, we will develop a prototype combined the principle of old Morse code with finger gesture recognition in machine learning together. A camera is used to capture a sequence of video frames, the prototype will recognize the finger gestures from these frames and convert the corresponding Morse codes to readable ASCII letters, characters or emotional symbols. The significant work could be applied to those special communications or dialogues, not allowed to speak loudly and explicitly. The contributions of this thesis are the finger gesture recognition based on empirical approaches for Morse code input; the highest recognition rate is up to 93%.

Keywords: gesture recognition, Morse code, fingertip tracking, SVM (support vector machine), Gaussian pyramid, BPNN

Contents

Abstract	2
List of Figures	5
List of Tables.....	6
List of Algorithms	7
Attestation of Authorship.....	8
Acknowledgment	9
Chapter 1 Introduction	10
1.1 Background and Motivation.....	10
1.2 Research Questions	13
1.3 Objective of this Thesis.....	14
1.4 Structure of this Thesis.....	14
1.5 Novelty of the Vision-based Morse Code Recognition	15
Chapter 2 Literature Review	16
2.1 Introduction	16
2.2 Segmentation Algorithms	16
2.1.1 Threshold Image Segmentation Algorithm.....	16
2.1.2 Active Contour Model Algorithm.....	17
2.1.3 Boundary-based Segmentation Algorithms	18
2.1.4 Region Growing Segmentation Algorithm	19
2.3 Classification Algorithms	23
2.3.1 Principal Component Analysis and Derived Algorithms.....	23
2.3.2 A linear Scaling Model for the Fingertip	24
2.4 Recognition Algorithms	24
2.4.1 Instance-based Learning.....	24
2.4.2 Hidden Markov Model.....	26
2.4.3 Artificial Neural Networks.....	27
2.4.4 Support Vector Machine	28
2.4.5 Convolutional Neural Network.....	29
2.5 Relevant Colour Space Analysis.....	31
2.5.1 Two-dimensional Gaussian Distribution.....	32
2.5.2 YCbCr Colour Space	32
2.6 Scale-invariant Feature Transform (SIFT).....	34

Chapter 3 Research Methodology.....	36
3.1 Introduction	36
3.2 Related Work	36
3.3 Data Acquisition.....	38
3.4 Research Design.....	39
3.4.1 Morse Code Design.....	39
3.4.2 Architecture.....	41
3.5 Algorithms	47
3.5.1 Image Segmentation Method	47
3.5.2 Feature Extraction & SVM/BPNN Classification Method	48
3.6 Expected Outcomes.....	54
Chapter 4 Research Findings	55
4.1 Introduction	55
4.2 Experimental Environment	55
4.3 Experiments.....	55
4.3.1 Experimental Algorithms	55
4.3.2 Experiment	58
4.4 Experiment Results	61
4.4.1 Finger Gesture Recognition Based on Image Segmentation	61
4.4.2 Finger Gesture Recognition Based on SVM/BPNN	62
Chapter 5 Discussions and Analysis	77
5.1 Introduction	77
5.2 Analysis and Evaluation.....	77
5.3 Justifications.....	78
5.4 Limitations	79
5.5 Discussions.....	80
5.6 Recommendations	80
Chapter 6 Conclusion and Future Work	82
6.1 Conclusion	82
6.2 Future Work	82
References	83
Appendix.....	93

List of Figures

Figure 1.1 The basic framework of finger gesture recognition.....	14
Figure 2.1 Threshold segment.....	19
Figure 2.2 Region growing.....	23
Figure 2.3 Hidden Markov model.....	30
Figure 2.4 Neural network model.....	31
Figure 2.5 The generation of a Gaussian differential pyramid.....	39
Figure 3.1 Flowchart of the finger gesture recognition based on image segmentation...	46
Figure 3.2 Flowchart of image pre-processing.....	47
Figure 3.3 Flowchart of finger gesture analysis & recognition.....	48
Figure 3.4 Flowchart of decoding and video output.....	49
Figure 3.5 Flowchart of the finger gesture recognition based on a machine learning algorithm.....	49
Figure 3.6 Flowchart of detailed shallow learning algorithm.....	50
Figure 3.7 Cb image after colour segmentation.....	51
Figure 3.8 Processed binarized image.....	52
Figure 3.9 Training sample selection.....	53
Figure 3.10 Gaussian pyramid.....	54
Figure 3.11 Image at different scales.....	55
Figure 3.12 Support vector.....	56
Figure 4.1 Original frames.....	64
Figure 4.2 Segmented binary image.....	64
Figure 4.3 Final output frames.....	65
Figure 4.4 Video frames of fingertip recognition.....	65
Figure 4.5 Match metric of gesture recognition.....	66
Figure 4.6 Fingertip recognition match metric.....	66

List of Tables

Table 3.1 Original Morse codes.....	43
Table 3.2 Punctuations.....	44
Table 3.3 SMS and emoticons.....	44
Table 3.4 Morse codes of mathematical symbols.....	44
Table 3.5 Table 3.5 Morse codes of simplified Chinese characters.....	45
Table 4.1 Results of accuracy of Morse codes for all testers.....	67
Table 4.2 Overall average accuracy for all characters.....	75

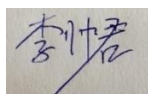
List of Algorithms

Algorithm 4.1 Processing Binarized Image.....	61
Algorithm 4.2 Image Segmentation	62

Attestation of Authorship

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person (except where explicitly defined in the acknowledgments), nor material which to a substantial extent has been submitted of the award of any other degree or diploma of a university or other institution of higher learning.

Signature:

A handwritten signature in Chinese characters, appearing to be '李仲君' (Lǐ Zhōngjūn), written in blue ink on a light-colored background.

Date: 14 July 2017

Acknowledgment

This thesis would not have been completed without much assistance, encouragement and support from a lot of people. I sincerely avail this opportunity to express my cordial thanks to those who have granted me invaluable instructions during the thesis writing.

First and foremost, I extend my greatest gratitude to my supervisor Dr Wei Qi Yan for his insightful guidance and earnest help. He advised me to think over the selection of this topic and carry out a series of relevant research experiments at a very early time; and during the process of writing, he spent a lot of time guiding me in the right direction and provided a great deal of useful suggestions. It is under his help that I could complete this thesis in time.

Moreover, my sincere thanks to the allied staff from school of engineering, computer and information sciences at the Auckland University of Technology. They provided me with a wonderful learning environment where I learn and grow up for their tireless instructions that will definitely exert a deep influence on my later life.

Besides, I express my gratitude to my friends and fellow classmates. They shared their knowledge with me and helped me out when I faced with any difficulties. They have tried their best to give me their precious suggestions during the process of writing the thesis.

Last but not least, I am deeply in debt to my beloved parents for their encouragement, understanding and endless love during my life. They have created the best environment for me to focus on the thesis writing during this year, and all this could not be possible without their selfless and persistent support.

Chapter 1 Introduction

1.1 Background and Motivation

Vision-based gesture recognition technology is a branch of natural human-computer interaction and is one of the important components in the computer vision. Gesture recognition is the recognition of various gestures conducted in accordance with certain rules through computers. This indicates the appropriate control commands or semantics to achieve computer control or exchange of information (Wu & Huang, 1999).

In the history of the development of human-computer interaction, human initially adapted to the computer and then moved on to a period where the computer was adapted to human preference (Bo, 1982). Human beings have spent years learning how to make computers and how to write computer programs. The next stage is naturally to make computers serve and adapt to human needs using artificial intelligence (Rautaray & Agrawal, 2012).

A natural and harmonious human-computer interaction mode is dominated by direct manipulation and the command language, especially natural language, coexists with human-computer interaction. The ideal human-computer interaction model is a user-friendly and straightforward user interface which has become the trend of the future (Dix, 2009). This chapter first analyses the shortcomings of the current human-computer interaction system; then, an introduction of the basic framework, classification, design ideas and related issues, and the development of gesture recognition will be presented. Lastly, the overview of this thesis will be presented.

With the rapid development of computer hardware, the user interface is constantly improving the quality of our ordinary life. A keyboard is based on the model of the initial user interface, and the invention of the mouse has moved us into the era of the graphical user interface; however, these mechanical devices are inconvenient and inappropriate for direct interaction because these devices have difficulties in performing 3D and highly free input (Segen & Kumar, 1999). With the rapid development of computer technology, the study of innovative human-computer interaction technology has become a high-interest area, and has made gratifying progress; these studies include face recognition, facial expression recognition, lip reading, head movement tracking, gaze tracking, gesture

recognition, and body recognition (Toole, Millward & Anderson, 1988). In general, the human-computer interaction technology has shifted from a computer-centric to a people-centred, multimedia and multimode interactive technology.

Human interaction often has sound and expression, in addition to the use of natural languages (spoken language, written language), the body language (expression, body, gestures) is some of the basic forms of human interaction. Compared with human interaction, human-computer interaction is much more rigid. Research on human body language is very meaningful to enhance the utility of the man-machine interface. Gesture is a language used which is a relatively stable for expression. Composed of hand movements and facial expressions, it is a special language that communicates by action and vision (Soni,,Nagmode & Komati,. 2016). Thus, in human-computer interaction, gestures can be used. Here are some areas that gestures can be applied.

- In the virtual reality environment, the application of environment and virtual objects can be controlled by gesture.
- Intelligent home appliances.
- Robotic control and remote robot operations.
- The education and lives of children, the elderly or deaf people.

The gesture recognition is mainly divided into a glove recognition and a vision-based gesture recognition. Gesture recognition is based on data gloves. With the use of gloves, locating tracking gestures in the spatiotemporal space has the advantage of a high recognition rate. The disadvantage is the need to wear the gloves and a position tracker (Weissmann & Salomon,, 1999). This method is now somewhat out of date, but still, some research on gesture recognition is based on data gloves, Using a 54-dimension data glove based on hand gesture recognition, it achieves better performance in robotic teleoperations (Lu, Yu & Liu, 2016).

Nowadays, most researchers mainly investigate vision-based gesture recognition, also known as naked hand recognition. Vision-based gesture recognition systems use cameras to capture gestures and recognize them. The advantage of this method is that the input device is relatively inexpensive, but vision-based gesture recognition is much sensitive to

a complex background, including lighting and shadow, distance, angle, and so on; these conditions affect the accuracy of identification. Foreign researchers have studied vision-based gesture recognition for a long time. Fujitsu Laboratories completed the identification of 46 hand gesture symbols in 1991 (Takahashi. & Kishino, 1991). Davis and Shah wear a fingertip with a brightly coloured glove to make gestures as an input tool; seven specific hand gestures can be identified (Davis, & Shah, 1994). Starner et al. could identify American sign language using hand gesture recognition; the accuracy rate reached 99.2% (Starner & Pentland, 1995). Grobel and Assam extracted features from the video recording using HMM to identify 262 isolated words with an accuracy rate of 91.3% (Grobel & Assam, 1997).

A complete vision-based gesture recognition consists of three parts: the acquisition, classification and recognition parts. The three components are: segmentation, analysis and recognition shown in Figure 1.1.

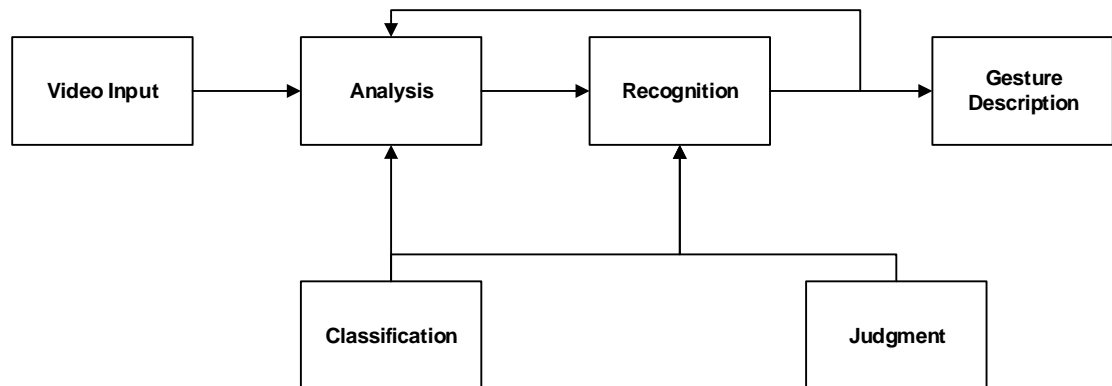


Figure 1.1 The basic framework of a finger gesture recognition

The acquisition devices include a camera, video card and the memory component. In the multiple gesture recognition (Usabiaga, Erol, Bebis, Boyle & Twombly, 2008), the cameras were placed in front of the user. In the case of single-gesture recognition, the plane of the camera should be flat with the level of the user's hand movement (Segen & Kumar, 1999).

The classification section includes the classifier to be processed and the result is then fed back to the receiver. It is used to correct the previous recognition result. The recognition part includes the syntax corresponding unit and the corresponding tracking mechanism,

where the classification of the hand shape is obtained by the corresponding semantics and control commands.

The segmentation process involves a frame-by-frame based segmentation of video frames. We first obtain the regions of concern, and then split the region until the desired finger and palm shape are obtained. The tracking process includes tracking the hand and estimating the position of the next frame. The identification process determines the meaning of the gesture through previous knowledge and makes a corresponding response, such as showing a corresponding gesture or making a corresponding action.

The inspiration of this project is a movie that shows a kidnapped child secretly communicating with his father by Morse code. Morse code is a unique method using hidden communication in this digital era. Historically, Morse code has been used for military purposes over a hundred years. Morse code can easily be described by any parts of our bodies, as it only contains two signals called dots and dashes. In this project, the focus will use hand gestures to send out Morse codes.

Another critical component of this project is surveillance. Nowadays, people are being watched everywhere. Most places such as streets, shopping malls, campuses and business buildings are covered by cameras. Since surveillance is everywhere in this modern world, sending Morse codes by finger gestures can be a secret way of sending messages. It can be used in many special and extreme circumstances, such as a spy sending encrypted messages, or a hostage asking for help. People can freely and secretly send out a message through a connected camera to the other side of the Internet. It is also a unique and creative way to combine a new technology and a traditional and soundless communication method together.

1.2 Research Questions

The purpose of this thesis is to design a finger gesture recognition system to enter Morse codes. For finger gesture recognition, we use our finger to tap and slide on the table to represent Morse code “dots” and “dashes”.

Question 1: Can single camera-based gesture recognition be achieved?

Question 2: Is machine learning better than the traditional methods?

In this thesis, the experiment is to create a system which has the function to recognise what kind of Morse code a human finger can type. All the questions will be answered during our research.

1.3 Objective of this Thesis

Firstly, as well as the goal of this thesis is to find a suitable method so as to achieve the tracking of finger recognition, the key factor is to find a way to get a single camera to identify moving fingers. In general, many researchers use the in-depth dual cameras to segment the object from the image; the purpose of this thesis is to find a suitable algorithm for segmenting the fingers in the image.

Secondly, to achieve finger gesture recognition, the overall objective of this thesis is divided into four different parts: finger segmentation and recognition, behaviour detection, behaviour recognition, and gesture classification. In this thesis, we are going to work for two parts, which are finger segmentation and recognition.

Finally, this thesis learned in-depth a variety of segmentation algorithms which can achieve this goal; we will find a suitable algorithm to detect the accuracy of our finger gesture recognition.

1.4 Structure of this Thesis

This thesis contains six chapters in total. In the first chapter, the background of gesture recognition is introduced and the evolution of gesture recognition is featured. Motivation and objective are also introduced in this chapter.

In Chapter 2, there is a comprehensive literature review of vision-based gesture recognition. A great deal of techniques are introduced, including some segmentation algorithms, classification algorithms and learning algorithms. The closed-value segmentation algorithm, the region growing algorithm and the related colour space analysis are also introduced.

Chapter 3 introduces the research methodology. In addition, potential solutions and answers are also presented. Moreover, the experimental layout and design, as well as the data set, implementation and evaluation methods will be introduced.

In Chapter 4, the methods and algorithms proposed by this thesis will be implemented. The specific experimental environment is explained, selecting algorithms, as well as the realisation and results of the experiment. In addition, the experimental results and findings will be described in detail with the support of tables and figures.

In the Chapter 5, analysis and discussion are figured, based on the experimental results and findings obtained in Chapter 4.

Finally, Chapter 6 contains the conclusion and future work. In this chapter, we will draw a conclusion and give our expectations for the future.

1.5 Novelty of the Vision-based Morse Code Recognition

In this thesis, the most novel part is that we have combined the old communication method of Morse code with the single camera-based finger gesture recognition technology. The Morse code entered by a finger is identified by single cameras and displayed in the resulting image. People can understand the meaning of Morse code sent in their special environment without needing the help of people to decode.

Chapter 2 Literature Review

2.1 Introduction

Smart cameras are becoming popular in human-computer interaction field nowadays. It allows people to interact with a computer mutually by using natural communications (Ham & Shi, 2009). Finger gestures includes static gestures and dynamic gestures (Erden & Cetin, 2015). For recording all the gestures, there are a few different types of cameras which can be used for gesture recognition. There is the Kinect-depth camera, the stereo camera, Pyroelectric Infrared 3D camera and a normal web camera (Luo & Ohya, 2010), which can all be used for gesture recognition.

2.2 Segmentation Algorithms

2.1.1 Threshold Image Segmentation Algorithm

A thresholding algorithm is a traditional image segmentation method. It depends on whether it uses local information or the global information of a video frame, which can be divided into a non-contextual (also called Point-Dependent) method and a contextual (also called region-dependent) Method. Whether the image uses a uniform threshold or uses a different threshold for different regions, it can be divided into global thresholding or Local thresholding, also known as the adaptive thresholding (Zhang, 2002).

The fundamental principle is to set different characteristic thresholds, dividing the image into pixels of several classes (Billon, Nedelec & Tisseau, 2008). Broadly used features include intensity-based colour features of the original image. The original image $f(x, y)$ is set to find a threshold t in $f(x, y)$, segmenting the image into two parts after the image segmentation, the equation is

$$g(x, y) = \begin{cases} b_0 & f(x, y) < t. \\ b_1 & f(x, y) \geq t. \end{cases} \quad (2.1)$$

where $b_0 = 0$ (black), $b_1 = 1$ (white) refers to image binarization.

In general, a threshold can be regarded as a function of pixel intensity of an image. $f(x, y)$ is the intensity of pixel at (x, y) , $N(x, y)$ is the neighbourhood pixels of the pixel at (x, y) .

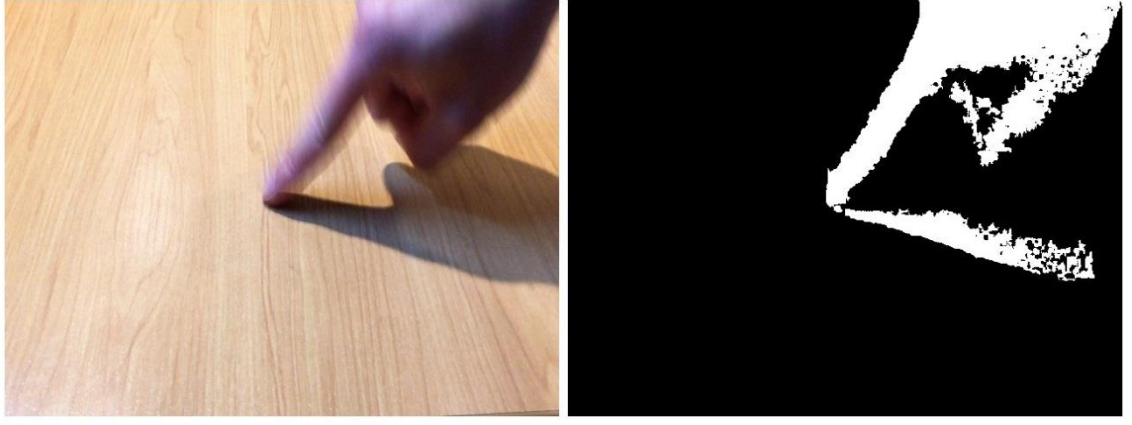


Figure 2.1 Threshold segmentation

2.1.2 Active Contour Model Algorithm

An active contour model is also known as a “Smart Snakes algorithm”, it is a classical algorithm for locating static images (Cootes & Taylor, 1992). The model considers that the contours of each region in the image should be smooth curves and the energy of each contour composed of internal energy. The internal energy characterises the smooth constraints of the contours, this model merges the three phases of the segmentation process so that the final target boundary is a smooth curve. The goal of optimisation is to find a minimised energy function during which the process from the initial position towards the true contour seeks the local minimum. This is done by dynamic optimisation of the energy function by approaching the actual object contour so that the edge-extraction problem merges into the optimisation problem. This kind of algorithm needs to give the initial contour and then iterate to make the snake approach in the direction of energy reduction (Deepak, Nayak & Manikantan, 2016). Finally, an optimal boundary is obtained. When the target is more complex or closer to other objects, the initial snake is not easy to find and the result of iteration often cannot meet the requirements. The active contour model gives the active contour $v(s) = (x(s), y(s))$ of the parameters, energy function E_{total} is expressed as

$$E_{total}(v) = E_{int}(v) + E_{ext}(v) \quad (2.2)$$

$$E_{int}(v) = E_{cont}(v) + E_{curv}(v) = \int_0^1 (a(s)|v'(s)|^2 + \beta(s)|v''(s)|^2) ds \quad (2.3)$$

$$v'(s) = \frac{\partial v(s)}{\partial s}, v''(s) = \frac{\partial^2 v(s)}{\partial^2 s} \quad (2.4)$$

where internal energy $E_{int}(v(s))$ defines an extensible and bendable profile $v(s)$ internal deformation energy, includes continuous energy E_{count} and curvature energy E_{curv} . The first order coefficient α is used to limit the distance on the snake, so that the two points cannot be too far or too close. The second order coefficient β as a function of the angle formed by a point between two neighbour point controls the stiffness of the contour.

External energy $E_{ext}(v)$ is used to extract the image features, including the energy produced by the image force E_{image} and E_{sanc} which represents the energy generated by the external constraint force. The image force indicates that the contour points coincide with the local features of the image; the constraints are normally set to zero.

$$E_{ext}(v) = E_{image} + E_{sanc} = \int_0^1 \gamma(s)p(v(s))ds + E_{sanc} \quad (2.5)$$

where $P(v)$ is a scalar function defined over the entire image surface $I(x, y)$, if $P(x, y) = \nabla(I(x, y))$, then the snake will be attracted to the edge of the image. This technique is applied to gesture recognition in 1995, which the point distribution model is used to achieve the object tracking (Cootes, Taylor, Cooper & Graham, 1995).

2.1.3 Boundary-Based Segmentation Algorithms

To overcome some of the limitations of region-based methods for classification and segmentation, boundary-based methods are often used to look for explicit or implicit boundaries between regions which correspond to different tissue types (Popescu, Lancu, Brezovan & Burdescu, 2010). A variety of classic edge detection algorithms have been proposed such as a Laplace operator (Feynman, Leighton & Sands, 1970), a LoG operator (Marr & Hildreth, 1980), a Sobel operator (Gonzalez & Wood, 2005) and a Canny operator (Canny, 1986).

The pixel intensity of the image boundary changes more sharply, where the boundary-based segmentation could be used to detect the edge between regions by using the characteristic of discontinuing pixels in different regions. The simplest edge detection method is the differential operator method, which uses the nature of the characteristics of discontinuing the intensity, and applies the first or second derivatives to detect the edge

points. Generally, the first derivatives are the gradient operator, e.g. the Prewitt operator or the Sobel operator; the second derivatives are Laplace operator, Kirsch operator, or Willis operator, etc. the gradient operators are sensitive not only to image edges, sensitive but also to image noises.

2.1.4 Region Growing Segmentation Algorithm

The region-growing algorithm first finds a seed pixel for each region to be segmented as the starting point for growth. It then merges the pixels in the neighbourhood of the seed pixel that has similarities to the seed pixel and then to the seed-pixel set, and so on until no more pixels are available. These are then merged and an area is formed (Tremeau & Borel, 1997). Obviously, the seed pixels, growth criteria and termination conditions are critical to the algorithm. The follows focus on the two types of region growing algorithms.

2.1.4.1 Skin colour region growing segmentation algorithm

The segmentation needs to consider the colour of the object itself. While the colour information itself is represented by a two-dimensional quantity, so that HSV (Hue, Saturation, and Value)-based analysis can be carried out in the chromaticity space (Narkhede & Gokhale, 2015). A vector (H, S, P) represents a three-dimensional space, where P stand for the current colour, which can be expressed as a 2D function skin colour in a single image.

For each pixel on the detected image, probability of the skin colour is obtained

$$P_{s_0}(x, y) = M_s(h(x, y), S(x, y)) \quad (2.6)$$

where (x, y) is the coordinates of the pixel, $h(x, y)$ and $s(x, y)$ are the pixels h and s . A region growing algorithm is used, the results obtained from the i frame are as

$$P'_{si}(x, y) = \text{Max}_{dx, dy} \{M_E(\Delta h(x, y), \Delta s(x, y)) \cdot P_{s(i-1)}(x + dx, y + dy)\} \quad (2.7)$$

$$P_{si}(x, y) = \text{Max}\{P'_{si}(x, y), P_{s(i-1)}(x, y)\} \quad (2.8)$$

where $M_E(\Delta h(x, y), \Delta s(x, y))$ is the probability skin growth model, it records the probability that a spot can grow from the original skin colour area. By multiplying the

probability of growing with the probability of the original skin colour, it will obtain the probability of growing the spot.

In the skin colour image, the probability of each point will affect the surrounding points. Similar to the method of generating regional growth, if the probability of the point is colossal, and the probability of its growth adjacent point is also enormous, the probability of multiplication is also significant. Growth can be carried out in four directions or eight directions, and it can grow from the most probable point in these instructions. This is the function $Max(\cdot)$ in Eq.(2.8).

We integrate the growth probability and the original probability after growing the skin colour probability image – this is obtained by finding the two maximum probabilities. Finally, in order to determine the conditions for the end of growth using the classical method, if the new growing points are not generated it means the growing is over, but this method does not determine whether there is a new growth point.

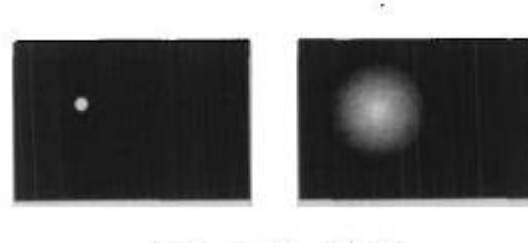


Figure 2.2 Region growing

If the probability of only one original skin colour (x_0, y_0) is 100% and the other is zero, while the growing probability of the same point is $q(0 \leq q \leq 1)$, then the increased result is centred on (x_0, y_0) , which will expand around index q . It stops growing at the edge of the image. In practice, it can be assumed that if there is no new probability of more than 50% of the points, it can be considered as the end of growing.

2.1.4.2 ROI Region Growing Algorithm

In image analysis, the Region of Interest (ROI) is the most interesting and the most representative of the image content. If these regions can be extracted, the efficiency and accuracy of image processing and analysis will be significantly improved (Liu & Fan, 2003).

(a) ROI seed point selection and parameter determination

The seed point should be a point that attracts more attention and its position should be inside the object. Therefore, the seed point should have the largest significant degree and the smallest relative position.

Assume that P is a pixel in the image, R represents the area where $\lambda \times \lambda$ is the centre of P , RPD shows relative bits, and S to show saliency. μ_{RPD} represents the relative position mean of all points in region R , σ_{RPD} stands for its standard deviation. Similarly, μ_s represents the mean value of all points in area R , σ_s shows the standard deviation, the point of the region R is measured as

$$A_R = \exp^{-(\mu_{RPD} + \sigma_{RPD})} - \exp^{-(\mu_s - \sigma_s)} \quad (2.9)$$

where the greater value A_r , the greater the attention focused on the area, the closer the location is to the centre of the area. Therefore, the corresponding area is more preferably selected as the seed region. To reduce the impact of noise, select $\lambda \times \lambda$ small area as the seed point. If λ equals 5, considering both calculation and effect, the colour sub-region of feature describes an averaging colour method.

$$F'_{colour} = \frac{1}{\sigma_{colour}} (F_{colour} - \overline{F_{colour}}) \quad (2.10)$$

where $\overline{F_{colour}}$ is the colour mean of the sub-region, σ_{colour} is the variance, binarized to [0,1] range. The texture of the sub-region is described by the method of grey-level co-occurrence matrices.

To make the segmentation algorithm adaptively change the colour and texture features in the segmentation, we use two types of linear weighted combinations as follow

$$F_{Rogin} = \lambda_{colour} F'_{Rcolour} + \lambda_{texture} F'_{Rtexture} \quad (2.11)$$

where weight $\lambda_{colour} \geq 0 + \lambda_{texdture} \geq 0 = 1$. $F'_{Rcolour}$ and $F'_{Rtexture}$ are the binarized colour and texture features in each sub-region.

(b) ROI region growing

The ROI region growing method based on pixels is affected by noise and the growth rate is slow. We select the appropriate area length; and for each region check the similarity of

the attention value, relative position, colour, and texture (Wang, Cheng & Huang, 2007).

The steps are as follows:

Step 1. Select the starting seed region R_0 ; and obtain the coloured texture feature for the selection region F_0 , attention value S_0 , and representative position description RPD_0 .

Step 2. To create “up” “down” “left” and “right”, four sub-regions adjacent to the seed region, initialise the set of candidate sub-regions record as $C = \{Candidate\ sub-regions\}$.

Step 3. If candidate sub-regions C is not empty, then one candidate sub-region R_1 is selected from C to obtain the eigenvectors F_1 , attention value S_1 , and representative position description RPD_1 .

Step 4. The difference between the colour texture features of the seed region and the starting region is calculated by using $\Delta F = |F_0 - F_1|$.

Step 5. If satisfied $(S_1 \geq T'_s \text{ and } RPD_1 \leq T'_{RPD}) \text{ or } [(S_1 < T'_s \text{ and } RPD_1 \leq T'_{RPD}) \text{ and } \Delta F \leq F']$ the candidate sub-region is a labelled area. This is a larger attention area, the region is not at the edge, or a small attention area are not at the edge but has similar colour texture features to the marked area. To create more “up” “down” “left” “right” four adjacent sub-regions, add new candidate sub-regions into the candidate set C (already present candidate sub-regions in the set are no longer incorporated).

Step 6. If the condition is not satisfied, then remove from the candidate sub-region, then go back to step three.

Step 7. If the candidate sub regions are empty, then process merged regions.

Step 8. Return.

It is not easy to select the seeds in the regional growth method; some attempts have been made to determine seed points by edge detection; however, due to the lack of the edge detection algorithm itself, it cannot avoid missing important seed points (Lee & Liew, 2015).

The advantage of a ROI region growing algorithm is that they are easy to calculate, effectively eliminate the interference of isolated noise and it has strong robustness. The disadvantage of a ROI region growing algorithm is the need for artificial interaction

to obtain seed points so that the user must extract out of each area needing to implant a seed point, sensitive to noises.

2.3 Classification Algorithms

2.3.1 Principal component analysis and derived algorithms

Principal component analysis (PCA) is a statistical method which reduces the dimension of a dataset with a great quantity of related variables so on to keep the corresponding changes (Anthony, Hines, Barham & Taylor, 1990). The original dataset is transformed by calculating the eigenvalues of the eigenvectors and the set of covariance matrices.

The actual processing of the first few main components selected achieves the purpose of reducing dimensions. The analysis is (a) standardization of data indicators; (b) index correlation determination; (c) determine the number of principal components m ; (d) principal component F expression; (e) principal component F is named.

Principal component analysis can be used for: reducing the feature space dimension, determining the linear combination of variables, selecting the most useful variables and variable identification, and recognition of target or group of outliers and so on. The principal component subspace provides data compression from high dimensional data to low-dimension data in the sense of the mean square error which minimises variances.

Sirovich (Sirovich & Kirby, 1987) and others, have attempted aspects of face recognition which is the earliest application of the algorithm to the field of computer vision. Later it was introduced into the field of vision-based gesture recognition, when Birk and his colleagues achieved a static international sign language, i.e. hand recognition (Birk, Moeslund & Madsen, 1997) (Jerome & Crowley, 1997). Principal component analysis is sensitive to position, orientation and the target objects in the image. The advantage of principal component analysis is that it can recognise more of the hand shape. The disadvantage of principal component analysis is the need to go through more than one person's training to achieve the accuracy and independence of identification. The image needs to be normalised to keep it consistent.

2.3.2 A linear scaling model for the fingertip

The model assumes that the movement of most fingers is linear and has a small amount of joint rotation. First, we define the state vector $x_t = (x(t), y(t), v_x(t), v_y(t))^T$, then mark the fingertip position and the tip movement speed at frame T . Observation vector Y is defined as the fingertip position detected at the time of the frame T . The relationship between the state vector X and the observation vector Y is

$$X_{t+1} = FX_t + GW_t \quad (2.12)$$

$$Y_t = HX_t + V_t \quad (2.13)$$

where F , G , and H are the state transition matrix, the driving matrix and the observation matrix. Shah used this model for the first time in vision-based gesture recognition, whereas Davis used a glove with a brightness sign to enhance recognition (Davis & Shah, 1993). Shah used histogram segmentation to extract the position of the fingertip. The system records a series of motion data as part of the model, including the gesture name, the direction and the absolute value of the vector for each finger, and the gesture is matched if all the finger directions coincide with the absolute value of the vector. The system achieved at least a 90% accuracy rate. The advantage of this model is high accuracy and is easy to implement also, it only records the starting point and the end of the finger. The disadvantage of this model that it is difficult to capture the fingers of the non-linear movement. When more gestures appear, the system will cause problems.

2.4 Recognition Algorithms

Gesture recognition in the learning algorithm involves the field of artificial intelligence. The main characteristic of most algorithms is that with the increase of training intensity, the accuracy also increases, which is divided into a single algorithm and compound algorithm.

2.4.1 Instance-based Learning

This method comes from machine learning. The main difference between this method and other learning algorithms is the different ways of training data (Chan, 2000). This method simply stores the training examples; this instance-based learning is also known as lazy

learning. Whenever the learner encounters a new query instance, it analyses the relationship between this new instance and previously stored instances; a new instance is assigned to an objective function value (Shi, Taib & Lichman, 2006).

For example in gesture recognition, the feature vector is the position and orientation of the hand and the curvature of each finger. In supervised learning, when the training data passes through the nodes, the weights of the nodes are automatically updated according to the dataset. While in instance-based learning, the training data is only used as a database to classify other instances. Examples may appear in points of Euclidean space, such as the k -Nearest Neighbours Algorithm, which is mentioned in the literature (Mitchell, 1997).

It is assumed that all instances correspond to points R^n in the n -dimensional space, where an arbitrary instance is represented as a feature vector $\langle a_1(x), \dots, a_n(x) \rangle$, the distance between two instances x_i and x_j is defined as

$$d(x_i, x_j) = \sqrt{\sum_{r=1}^n (a_r(x_i) - a_r(x_j))^2} \quad (2.14)$$

where discrete objective function $f: R^n \rightarrow V, V = \{v_1, \dots, v_s\}$, approximation of discrete valued function $f: R^n \rightarrow V$ k -Nearest Neighbor is

- Training algorithm: Take each training sample $\langle x, f(x) \rangle$ and add to list *training_examples*
- Testing algorithm

We give a query instance to be classified x_a , select instance k in *training_examples* which is closest to x_a , and use $x_1 \dots x_k$ to show

$$\delta(a, b) = \begin{cases} 1 & a=b \\ 0 & a \neq b \end{cases} \quad (2.15)$$

where the return value is an estimate of $f(x_a)$, which is the most common value f among the k -th training samples closest to x_a , the result is related to the value k (Suarez & Murphy, 2012). Another form of instance-based learning is event-based reasoning, a description of the mechanism part. A key difference in instance-based learning approaches is that different object function approximations can be established for different instances of the query to be classified, compared to other methods (Berci. &

Szolgay, 2009). Many techniques do not establish an approximation of the objective function over the entire instance space, only the local approximation is established and used in instances close to the new instance. The advantage is that the objective function is sometimes complex, description (Cao, Lu, Gu, Peng & Wang, 2004).

2.4.2 Hidden Markov Model

The Markov chain is a simple finite state control system. There is a certain probability of interrelationship between each state transitioning to another state, the sum of all probabilities for one state transition to another state is one (Kim, Kim, & Lee, 2002). A hidden Markov Model is regarded as a general form without Constraint Markov Chains (Charniak, 1993). Since there is more than one migration curve for a given output, so the result is uncertain, the state matrix of the input sets cannot be determined directly from the output. Markov chain and conditional probability are inseparable.

Assuming that the state space is a non-negative integer (0,1,2,...), for discrete random sequences X_n, X_{n+1} the probability in state j , only with the previous state X_n which is called a Markov chain. Remember the probability of this step is

$$P_{ij}^{n,n+1} = P(X_{n+1} = j | X_n = i) \quad (2.16)$$

The above definition not only indicates that the transition probability is not related to the initial state and the final state, but is also related to the time series n . When the probability of transition is independent of the time series, it is called a smooth transition probability. Since most of the models are related to the probability of smooth transition, therefore, in the case of not specified, it is the default and is smooth.

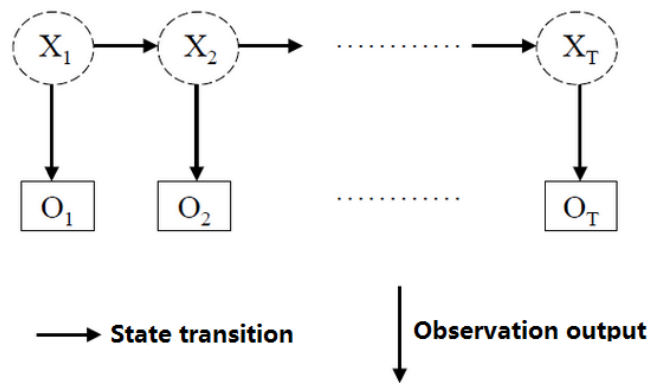


Figure 2.3 Hidden Markov model

As a widely used statistical method, the Hidden Markov Model under general topology has a strong description of the gesture signal time and space variations. It has been a dominant position in the field of dynamic gesture recognition. Liang's team used the Hidden Markov Model to analyse the gestures within the interval (Liang & Ouhyoung, 1996). Schlenszig has applied the Hidden Markov Model to vision-based gesture recognition (Schlenszig, Hunter, & Jain, 1995). Grobel and Assam used the Hidden Markov Model to identify the 266 isolated words that the user wearing the colour-coded glove inputs from the camera – the correct rate reached up to 91.3% (Grobel & Assan, 1997). However, due to the generality of HMM topology, this model is too complex to analyse sign language signals, which makes the HMM training set and recognition too complex overall, especially in continuous HMM.

2.4.3 Artificial Neural Networks

An artificial neural network is a system which imitates the operation of human brain to process information. The artificial neural network uses the node as the necessary element of the operation, where the nodes are linked together and the corresponding weight rates each link. Murakami's system is one of the earliest systems that used the artificial neural network in hand gesture recognition (Murakami & Taguchi, 1991). He is using a three-layer neural network, which includes 13 input nodes, 100 hidden nodes and 42 output nodes. After the initial training, 77% accuracy is achieved by using back propagation in the neural network, such as in Figure 2.1. Fully utilised in the vision-based approach is the Banarse and others they used to carry out the biometric network; that is a visual cortex of the spatial recognition system (Russell & Norvig, 1995). Artificial neural networks have the characteristic of classification and anti-interference; however, due to its weak ability to deal with time sequences, it is used more on static hand gesture recognition.

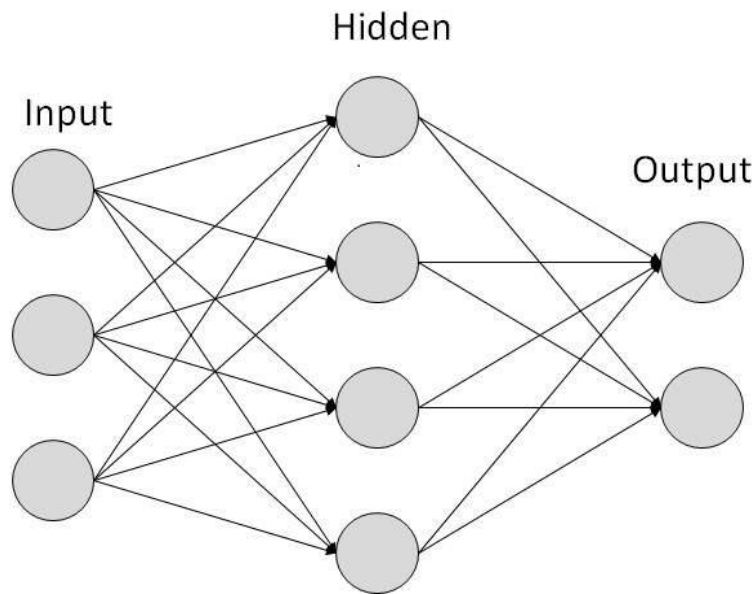


Figure 2.4 Neural network model

The GloveTalk system designed by Fels uses the neural network method as recognition technology (Fels & Hinton, 1993). After the neural network algorithm was put forward in 1988, many deformations appeared, including the replacement error function, the dynamic adjustment of the network topology, the learning rate and the dynamic adjustment of the factor parameters (Adachi, Furuya, Greene & Mikuriya, 1991). In recent years, more and more attention has been paid to extract rules from neural networks, mainly in the following two ways:

- Rule extraction of network structure decomposition
- Extracted by the nonlinear mapping relation of the neural network

The future development of neural networks can further reduce the complexity of the algorithm to improve the extractability of the rules which can be understood and the applicability of the direction of the algorithm (RongQing, WenHui, Duo & Hua, 2008).

2.4.4 Support Vector Machine

The support vector machine is a new learning machine based on statistical learning. The SVM algorithm is a nonlinear generalisation of the generalised description algorithm developed in Russia in the 1960s (Vapnik, 1963). On the question of when to start tracking, there has been a series of research progress, by adding a certain range of colour, shape and movement to achieve it (Pavlovic, Sharma, & Huang, 1997) (Sharma, Pavlovic

& Huang, 1998) (Wren, Azarbayejani, Darrell & Pentland, 1997) (Azoz, Devi & Sharma, 1998).

The standard SVM training algorithm is developed from pattern recognition (Copot, Syafiie, Vargas, Keyser, Langenhove & Lazar, 2009). Training in SVM is done through the database; after training it is used to classify the hyperplane (Negri, Santanna & Dutra, 2013). For solving practical problems, we hope that in a period of time which has been divided into classes that they will not be changed, but in the tracking problem there cannot be such assumptions. In the three-dimensional case, the lighting conditions will also change using colour, shape and motion as a vector feature constructing the hyperplane; the support vector will also change with these characteristics.

Kalman filtering is used to track the moving target, to update the support vector, and then to reclassify the new support vector to obtain a new hyperplane (Anderson & Moore, 1979). Once the classification is determined, a new set of support vectors are obtained and accumulated. In the binary classification, independent on the probability distribution of the data and the expected risk of existence of the upper bound, we find the Eq. as 17 (Vapnik, 1995).

$$R(a) \leq R_{emp}(a) + \sqrt{\frac{h(\log(2l|h|)+1-\log(n/4))}{l}} \quad (2.17)$$

Simultaneously,

$$R(a) = \int \frac{1}{2} |d - f(x, a)| df(x, d) \quad (2.18)$$

$$R_{emp}(a) = \frac{1}{2l} \sum_{i=1}^l |d_i - f(x_i, a)| \quad (2.19)$$

where R_{emp} is the empirical risk given by using Eq. (2.17), $R(a)$ is the actual risk, and $P(x, d)$ is the unknown probability distribution of the acquired data. $f(x, a)$ is mapped from x to d . The Vapnik-Chervonenkis dimension of a set of functions $f(a)$ is defined as the maximum number of possible training points affected by using $f(a)$.

2.4.5 Convolutional Neural Network

A convolution neural network is a type of neural network, but its structure is different from others (Triantafyllidou & Tefas, 2016). The data of each layer is not composed of

several vector nodes, but consists of several matrix data nodes. In the calculation it is used in the form of convolution, it is consequently named a convolution neural network. CNN has become a research hotspot model in the field of image recognition. Its weight-sharing features are similar to biological-optic neural networks. This is a feature that greatly reduces the complexity of the network model on the grounds that weight sharing greatly reduces the number of weights, thereby reducing the local minimum. A convolution neural network can make the original image directly as the input of the network, rather than other neural networks which need to vectorise the original image before the network. This approach however, avoids the traditional identification algorithm in the complex feature extraction and data reconstruction process (Wang, Li, Liu, Zhang, Gao & Ogunbona, 2016). The network has a high degree of invariance in scaling, translating or rotating the image. The down-sampling of the convolutional neural network is a big bright spot, which reduces the computational complexity. With the increase of the model network layer, it can extract more abstract information. It is not only used for visual neural networks but also for acoustic neural networks and produces great social values (Kim, Lee, & Park, 2008).

Convolution neural networks contain three structural characteristics to ensure that invariance of translation, rotation, and scale transformation invariance, which are either local field, weight sharing, space or time down sampling. Local field neurons can extract basic visual features such as edges, endpoints, or corners. These features are then connected to subsequent layers to detect higher-order feature combinations. They had weight shared network parameters so that the number decreased a lot, which helps to find a better local lowest point during training. Subsampling can reduce the calculation time to build deeper abstract information, and highlight characteristics of information; subsampling also helps to suppress noise. a LeNet-5 structure convolutional neural network for identification (Cun, Bottou & Haffner, 1988).

LeNet-5 network includes 7-layers (not including the input layer), so the parameters of network layer can be trained. The input layer is an image of 32×32 pixels – the more pixels of the input image, the more information available, without changing the convolution kernel and the subsampling rate under the premise that it will increase the

corresponding network layer. When fed into the network for training, it needs to make a normalised input image process which is conducive to improving the learning convergence rate.

Due to the great attraction of deep learning, many researchers have proposed different structures of convolution neural networks. The CNN structure proposed by Sun et al. at the (Sun, Chen, Wang & Tang, 2014). It has been used for identification and has achieved excellent results; in the LFW database to obtain an accuracy rate of 99.15%. The deep CNN network structure used by the Hinton in the 2012 ImageNet contest (Krizhevsky, Sutskever & Hinton, 2012). In 2013, they improved the performance of the network, expanded its size and the number of training samples. The front part of the 3D convolution neural network proposed by Ji et al. (Ji, Xu, Yang & Yu, 2012), which adds time to the front end. Convolution neural networks which are spatial 2D networks; 2D convolution neural networks are usually only identified for single image classification, but the structure can identify the sequence of images.

2.5 Relevant Colour Space Analysis

A colour space first needs to be selected for skin colour, spectral characteristics and electromagnetic radiation relating to the perception of the phenomenon, based on colour-based gesture segmentation. Since the purpose of this pursuit can be any aspect from efficiency to light independence, there are many classification methods to measure it. Current thinking indicates that the colour space has perceptual linearity. In any case, if the pixel unit changes, it will immediately cause a change in perception. the colour space is intuitive, while others are relatively abstract with the colour contact. Alternatively, the colour space is dependent on the equipment, where they only show up while using specific equipment such as cameras, monitors or other devices. In the experiment, they used two components – intensity and brightness, the colour space used in the pre-treatment stage of skin segmentation are HSV, normalised RGB, easy RGB, YUV and improved CIExyz, colour signal and motion signals are also used. (Tairi Z. H., Rahmat R. W., Saripan M. Q. & Sulaiman P. S. 2014)

2.5.1 Two-dimensional Gaussian Distribution

In the normalised colour space, the colour distribution of the hand area is concentrated in a small area. Although different skin colour appears to fluctuate within a wide range, in the normalised colour space it is not very different. In other words, different skin colours are similar to the main difference which depends on the strength of information. However, due to changes in light, a small change in colour distribution may still occur in the same person. To solve this problem, an ordinary skin colour distribution is defined, which includes a large range of variables, including possible changes under different lighting conditions (Kasson & Plouffe, 1992). This distribution can be modelled using a 2D Gaussian distribution.

The input colour gesture image can be converted into an image Z where the specific colour region is treated with the GSCD. When the input pixel colour is close to the central region of the Gaussian distribution, we can make the output pixel in the output image Z have higher intensity (Jia, Jiang & Wang, 2008). The colour transformation is as follows:

$$Z(x, y) = G(r(x, y), g(x, y)) = \frac{1}{2\pi\sigma_r\sigma_g} \exp\left[-\frac{1}{2}\left\{\left(\frac{r(x, y)-m_r}{\sigma_r}\right)^2 + \left(\frac{g(x, y)-m_g}{\sigma_g}\right)^2\right\}\right] \quad (2.20)$$

where (x, y) is a pixel. Both $g(x, y)$ and $r(x, y)$ are the corresponding pixel red and green elements normalised colour values. $G(\cdot)$ is a two-dimensional Gaussian function. σ_r and σ_g are the standard deviation for red and green elements.

2.5.2 YCbCr Colour Space

In colorology, the commonly used colour spaces are RGB, HSV, YUV, YCbCr, HIS, CMY and so on. One of the most widely used colour spaces is RGB; it uses three basic colours of red, green and blue to define most colours and it can cover a large colour gamut (Amma, Yaguchi, Niitsuma, Matsuzaki & Oka, 2013). But the colour space is rarely used in the field of scientific research because it presents tone, saturation, and brightness and is not conducive to variable research.

Using classical colour spaces, we employ YCbCr instead of the RGB space to separate the finger region effectively from an image with a variable background (Chelali, Cherabit & Djeradi, 2015). The YCbCr colour space has characteristics of chrominance and

luminance separation, and is good for clustering properties of skin colour. It is not much influenced by the variation in brightness and can easily distinguish the complexion region. According to data from the researcher Noda et al. (Noda, Niimi & Korekuni, 2006) we distinguish the human skin colour distribution range in the YCbCr colour space which is approximately: $77 \leq C_b \leq 127, 133 \leq C_r \leq 178$, (data may vary with different skin colouration). We selected this range as the colour segmentation value which is defined as,

$$SkinColour(x, y) = \begin{cases} 1 & \text{if } (77 \leq C_b \leq 127) \\ 1 & \text{if } (133 \leq C_r \leq 178) \\ 0 & \text{otherwise} \end{cases} \quad (2.21)$$

2.5.2.1 Information analysis and gesture recognition

In this stage, contours extracted from the binary hand mask are processed using the convex hull analysis.

$$c(x, y) = \frac{\sum_{m=0}^n p_m(x, y)}{N} \quad (2.22)$$

where $P_m(x, y)$ then m is the gesture area coordinate value of the first pixel, and N is the total number of pixels in the gesture area obtained in the $C(x, y)$ coordinates of the palm. Through the steps we accurately identify palm and fingertip positioning (Yu, Zhu, Xu, Wen & Ren, 1998).

2.5.2.2 Dynamic gesture recognition

The identification process is to identify the vertical and horizontal by using Eq.(2.22) and Eq. (2.23),

$$(\Delta x, \Delta y) = C_{count-1}(x, y) - C_1(x, y) \quad (2.23)$$

$$\theta = \arctan\left(\frac{\Delta y}{\Delta x}\right) \quad (2.24)$$

where $C_1(x, y)$ and $C_{count-1}(x, y)$ are the coordinates of the first frame and the last frame of the gesture, θ is the angle between two points. This is determined by the value θ where we know the general direction for gesture movement. For fist recognition, this can be defined as consecutive N frames of one finger and consecutive N frames reducing the number of T pixels. Through the steps, we identify seven kinds of gesture information

and thus produce seven different control signals (Zivkovic, Kliger, Kleihorst, Danilin, Schueler, Arturi & Aghajan, 2008).

2.6 Scale-invariant Feature Transform (SIFT)

Scale-invariant feature transform is a computer vision algorithm used to detect and describe the local features in the image. It finds the extreme points in the spatial scale and extracts its position, scale, and rotation invariants. Its application includes object identification, robot map perception and navigation, image stitching, 3D model establishment, gesture recognition, image tracking and action comparison. This algorithm has its patent.

The scale space is used to achieve the use of a Gaussian pyramid. The Gaussian pyramid is divided into two parts: the first Gaussian blur is of different scales on the image, the second samples the image. The pyramid model of the image refers to the constant reduction of the original image, resulting in a series of images of different sizes, from large to small, from bottom to the top of the hierarchical model.

The original image is the first layer of the pyramid; and each time the new image is obtained by sampling the pyramid layer; each pyramid altogether equals n layers. The number of layers of the pyramid is determined by the original size of the image and the size of the top image,

$$n = \log_2\{\min(M, N)\} - t, t \in [0, \log_2\{\min(M, N)\}] \quad (2.25)$$

where N is the size of the original image, t is the logarithm of the minimum dimension of the top image. To make the scale reflect its continuity, the Gaussian pyramid in the simple down-sampling based on the Gaussian filter is referred to in Figure 2.9. The image pyramid of each layer uses different parameters to do a Gaussian blur, so that each pyramid contains multiple blurred images, each layer of the pyramid having multiple images together as a group, whereby the pyramid of each layer is only one group. The number of images, the number of groups and the number of pyramids are equal using the Eq. (2.24); each group contains multiple images. In addition when down-sampling, the initial image (bottom image) of a set of images on the Gaussian pyramid is sampled from the penultimate image of the previous set.

In 2002, Mikolajczyk found the experimental comparison of the normalised Gaussian Laplacian function. For example, a gradient, Hessian or Harris angle feature comparison can produce the most stable image features.

Lindeberg (Lindeberg, 1994) discovered that the Difference of Gaussian (DoG) was very similar to the normalised Gaussian Laplacian function $\sigma^2 \nabla^2 G$ in 1994, where the relationship between $D(x, y, \sigma)$ and $\sigma^2 \nabla^2 G$ can be derived from the following Eq.(2.26)

$$\frac{\partial G}{\partial \sigma} = \sigma \nabla^2 G \quad (2.26)$$

Using differential approximation instead of the differential is:

$$\sigma \nabla^2 G = \frac{\partial G}{\partial \sigma} \approx \frac{G(x, y, k\sigma) - G(x, y, \sigma)}{k\sigma - \sigma} \quad (2.27)$$

So,

$$G(x, y, k\sigma) - G(x, y, \sigma) \approx (k - 1)\sigma^2 \nabla^2 G \quad (2.28)$$

where $k - 1$ is a constant and does not affect the position of the extreme point. In the actual calculation, the Gaussian pyramid is subtracted from adjacent upper and lower layers in each group to obtain a Gaussian difference image, as shown in Figure 2.10 for extreme value detection.

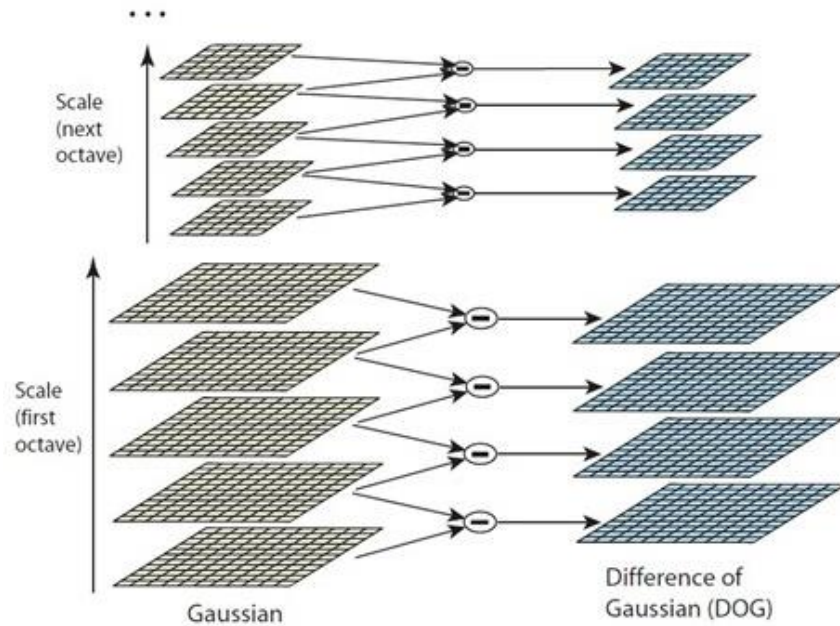


Figure 2.5 The generation of a Gaussian differential pyramid

Chapter 3 Research Methodology

3.1 Introduction

Although Morse code is an old invention, some advanced communications still cannot replace this traditional one. Powerful Morse code can be used in any visual and auditory situations. No matter which algorithm is chosen, given that the computer is not currently able to achieve 100% accuracy in pattern recognition, because the proposed method in this paper is a novel design combining old communication methods with new computer vision technology, the performance of finger gesture recognition must be good enough.

In this project, it is expected we use our finger gestures to input Morse codes, which means when we use our finger tap and slide on a desk top to simulate the short and long signals of Morse code called “dots” and “dashes”, or “dits” and “dahs”, a normal camera will capture the finger gestures. This thesis aims to design a “Morse code reader” with the aforementioned feature. In order to achieve this goal, it is very important that we have a thorough understanding of the literatures after review. In this chapter, we are going to understand finger gesture recognition, the finger gesture recognition will include text, SMS, Emoticons, and SOS urgent symbols based on basic Morse codes. For the recognition and classification, we will use two methods: one is the traditional image segmentation-based algorithm, the other is a machine learning-based algorithm. We will compare two different sets of algorithms; the final experiments will tell the advantages and disadvantages of the image segmentation algorithm and machine learning algorithm.

3.2 Related Work

Multiple technologies based on human gesture recognition have been developed for computer input. A text entry system was developed by using Morse codes with tongue gestures (Sapaico & Sato 2011); therefore, it is possible to “type” a letter by using human tongue gestures; the system gained 84.78% overall accuracy. According to the tests for the disabled, preliminary results shew that the system has usability, usefulness, and universality.

A static-hand recognition of international sign language was implemented by using principal component analysis (PCA). The methods is able to recognise 25 hand gestures

(Birk et al. 1997). The recognition rate is approximately 99% compared to other models; but the gestures must be shown before a black board so as to increase the overall recognition rate.

A linear scaling model was proposed to locate the position of fingertips (Davis & Shah 1993) which analyses a series of motions as one part of the model, including the gesture entity, direction, and absolute value of the feature vector of each finger; the gesture is matched if all the finger directions coincide with the absolute value. The system obtained 90% recognition rate at least. The advantages of this model are its high accuracy and easy implementation. The disadvantage of this model is too difficult to capture the finger movement which is not in a straight line; it also needs to record the starting point and the end one of a finger gesture.

As a widely used statistical method, hidden Markov model (HMM) has a strong ability to describe the gestures spatiotemporally; it has taken its dominant position in the field of dynamic gesture recognition. HMM has been applied to the vision-based gesture recognition (Liang & Ouhyoung 1996); it identified 266 isolated words when a user is wearing the color-coded glove for inputs; the correct rate of recognition was up to 91.3% (Grobel & Assam 1997).

An ASL (American Sign Language) system of the Carnegie Mellon University (CMU) and TSL (Taiwanese Sign Language) of the National Taiwan University (NTU) were both using hidden Markov model as the core technology of finger gesture recognition (Bergerman et al. 1995). However, due to the essence of HMM, this model was too complicated to analyse the sign languages, especially in the case of continuous HMM.

Support vector machine (SVM) is a classical classifier in machine learning (Vapnik 2013). In finger gesture recognition, it was usually applied to decide when to start the finger tracking by using a range of colours, shapes, and motions (Pavlovic et al. 1997). The standard SVM algorithm was developed for pattern classification (Suykens et al. 19989). By using a training dataset, a SVM conducts the classification with its hyperplane (Negri et al. 2013).

Artificial neural networks (ANNs) simulate our human brain in pattern classification which train the network nodes deployed on multiple layers as the necessary elements for finger gesture recognition. ANNs have the characteristics of pattern classification with anti-interference (Russell 1995). An ANN network in gesture recognition has been developed (Murakami & Taguchi 1991) by using a three-layer neural network, including 13 input nodes, 100 hidden nodes, and 42 output nodes. After the initial training, the accuracy 77% was procured by using the back propagation of ANN, for example, GloveTalk was designed by Fels using ANNs as its core technology (Fels et al. 1993).

A real-time hand tracking and gesture recognition system was developed by using ANN. The system can recognize Burmese by using human gestures, the recognition accuracy was 98% overall (Maung 2009). A sign gesture recognition system was developed by using recurrent neural networks (RNNs) in deep learning; the system can recognise 42 symbols. Under the RNN framework, the number of training samples do boost up recognition rate, the rate for registered people is up to 98% as well as that of unregistered people is 77% (Murakami & Hitomi 1991).

After ANN algorithms were put forward to practice, it has been greatly improved and generalized, including replacement of error function, dynamic adjustment of network topology, learning rate, and factor parameters (Murakami et al. 1991). The future development of ANNs can further reduce the complexity to enhance the extractability of ANN training rules and the applicability of the algorithms (adachi et al, 1991).

Different from the existing work, we will develop a prototype for Morse code input by using finger gesture recognition in machine learning and computer vision. Our proposed methods are more intelligent and smarter than those existing ones.

3.3 Data Acquisition

To complete the gesture recognition experiment, we need to collect a lot of data for testing and comparison. We used a camera based on the iPhone as the recording device, and the video output resolution is 1920×1080 with 30 fps , which is currently the mainstream bit rate of a video.

All videos required the testers to input Morse codes by using finger gestures. This includes 26 English letters, 10 numbers, 18 punctuations, five SMS messages, four emoticons, 21 mathematical symbols and Chinese characters from GB2312/80. Table 3.1 to Table 3.5 show the Morse codes for our finger gesture recognition.

3.4 Research Design

3.4.1 Design a symbolic set for Morse codes

In Table 3.1 and Table 3.2, as the standard Morse codes only consist of English letters and numbers, we need to extend the symbols to ensure we have a complete character set for the Morse code-based conversations.

In principle, a Morse code only uses “dis” and “dahs” to transmit signals or messages. This means we have to encode the input using our character set; correspondingly, we define a series of new Morse codes from Table 3.3 to Table 3.5 to encode the inputs by using finger gesture recognition

Table 3.1 Original Morse code table

A	..	K	--	U	...	0	-----
B	L	V-	1	-----
C	M	--	W	...-	2-
D	...-	N	-.	X-	3-
E	.	O	---	Y	----	4
F	P	Z	5
G	...-	Q	----			6
H	R	...-			7
I	..	S	...			8
J	T	-			9

Table 3.2 Punctuations

Period (.)	·-·-·-	Colon (:)	-·-·-·
Question mark (?)	·-·-··	Equal sign (=)	-·-·-·
Exclamation mark (!)	-·-·-·	Hyphen (-)	-·-·-·
Parenthesis (())	-·-·-·	After parenthesis ())	-·-·-·
At (@)	·-·-··	Semicolon (;)	-·-·-·
Double quotes (")	·-·-··	And (&)	· · · ·
Comma (,)	-·-·-·	Dollar (\$)	·-·-·-·
Apostrophe (')	·-·-··	Slash (/)	-·-·-·
Underline (_)	·-·-· ·-	Space	·-·-

Table 3.3 SMS and emoticons





SOS	·-·-·-·-·		·-·-·-·-·
How are you?	·-·-·-·-·		·-·-·-·-·
See you	·-·-·-·-·		·-·-·-·-·
LOL	·-·-·-·-·		·-·-·-·-·

Table 3.4 Morse codes of mathematical symbols

$\sqrt{\quad}$	·-·-·-	\div	·-·-·-	\emptyset	-·-·-·-
\times	-·-·-·	\approx	·-·-·-	α	-·-·-·-
Σ	·-·-·-	\int	-·-·-·-	\exists	·-·-·-·-
α	-·-·-·	β	-·-·-·-	∇	-·-·-·-·-
Π	·-·-·-	\otimes	-·-·-·-	\cup	-·-·-·-
∞	-·-·-·-·-	ω	·-·-·-	\in	·-·-·-·-
\approx	·-·-·-	\angle	·-·-·-	\equiv	-·-·-·-·-

Table 3.5 Morse codes of simplified chinese characters from GB2312/80

	00	01	02	...	87
00		.----- 啊	..----- 阿		-----... 鳎
01	-----. 蔼	.----- 矮	..----- 艾		-----... 鞑
02	-----. 按	.----- 暗	..----- 岸		-----... 骷
...					
87	-----... 璠	.----- 鲙	..----- 鳓		-----... 黠

As the original Morse code table only comes with the above letters and numbers, we need to extend our Morse codes by using the tables and create a bigger character set to ensure that we have a complete dialogue system.

3.4.2 Architecture

3.4.2.1 Image Segmentation Algorithm

In this chapter, we will discuss the architecture to implement finger gesture recognition. To reflect the advantages of machine learning algorithms, two algorithms of Morse code recognition are described in details. The first algorithm is used in the case of the fixed background using coloured space for image segmentation and a binarized image to identify the Morse codes. The second algorithm is based on machine learning, specifically in the case without image segmentation; we only need to mark the interested region, then train the target region to get the trained network, and finally use the network to identify the Morse code.

In the traditional image segmentation, the image will be processed, such as removing some interfering frames and extracting its key image information. To get the information of the key frame, the RGB image will be converted to YCbCr colour space, and only the intensity information is extracted. Then we get segmentation and morphological processing of the binarized images. After that, the function will read the binarized image and analyse the position of each finger gesture and marked it with “+1” or “-1”. After all

the images have been tackled, the information will be saved as a variable. Lastly, decoding the variable and outputting the video will help to complete the function, Figure 3.1 shows the basic flowchart of the finger gesture recognition.

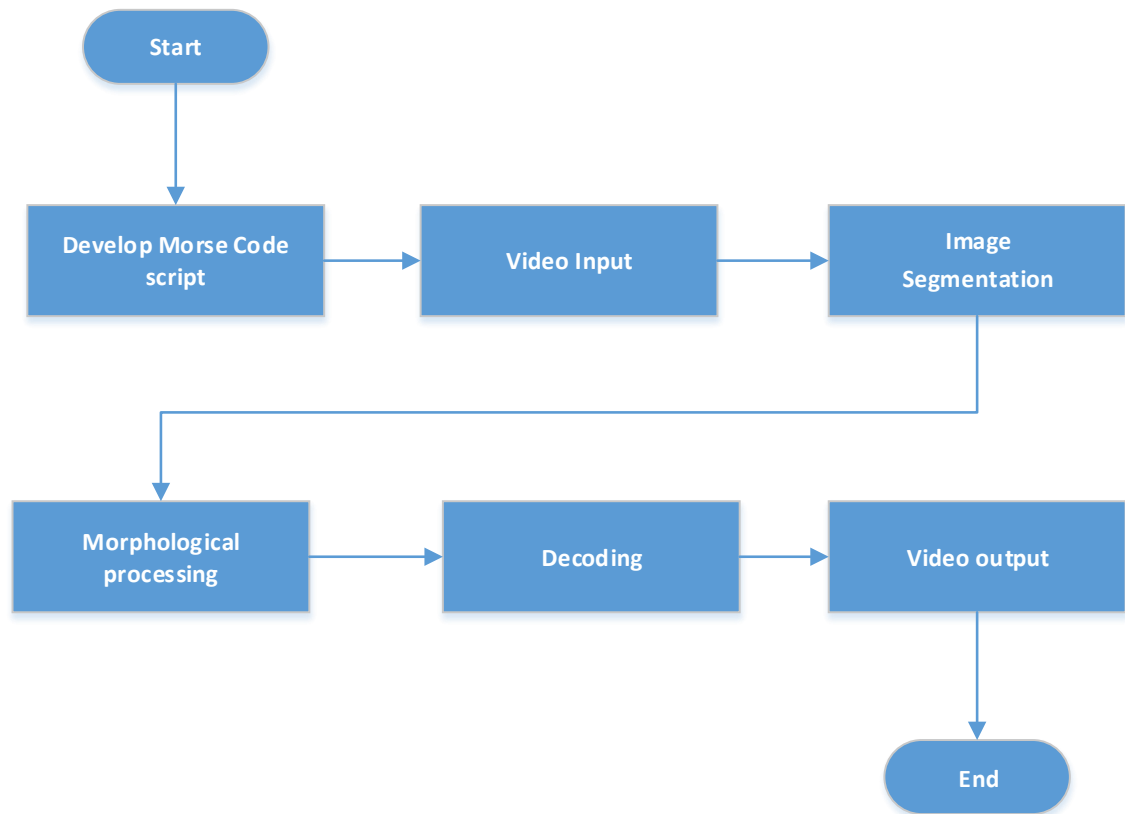


Figure 3.1 Basic flowchart of the finger gesture recognition system based on image segmentation

Figure 3.2 shows that image segmentation has been used, the function will scale each frame to 960×540 , then transfer all RGB image to YCbCr colour space, because YCbCr colour space has the characteristics of chrominance and luminance separation. It is good for clustering skin colour, and blob detection will be used to measure properties of image regions.

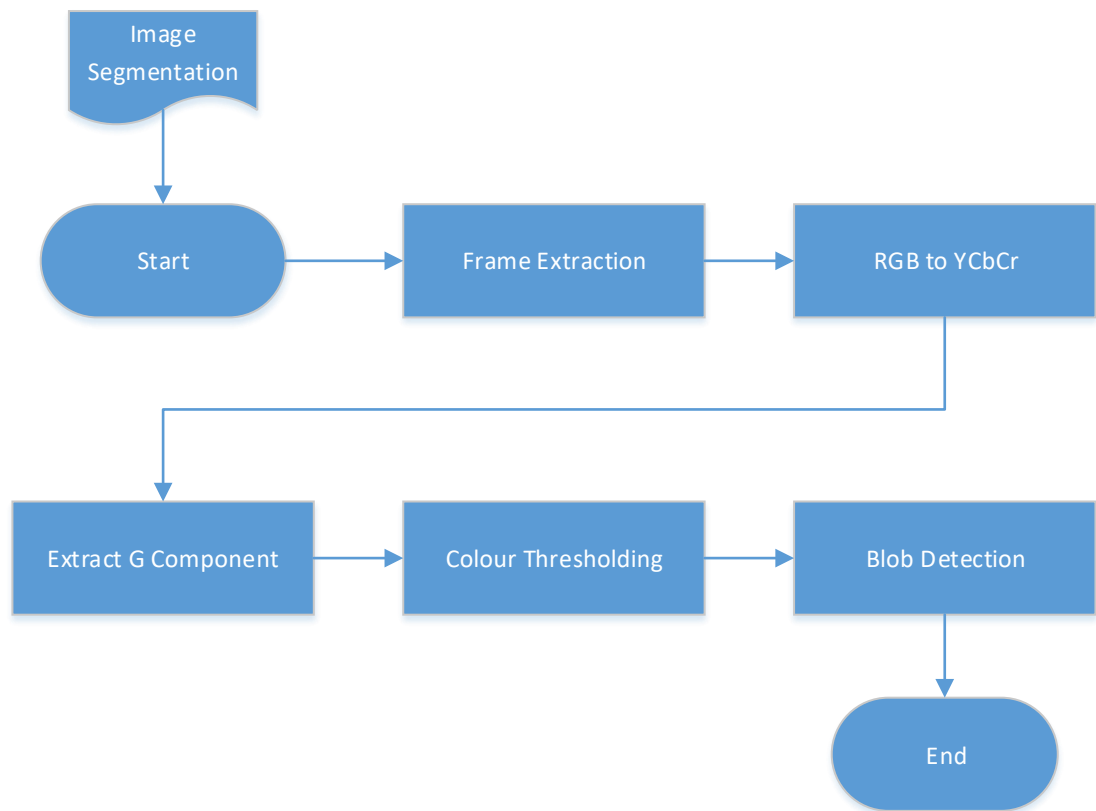


Figure 3.2 Flowchart of image pre-processing

The morphological process will be used to process all binarized images to segment the human hand within the background accurately. The other purpose of this part is to analyse finger gestures and the number of connected regions between the finger and its shadows. Figure 3.3 shows finger gesture recognition and its analysis.

Figure 3.4 shows the decoding process and video output. In the last step, all finger gestures have been tagged into “+1” and “-1”, when they coincide to determine the beginning or end of the Morse code. According to the above principle, by each successive set of the Morse code, they will have a longer waiting time. This goes from the duration to the subsequent Morse codes.

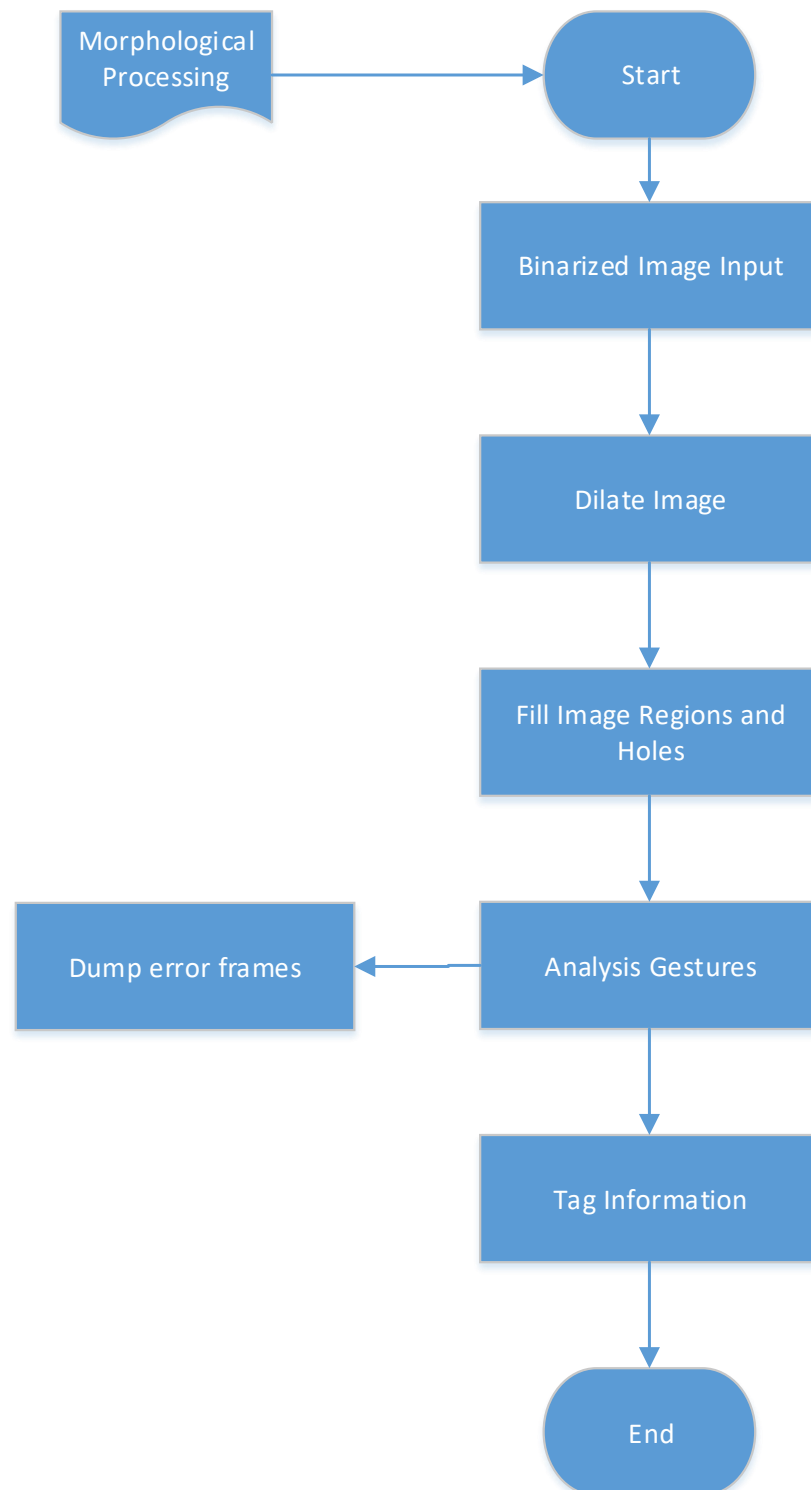


Figure 3.3 The flowchart of finger gesture analysis & recognition

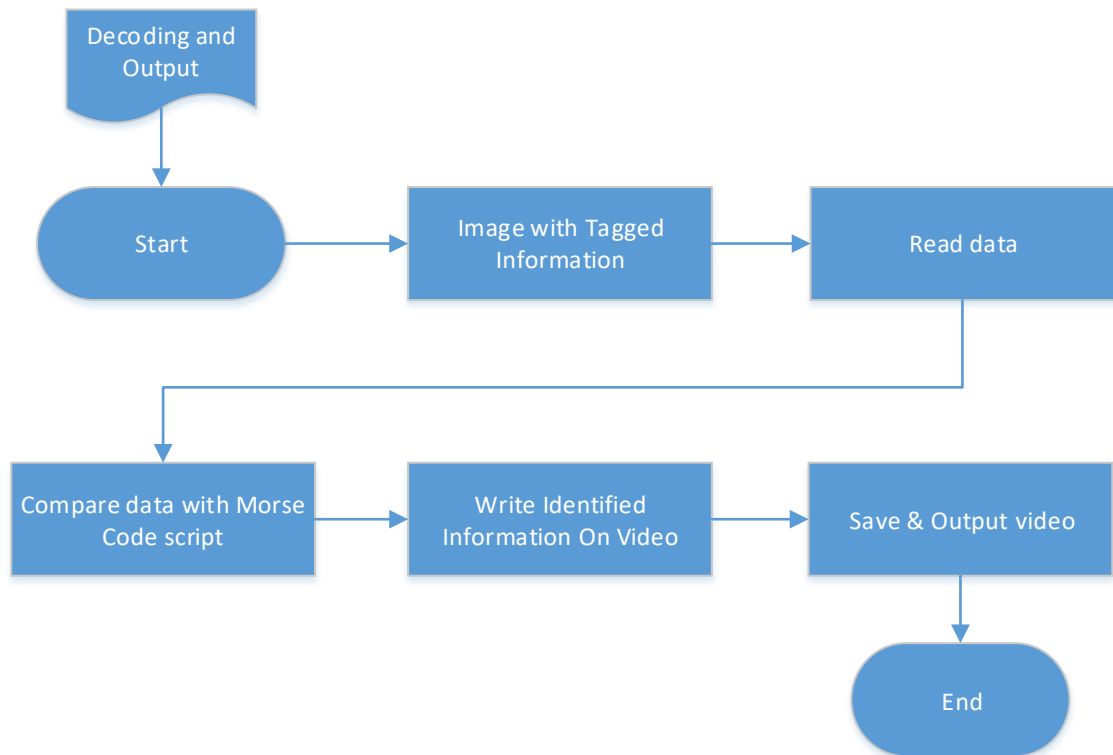


Figure 3.4 The flowchart of decoding and video output

3.4.2.2 Machine Learning Algorithm

Figure 3.5 shows the basic flowchart of a machine learning-based algorithm, where we have used a Gaussian Pyramid to define the ROI (region of interest) and segmentation instead of binary image segmentation; then, we use a support vector machine for classification.

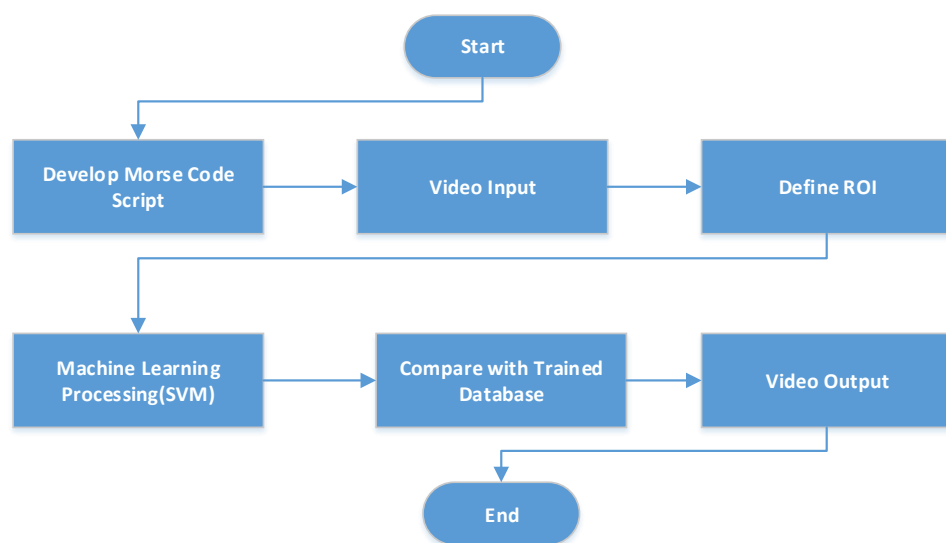


Figure 3.5 Basic flowchart of the finger gesture recognition system based on a machine learning algorithm

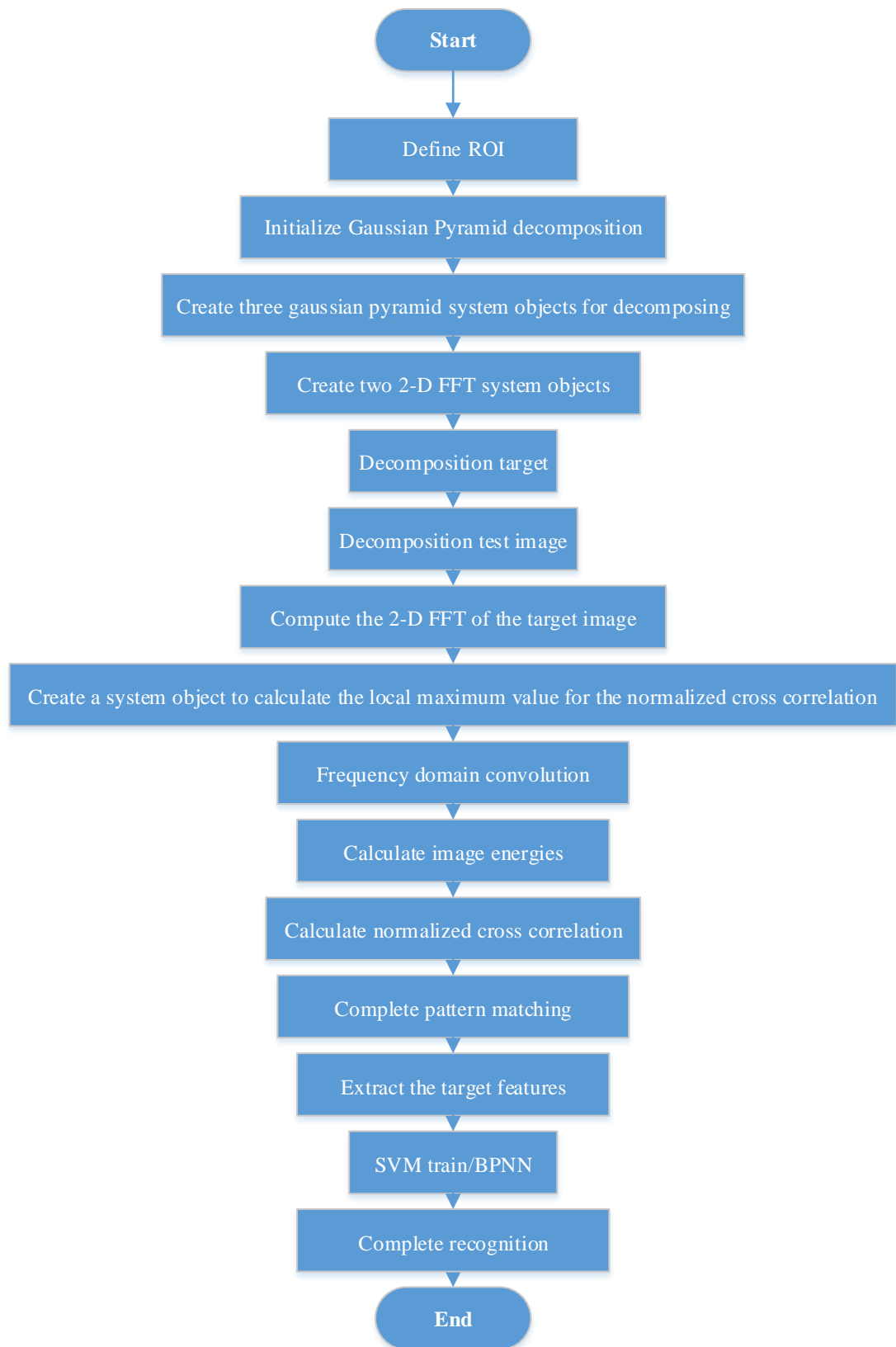


Figure 3.6 The flowchart of detailed shallow learning algorithm

The Gaussian Pyramid method has been used to define the region of interest in Figure 3.6, where the details show how the shallow learning algorithm is used to implement finger

gesture recognition. This is then further used to train the SVM data to achieve classification whatever the code is “dis” or “dahs” to complete the recognition process.

3.5 Algorithms

3.5.1 Image Segmentation Method

We have done a set using the traditional image segmentation recognition method at the beginning of the experiment. This method is based on the traditional image segmentation needed to identify the Morse code. The algorithm converts the RGB image from the algorithm 3.1 to the YCbCr colour space, the purpose being to get a Cb single channel image, because the Cb channel has a better ability to separate hand, shadow and background.

$$\begin{cases} Y = 0.257 \times R + 0.504 \times G + 0.098 \times B + 16 \\ Cb = -0.148 \times R - 0.291 \times G + 0.439 \times B + 128 \\ Cr = 0.439 \times R - 0.368 \times G - 0.071 \times B + 128 \end{cases} \quad (3.1)$$

where R, G, B represent the red, green, and blue channel of the RGB image, respectively.

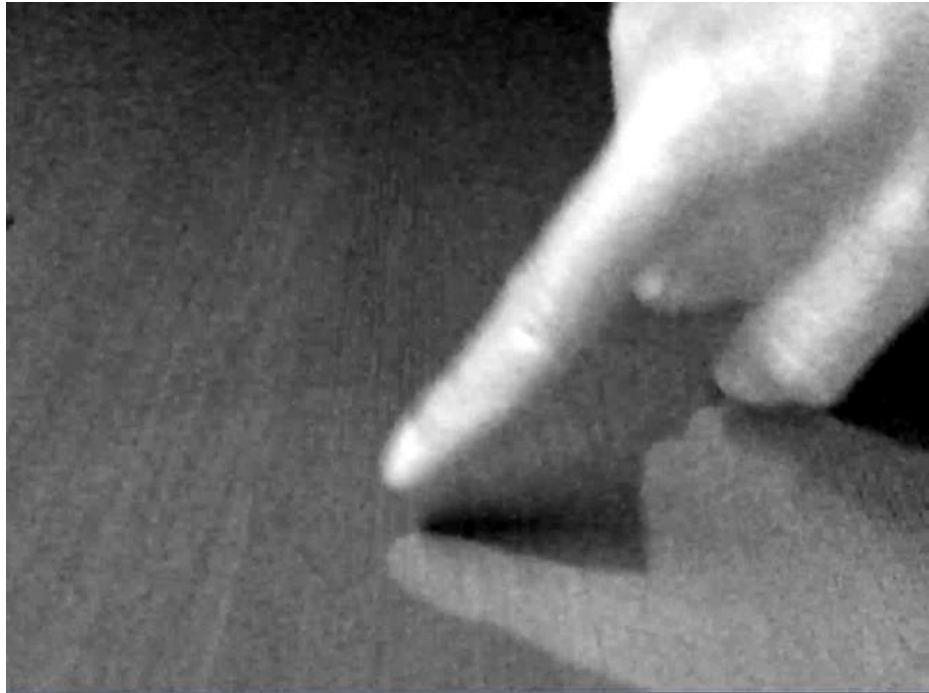


Figure 3.7 Cb image after colour segmentation

After the image in Figure 3.7 is acquired, the image is binarized to obtain the segmented image, and the segmentation is done by threshold segmentation.

$$Bw(x, y) = \begin{cases} 1 & Cb(x, y) \geq thresh \\ 0 & Cb(x, y) < thresh \end{cases} \quad (3.2)$$

where Bw is the binarized image, $Cb(x, y)$ is the intensity. The binarized image in Figure 3.8 is processed by Eq. (3.2).



Figure 3.8 Processed binarized image

When the finger touches the desk surface, there is a connected domain, the area of this connected domain is then calculated. This is also known as the shadow and finger area which determines whether it is a valid Morse code. Because when there is only one connected domain of the image, the Morse code is started, the area is much larger than the non-connected domain. The “dit” and “dah” are determined by using the length of the connected domain. In our experiment, the threshold is set to 70,000 and the area is calculated as in the following algorithm

$$S = \sum_{m=1}^{width} \sum_{n=1}^{high} Bw(m, n) \quad (3.3)$$

where S is the connected domain, Bw is the binarized image with index of the image (m, n) .

3.5.2 Feature Extraction & SVM/BPNN Classification Method

3.5.2.1 Feature Extraction

First, we need to select the training sample using its features to identify the finger gestures. As shown in Figure 3.9, the training sample selection can be found, and in this particular sample, it only happens when the finger touches the desk. The state of the sample picture is equal to “+1”, otherwise it equals to “-1”. The number of vectors consisting of ‘1’ and ‘0’ are used to determine the Morse Code, also known as ‘dit’ or ‘dah’. For example “1110000000” can be regarded as ‘1’ which has a less continuous frame rate – in this case, the machine would think it was “+1”. In the case of ‘1111111000’, the machine would think it was “-1”.

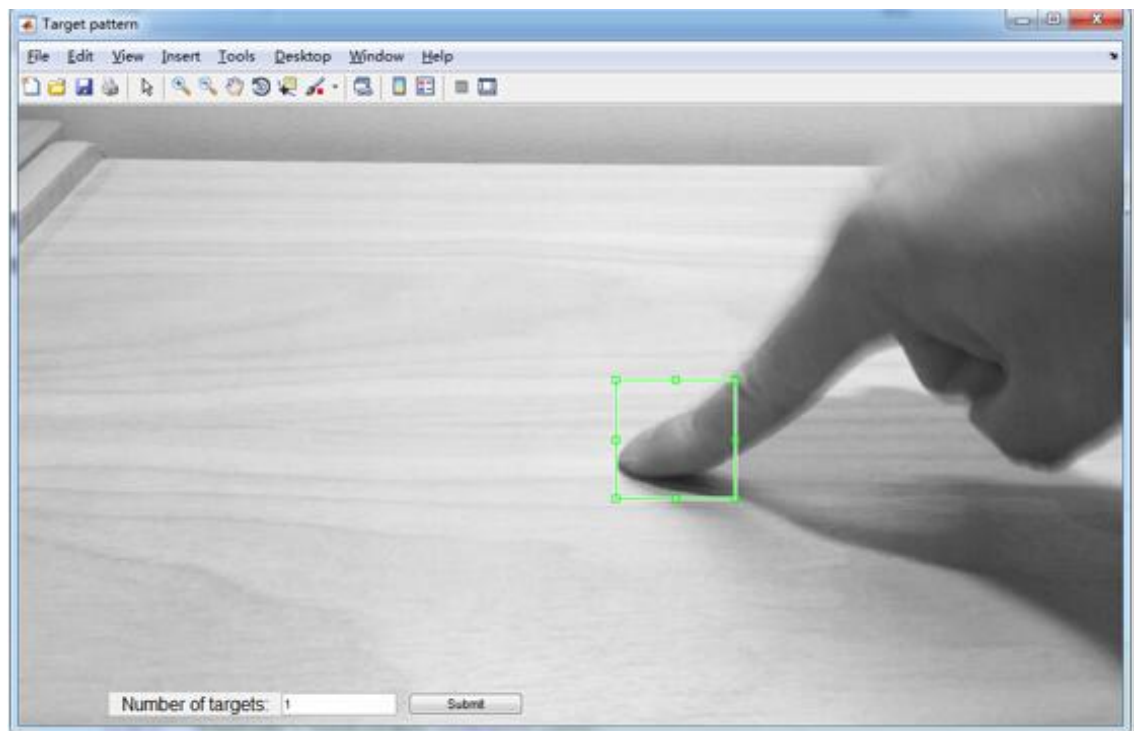


Figure 3.9 Training sample selection

The first step is to identify the samples and let the machine know the same object is perceived from different image scales. In practice, if the distance from the object is too far, it will be small; with the distance becoming shorter, the object gradually becomes larger, and finally becomes blurred; thus, we have the opportunity to use the Gaussian kernel function to represent this process.

The image is convoluted after the median filter, using nine different Gaussian masks respectively to get nine different kinds of images. This uses the Gaussian mask

$$G_i(x, y) = k_i \frac{1}{2\pi\sigma_i^2} e^{-\frac{(x-m_i/2)^2 + (y-\frac{n_i}{2})}{2\sigma_i^2}} \quad (3.4)$$

where $G_i(x, y)$ represents the i -th Gaussian mask, k_i stands for the weight of the i -th Gaussian mask, σ_i denotes the standard deviation of the i -th Gaussian mask.

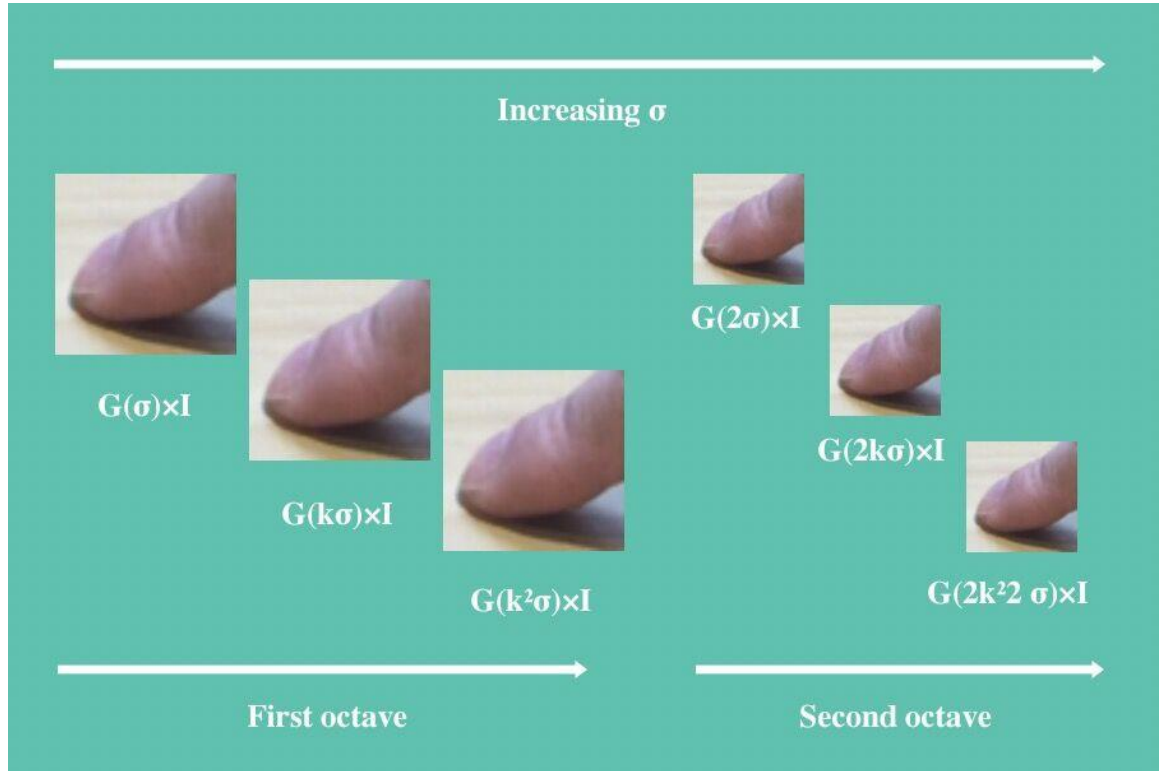


Figure 3.10 Gaussian pyramid

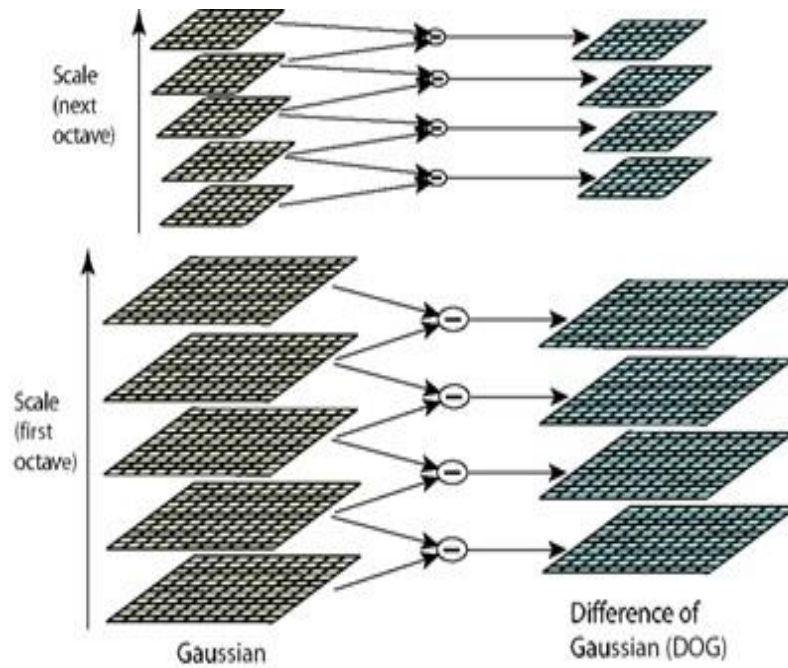


Figure 3.11 Image with different scales

After the Gaussian mask is convoluted, the differential Gaussian operator is used to detect the feature points, the differential Gaussian operator is shown as

$$D(x, y, z, k^i \delta) = [G(x, y, z, k^{i+1} \delta) - G(x, y, z, k^i \delta) * I(x, y, z)] \quad (3.5)$$

where k represents the scale factor, δ stands for the number of layers in the scale space pyramid. After the image has been undergone Gaussian convolution, the FFT features of different scale images for pattern matching are then extracted, we set the threshold as 0.99. Once the similarity reaches 0.99, the machine will think that is “+1”.

3.5.2.2 SVM-based Finger Gesture Classification

The problem to be solved in this thesis is to carry out convex quadratic programming of the problem of finger gesture recognition. From the theory of SVM, they are typical of two classifiers, and only answer the question to a positive or negative class; this is an easy and perfect method to resolve our classification problems.

We divided Morse codes into two categories: “+1” and “-1”. From the principle of SVM, it is easily applied to resolve the problem of finger gesture recognition.

This analysis is also conducted by taking the text extracted from the video frames having finger gestures of Morse codes. Through the optimization in three-dimensional space, the acquired images are grouped into “+1” and “-1”. The decision surface is expressed as

$$g(x) = \omega \cdot x + b \quad (3.6)$$

where ω is a variable. To classify the gesture category, we must find the two parameters ω and b . Because the samples have been given, the decision plane is thus determined, ω is confirmed by the sample points

$$\omega = \mathbf{A}\mathbf{Y} \quad (3.7)$$

where \mathbf{A} represents a vector, denoted as $\mathbf{A} = (a_1, a_2, \dots, a_n)$. ω not only holds the location of these sample points, but also relates to the type of these samples. $\mathbf{Y} = (y_1, y_2, \dots, y_n)^T$, $y_i \in \{-1, 1\}$ indicates the label or category of these samples, then we have

$$\omega = a_1 y_1 + a_2 y_2 + \dots + a_n y_n \quad (3.8)$$

where y_i represents the category of the i -th sample, the sample is used to determine the classification.

Thus, the nonlinear problem of finger gesture classification is possible to be converted into a linear one. To resolve the fault tolerance problem of these erroneous sample points, the slack variable ξ and the penalty factor c are introduced.

$$\min\{\mu(\omega, \xi)\} = \frac{1}{2}\|\omega\|^2 + c \sum_{i=1}^n \xi_i \quad (3.9)$$

If Eq.(3.9) reaches its minimum value, then we have,

$$\sum_{i=1}^n a_i y_i < x_i, x > + b + \xi \geq 1 \quad (3.10)$$

In our experiments, in order to make the deviation as small as possible, the selected kernel is a RBF kernel function; with this kernel, the obtained a and kernel function are combined together to achieve the classification.

Using the kernel function, we map the classification onto a higher dimensional space so that the classification problem could be solved by using a hyperplane linearly. In accordance with the previous description, the hyperplane is still based on the support vectors. The RBF kernel function-based classification is

$$f(x) = \sum_{i=1}^n a_i y_i < x_i, x > + b \quad (3.11)$$

The expression in higher dimensional space is

$$f(x) = \sum_{i=1}^n a_i y_i < \phi(x_i), \phi(x) > + b \quad (3.12)$$

If we use the radial basis functions (RBF) as the kernel, then the kernel function is

$$K(x, z) = \exp\left(-\frac{\|x - z\|^2}{2\sigma^2}\right) \quad (3.13)$$

3.5.2.3 BPNN-based Finger Gesture Classification

The BP algorithm of neural networks (BPNN) is also applied to classify the feature vectors extracted by using Gaussian pyramid as shown in Figure 3.10. The BPNN algorithm is a classifier in supervised learning, consisting of an input layer, an implicit layer, and an output layer, which are connected by modifiable weights. The core components of the algorithm are input learning samples, the back propagation algorithm,

the network weights and deviations to adjust the training, etc. The output vectors and the expected targets are desired as close as possible. When the error square sum of network output layer is less than a threshold, the specified error training is completed, the network weights and bias will be optimized.

The selection of an activation function is an important part of the BP neural network. We have used function *Sigmoid* for the training in this thesis. The BPNN algorithm consists of forward transmission and back propagation of the errors. In the forward transmission, the input information passes through the hidden layers to arrive the output; the previous layer only affects the neurons of its next layer. If the output layer does not get the expected outputs, then we need to calculate the output layer errors and reverse propagation along the connection path, get back to modify the weights of each layer. The steps of the BPNN algorithm is:

Step 1. Initialization:

$$o_i^{(0)}(t) = x_i(t) \quad (i = 1, 2, \dots, n) \quad (3.14)$$

where $i=1$ is an input layer.

Step 2. Integration:

$$u_j^{(l)} = \sum_{i=1}^{n_{l-1}} w_{ij}^{(l)} o_i^{(l-1)} \quad (j = 1, 2, \dots, n_l) \quad (3.15)$$

Step 3. Excitation:

$$o_j^{(l)} = \frac{1}{1 + \exp\{-\lambda_j^{(l)}(u_j^{(l)}(t) - b_j^{(l)})\}} \quad (3.16)$$

where $j=1, 2, \dots, n$ is the intermediate layer.

Step 4. Conditional transfer: $l < L$, $l = l + 1$, and jump to Step 2.

Step 5. Output:

$$o_j(t) = o_j^{(l)}(t) \quad (j = 1, 2, \dots, m) \quad (3.17)$$

In this thesis, we will implement the finger gesture recognition by using two classifiers: RBF kernel and BPNN algorithm; we will use them to recognize human finger gestures so as to enter the Morse codes into our computer system.

3.6 Expected Outcomes

The main expected outcome is to let the finger gesture recognition identify most of the finger gestures that users have made; 90% recognition rate is the target of this thesis. In this thesis, we will experiment with image segmentation and machine learning algorithms, but we will mainly study and experiment based on SVM/BPNN classification.

Chapter 4 Research Findings

4.1 Introduction

In this thesis, we recognise the Morse code, which includes all the listed Morse codes and their combinations. In this chapter, our experiments test the actual performance of all the functions. We found four testers for testing our system. The final experimental results will be shown in Section 4.4.

4.2 Experimental Environment

To facilitate the implementations, we used a mobile camera at the top of the desk plane with the perspective view of 45 degrees opposite to the hand so as to shoot the satisfactory videos, the resolution of these videos is 1920×1080 with 30 fps.

We use a standalone desktop computer facilitated with Intel 3.0 Ghz CPU plus 4GB memory. Graphic card is AMD Radeon HD6950 with 2GB graphics memory. The Morse code-based gesture recognition is developed by using MATLAB R2016a.

4.3 Experiments

4.3.1 Experimental Algorithms

Algorithm 4.1 is used to deal with the binarization images. The original RGB frames will be transformed to the YCbCr. The YCbCr colour space has better performance than the HSV colour space. After all the images are binarized, we label connected components in the binary image and output the binarized image.

Algorithm 4.1 Processing binarized image

Input: Original video

Output: Binarized frames

Procedure:

```
out = VideoWriter; Read the video
Get number of frames
Open the Handle
While (loop all the video frames)
    Read the first frame;
    Give frame variable;
```

```

Uniform frame size;
Change RGB colour to YCbCr colour space;
Extract the H component;
Create a 3-latitude image that assigns YCbCr color space separately;
2-latitude;
Write the frame to store;
Image binarization;
Remove small objects from binary image;
Mark the image information;
Count the white area in the binarized image;
Measure a set of properties for each labelled region in the label matrix L;

```

```

For loop
    Statistics of the white area
end
if
    sort the white area from large to small
end
if
    select largest white area greater than 120000
    ignore this image
end
close(out);
delete(out);
clear out;
delete(vid);
clear vid;

```

In Algorithm 4.2, in order to read the upper part of the processing binarised image, it is further processed in this step. The regions of the image and the number of the connected regions are used to judge finger gestures.

Algorithm 4.2 Image Segmentation

Input: Binarized frames

Output: Image segmentation, judge of gesture

Procedure:

```

for Loop all the Binarized frames
    Concatenate strings horizontally
    Read image from graphics file;
    B=[1 1 1; 1 1 1; 1 1 1];
    Dilate image;

```



```

Fill image regions and holes;
Label connected components in binary image;
Determine if binary image is all black
Count the area of each connected domain I;
for k = 1:NUM
    Filled area plot for all frames;
end
set value area to large then drop.

frame image information is obtained
if(if number 1 and area 2 is>3000)
    set as area (1);
    set as area(2);
elseif(if number 2 and area 1>3000)
    if image area is greater than 3000, set as the starting position
    c(area) = nArea(1);
    d(area) = 0;
else
    c(area) = 0;
    d(area) = 0;
end
nArea = [];
I = [];
end

```

The algorithm of finger gesture recognition based on SVM and the Gaussian image pyramid is obtained by pyramid decomposition of the image of the selected region of interest. The average energy of the image is calculated to obtain the cross-correlation coefficient of the image for matching the template to extract the features. In the feature training, the two parameters ω and b in the hyperplane equation $y = \omega x + b$ are calculated by using iterations to make the algorithm converge and finally complete the training; then we use it to determine the hyperplane SVM so as to identify the unknown image. In the training process, linear, polynomial and RBF (radial basis function) kernel functions are used to compare the experiments. The three kernel functions are expressed as follows

$$K(x, z) = \phi(x)^T \phi(z) \quad (4.1)$$

$$K(x, z) = (x^T z + c)^d \quad (4.2)$$

$$K(x, z) = \exp\left(-\frac{\|x-z\|^2}{2\sigma^2}\right) \quad (4.3)$$

4.3.2 Experiments

In this section, a video footage is presented from Figure 4.1 to Figure 4.4. Figure 4.1 shows the original frame output; while Figure 4.2 shows a segmented binary image in YCbCr colour space and threshold segmentation. Figure 4.3 shows the final output video with the output character displayed at the upper-left corner. Figure 4.4 shows the Gaussian pyramid method being used; it only shows while the fingertip touches the desk top.

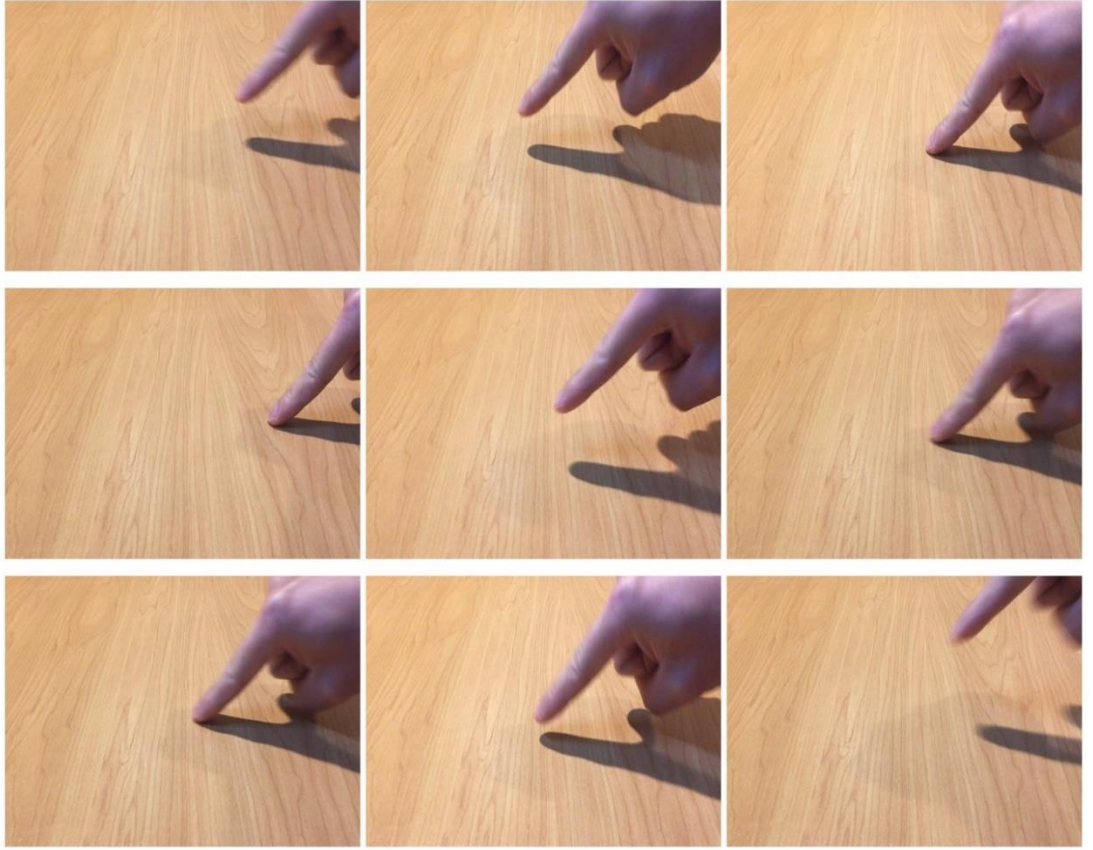


Figure 4.1 Original frames

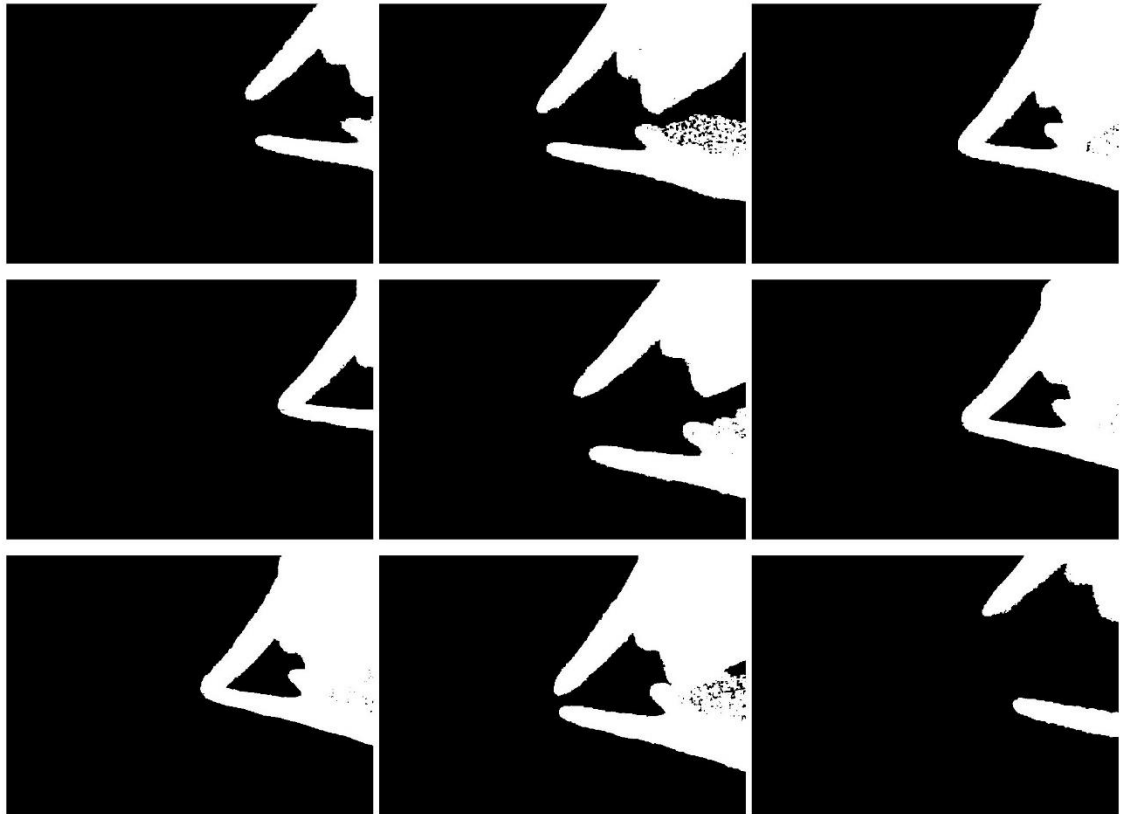


Figure 4.2 Segmented binary image

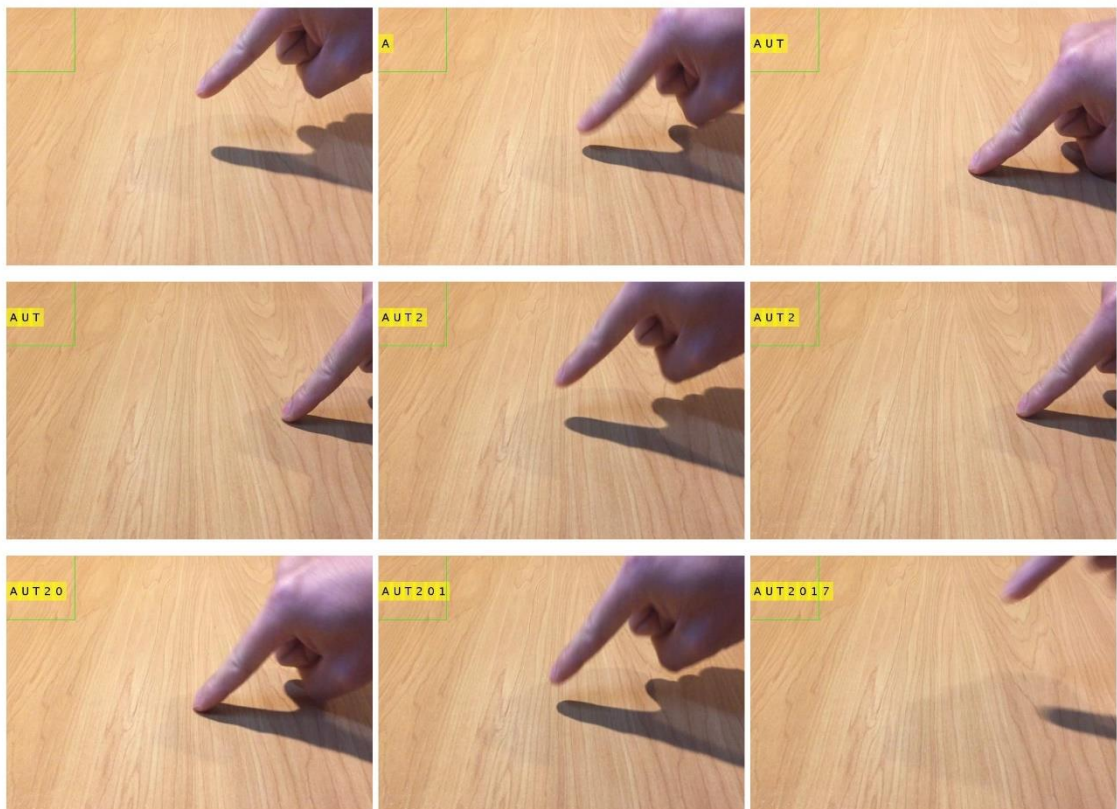


Figure 4.3 Final output frames

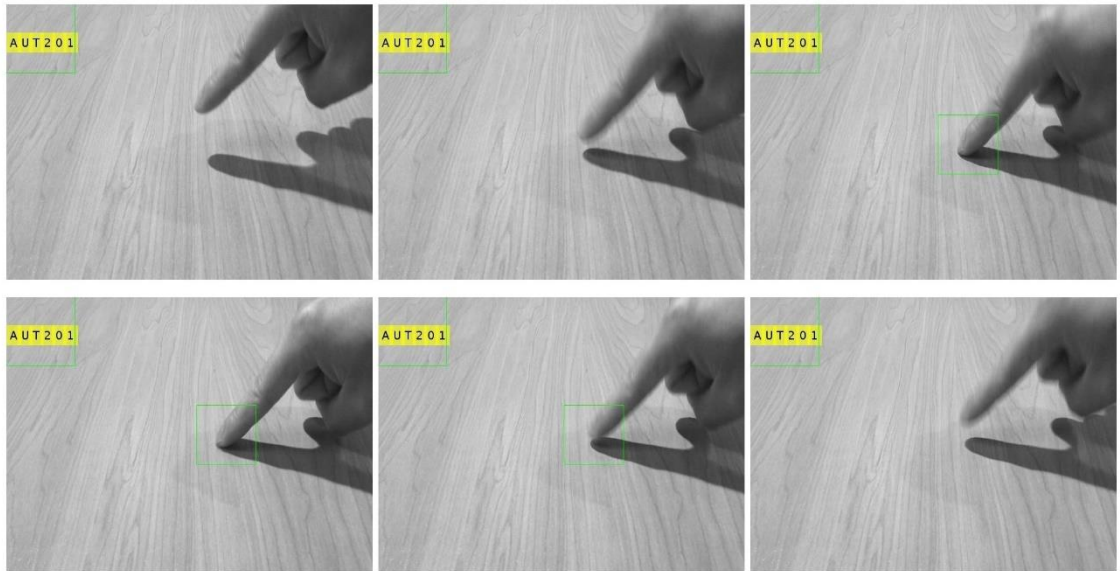


Figure 4.4 Video frames of fingertip recognition

In Figure 4.5, the vertical projection lines clearly show the duration of fingertip touching desk top; the x -axis indicates the total number of frames; whereas, '1' refers to the fingertip touched the desk surface as well as '0' means no touching has been detected.

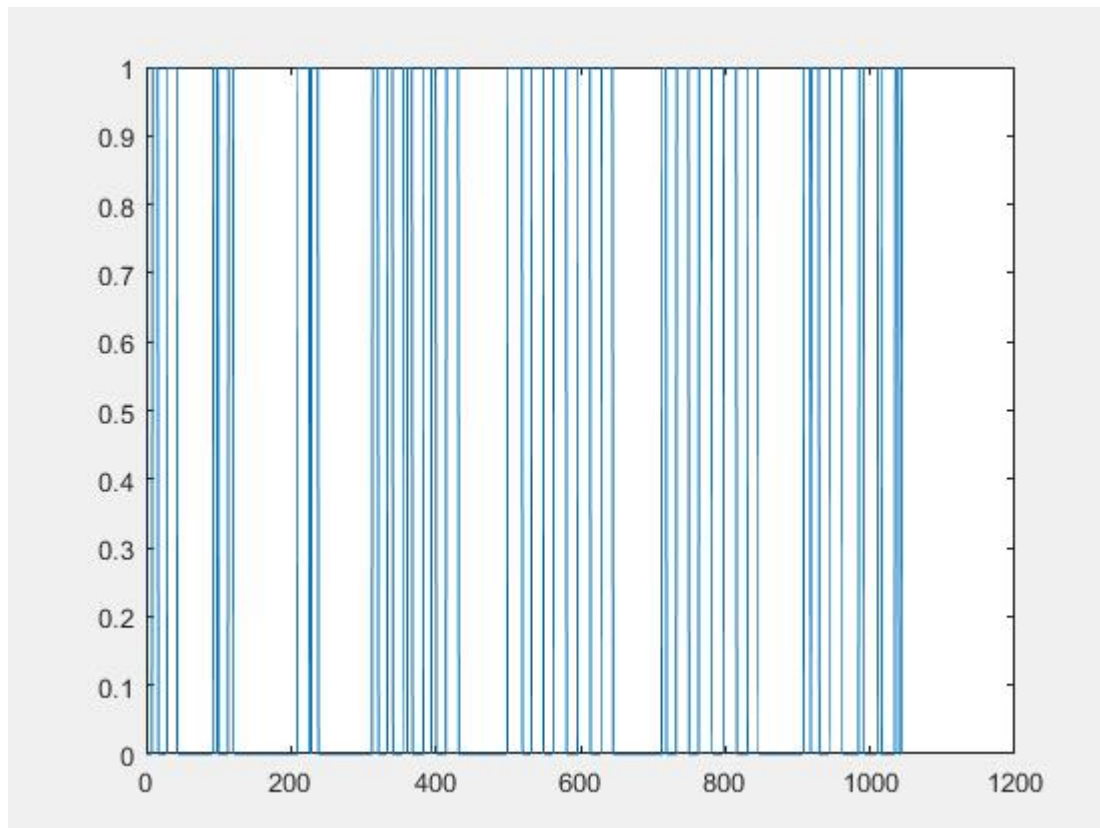


Figure 4.5 Match metric of gesture recognition

In Figure 4.6, the horizontal projection line shows that the fingertip has been detected; the x -axis shows the total frames of the video; where 0.99 means a fingertip was detected, and ‘1’ means that the detected fingertip has already touched the desk top. In Figure 8, we clearly see the motion of our finger; the short peak duration was identified as “dis”, and longer peak duration was identified as “dahs”

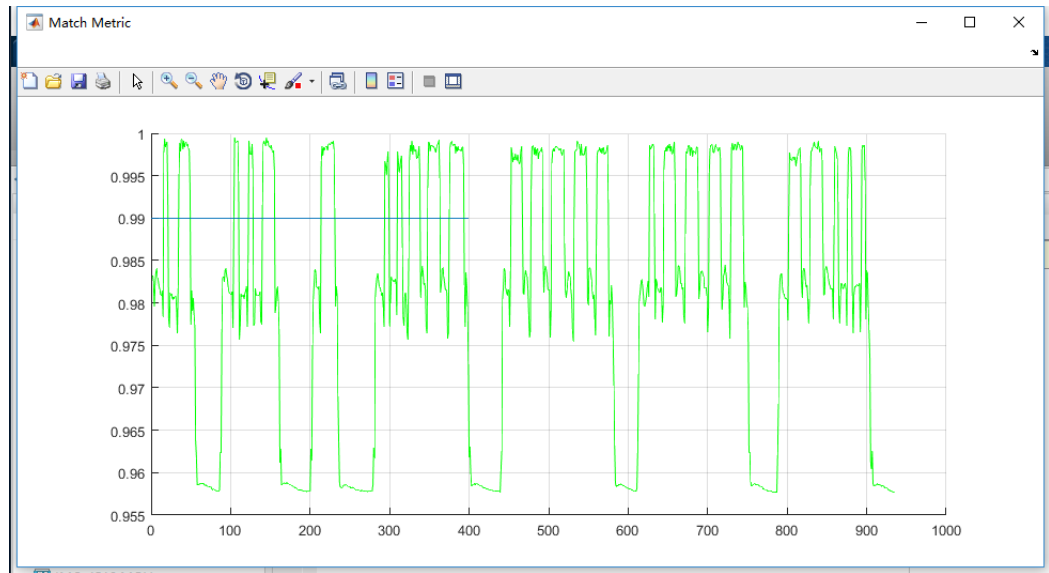


Figure 4.6 Fingertip recognition match metric

4.4 Experiment results

4.4.1 Finger gesture recognition based on image segmentation

The result of the experiment is to test the accuracy of Morse code recognition. A total of four testers participated in this experiment. The content of our test is to require our testers to enter all the characters in the database in the testing environment using Morse code gestures, which includes 26 English letters, ten Arabic numbers, eighteen punctuations, four SMSs and two emoticons. Each tester will enter each character five times to test the accuracy of the finger gesture recognition. In this testing environment, the desk top environment as well the lighting conditions remain unchanged.

In Table 4.1, the first tester entered a total of 60 characters, where each character was repeated five times. Similar characters would be classified as a group, thus facilitating the statistics. Average character accuracy by the first tester was 60.9% the second tester was

62.7% the third tester was 60.6%, the fourth tester was 34% and the total accuracy rate was 54.6%.

Table 4.1 Results of accuracy of Morse code for all testers

Testers	Average accuracy for alphabet	Average accuracy for numbers	Average accuracy for punctuation	Average accuracy for SMS & emoticons
A	65.4%	72%	53.3%	52.8%
B	66.2%	70.8%	59.7%	54.1%
C	59.6%	67.6%	52.8%	62.5%
D	30.5%	42.2%	38.4%	25%

4.4.2 Finger gesture recognition based on SVM/BPNN

Firstly, the algorithm is decomposed by the pyramid to get the image in different scales. We calculate 2D-FFT, energy and cross-correlation coefficients to achieve the matching template. We extract the image feature input path from the SVM training, get the classifier, and finally complete the identification. We see that the recognition algorithm is different from the segmentation recognition algorithm in the previous section. The algorithm of machine learning does not need to be segmented and has better adaptability. The algorithm can work better in different backgrounds, while at the same time there is great improvement in the recognition rate compared to the previous algorithm.

To verify the effectiveness of different kernel functions on the classification, three groups of Morse codes were randomly selected. Each group of Morse codes contained eight different labels. Each Morse code concluded in a group 100 images is regarded as a feature; and the training dataset is randomly selected, half for training, half for test. Respectively. The confusion matrix and corresponding ROC curve to reflect the classification and accuracy of the situation are provided.

4.4.2.1 Experiment result set A

In this experiment, we use the linear, polynomial, and RBF as the kernel functions. We randomly use the Morse codes '3', '6', 'A', 'B', '?', 'T', '.' and 'I'. The confusion matrix and the ROC curve are provided to evaluate the classification. First, we choose the linear function as the kernel function of the SVM.

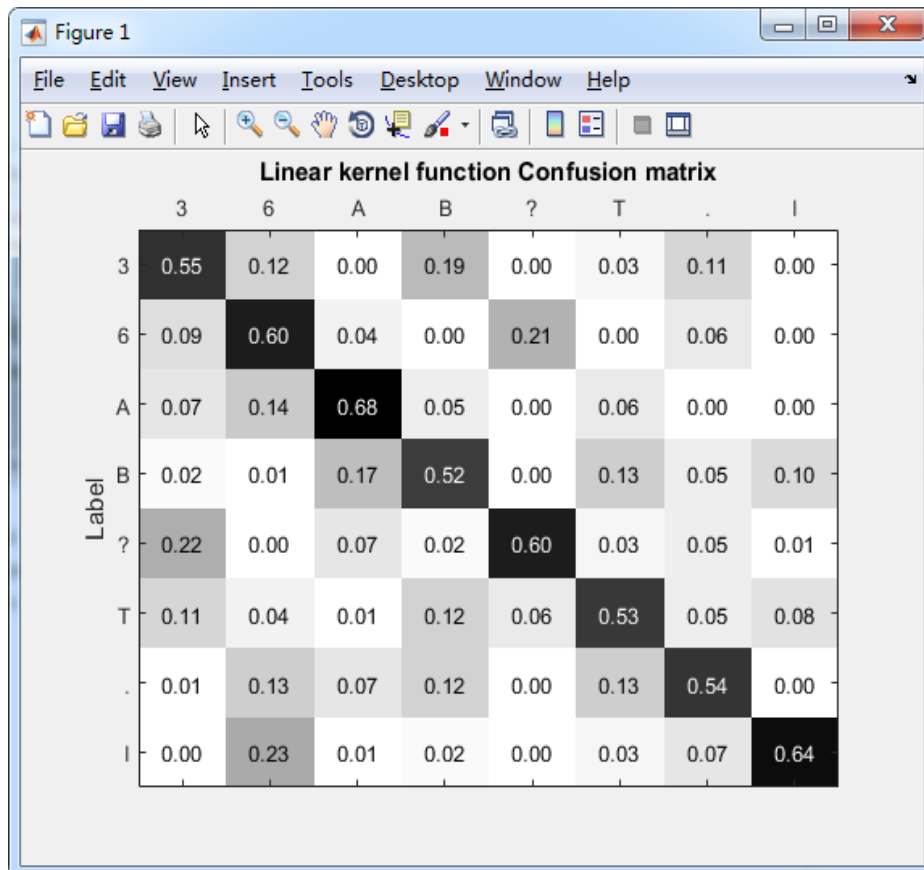


Figure 4.7 The confusion matrix of the test dataset

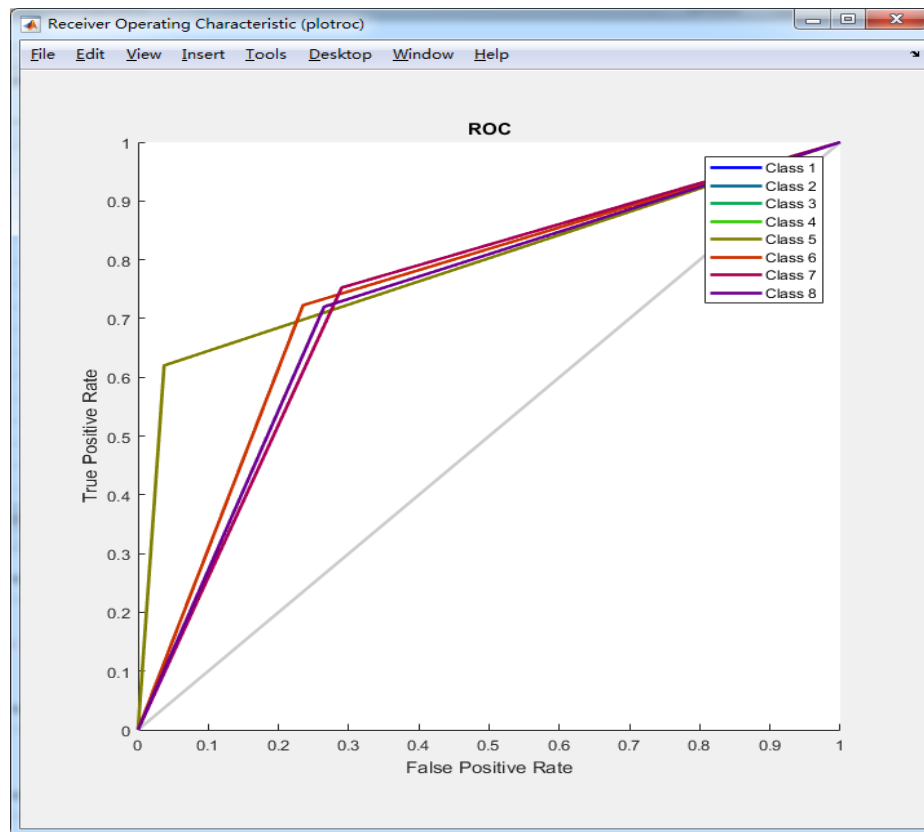


Figure 4.8 The ROC result for the test dataset

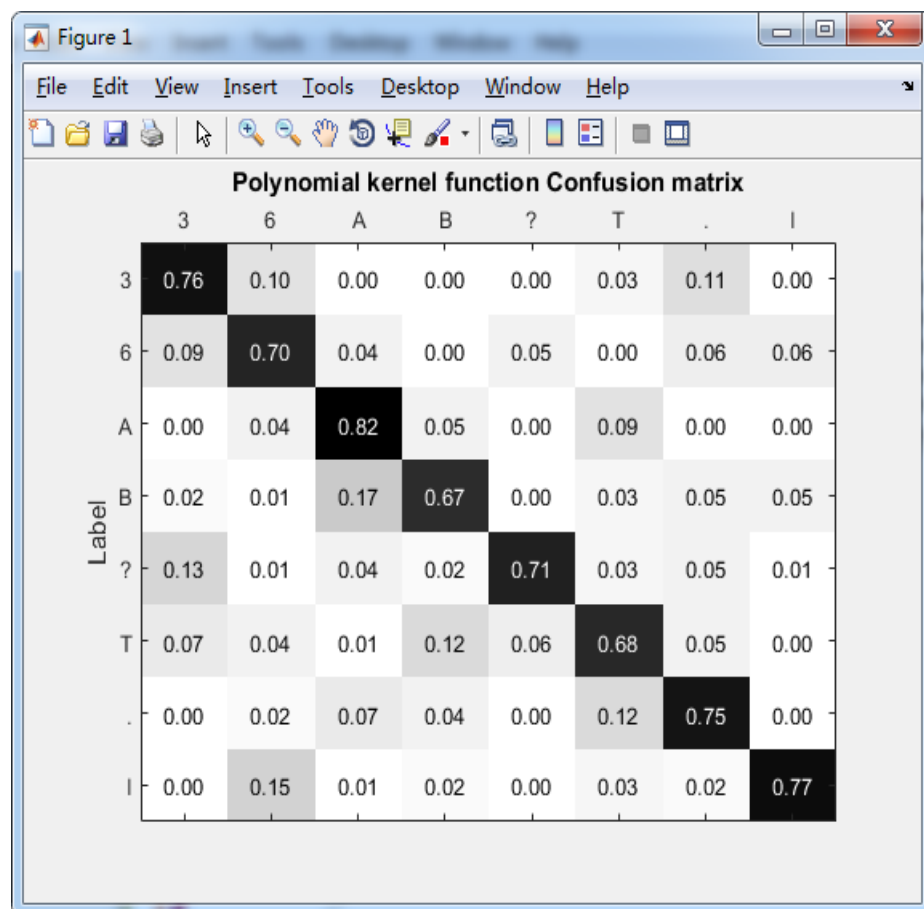


Figure 4.9 The confusion matrix of the test dataset

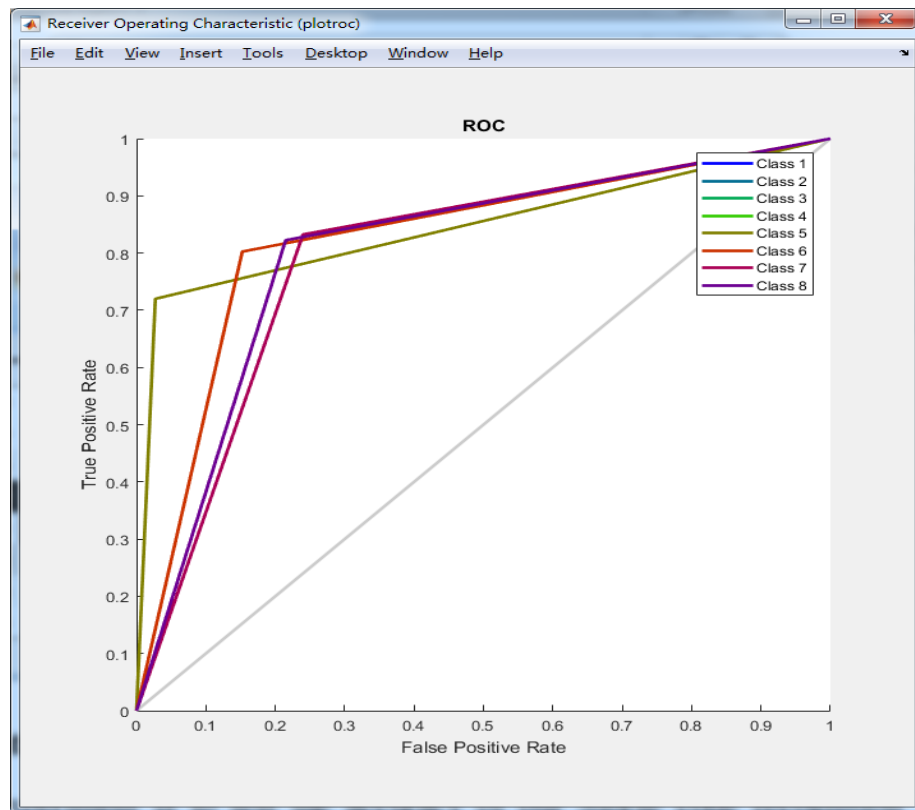


Figure 4.10 The ROC result for the test dataset

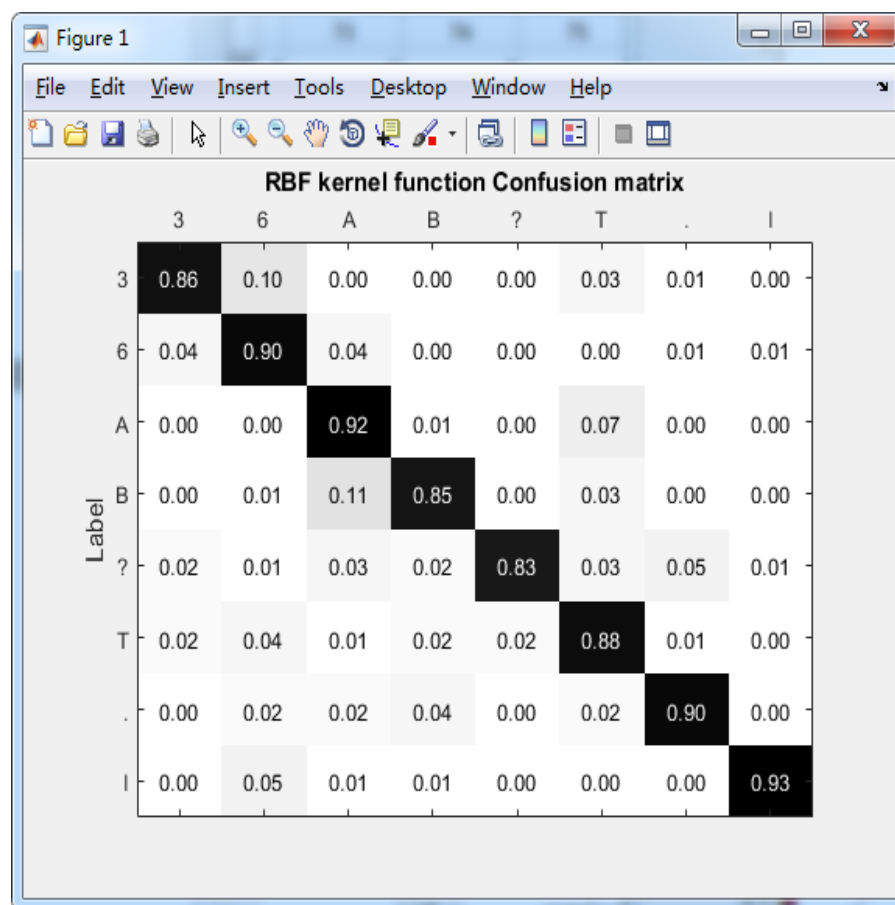


Figure 4.11 The confusion matrix of the test dataset

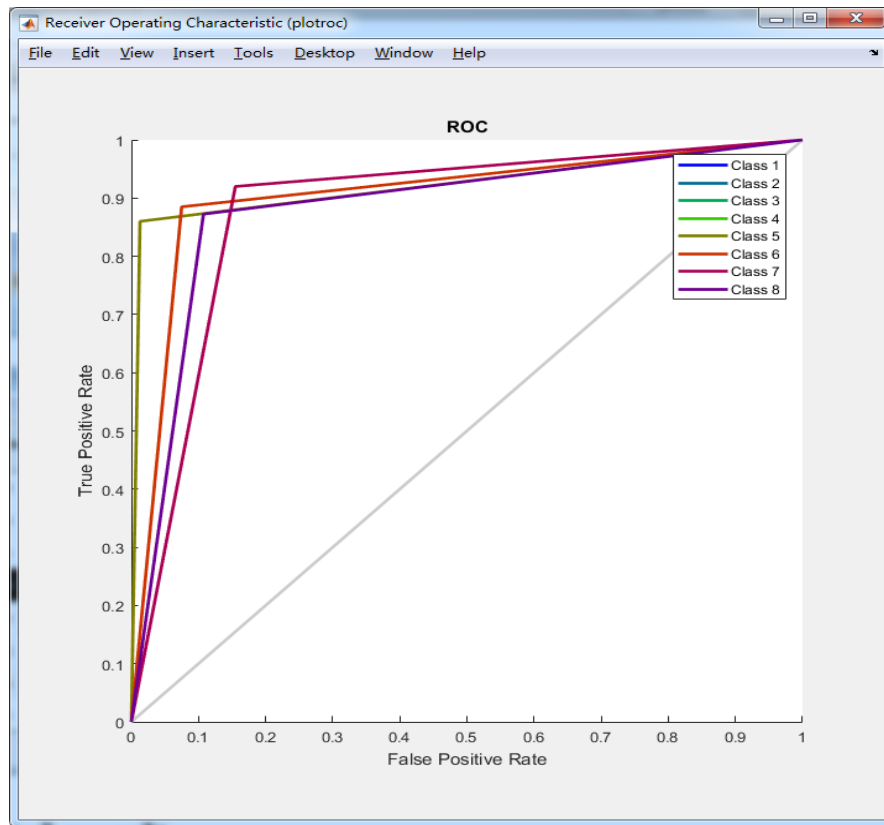


Figure 4.12 The ROC result for the test dataset

4.4.2.2 Experiment result set B

In this experiment, we use the linear, polynomial, and RBF as the kernel functions. We randomly use the Morse codes '9', '0', 'C', 'O', '!', 'W', ',' and 'U'. The confusion matrix and the ROC curve are used to evaluate the classification.

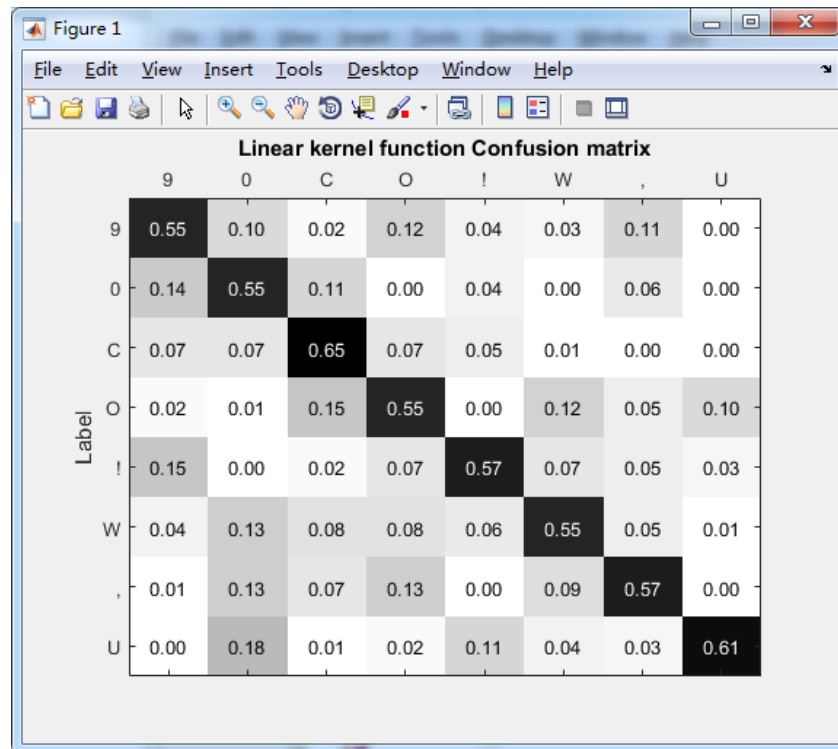


Figure 4.13 The confusion matrix of the test dataset

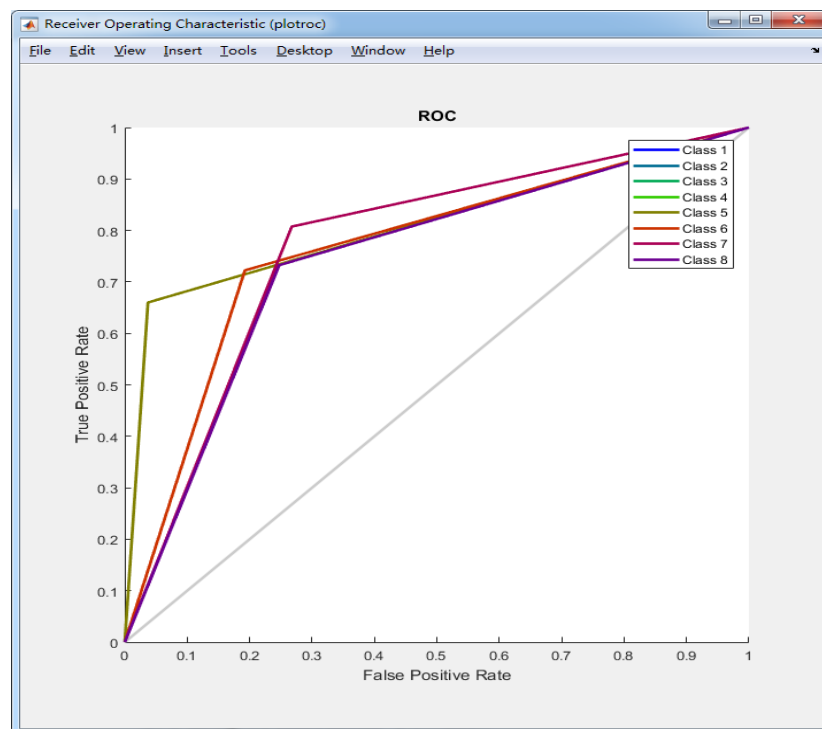


Figure 4.14 The ROC result for the test dataset

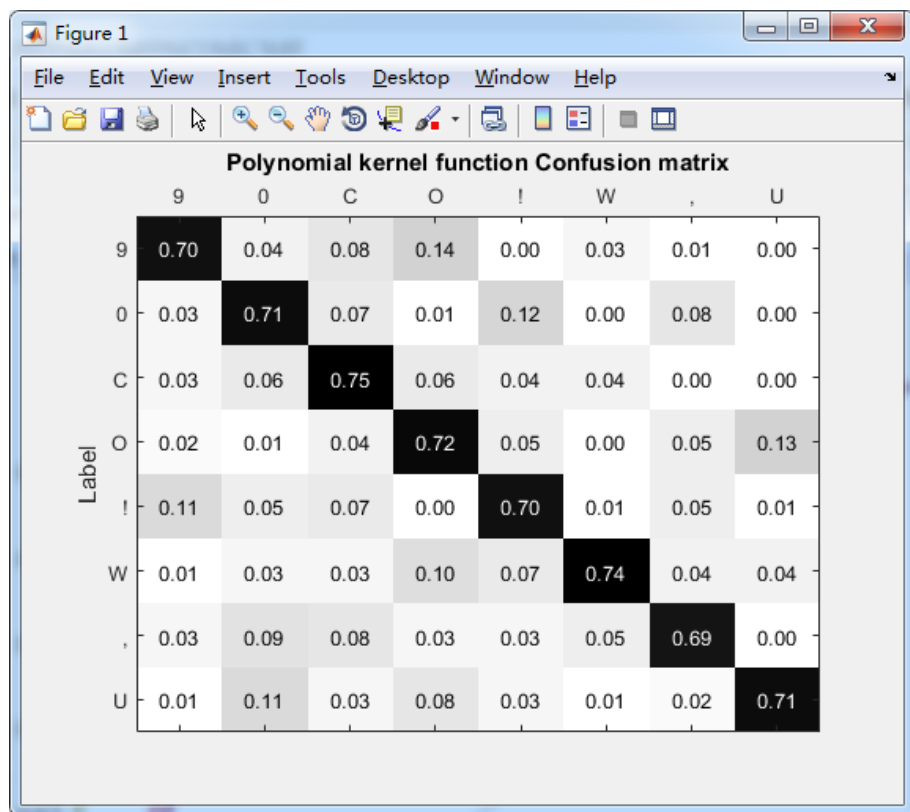


Figure 4.15 The confusion matrix of the test dataset

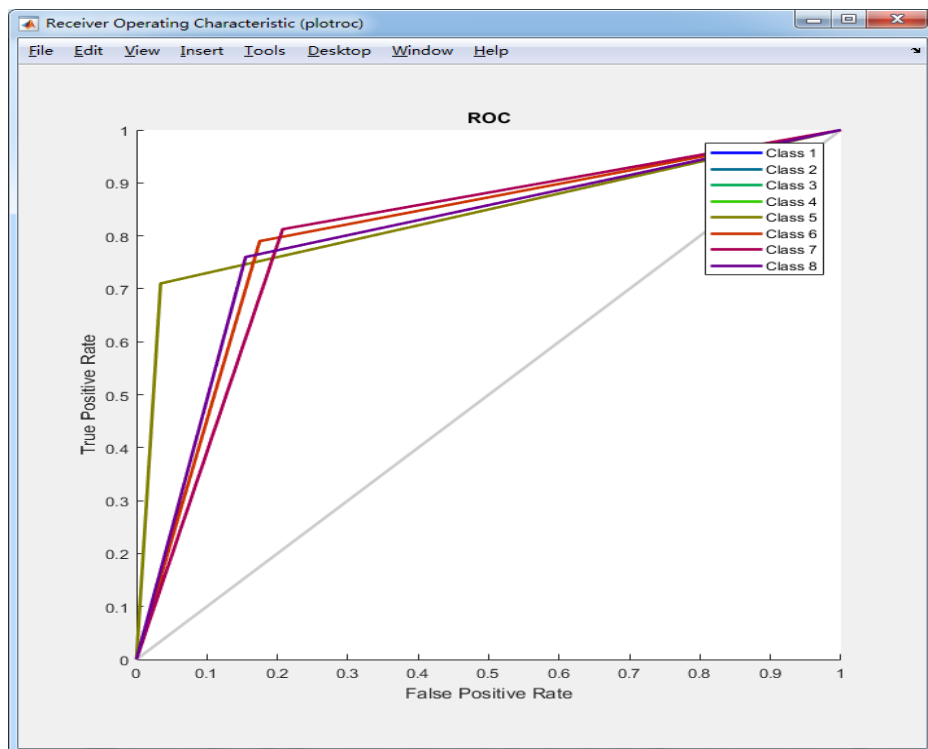


Figure 4.16 The ROC result for the test dataset

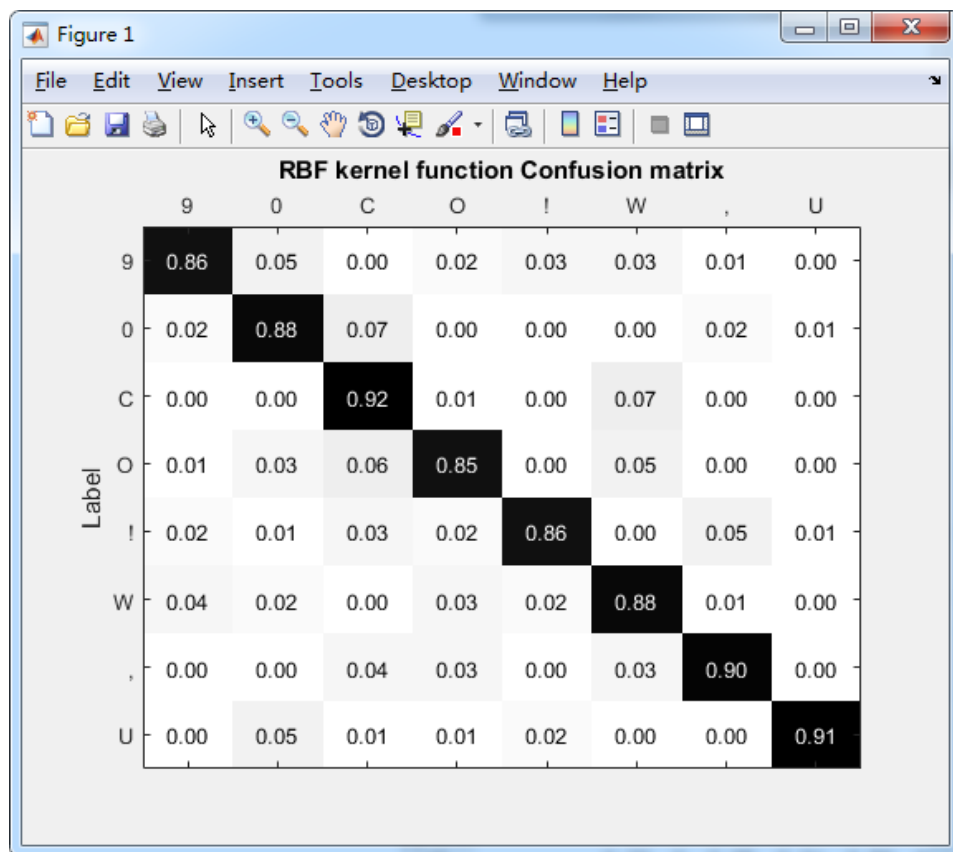


Figure 4.17 The confusion matrix of the test dataset

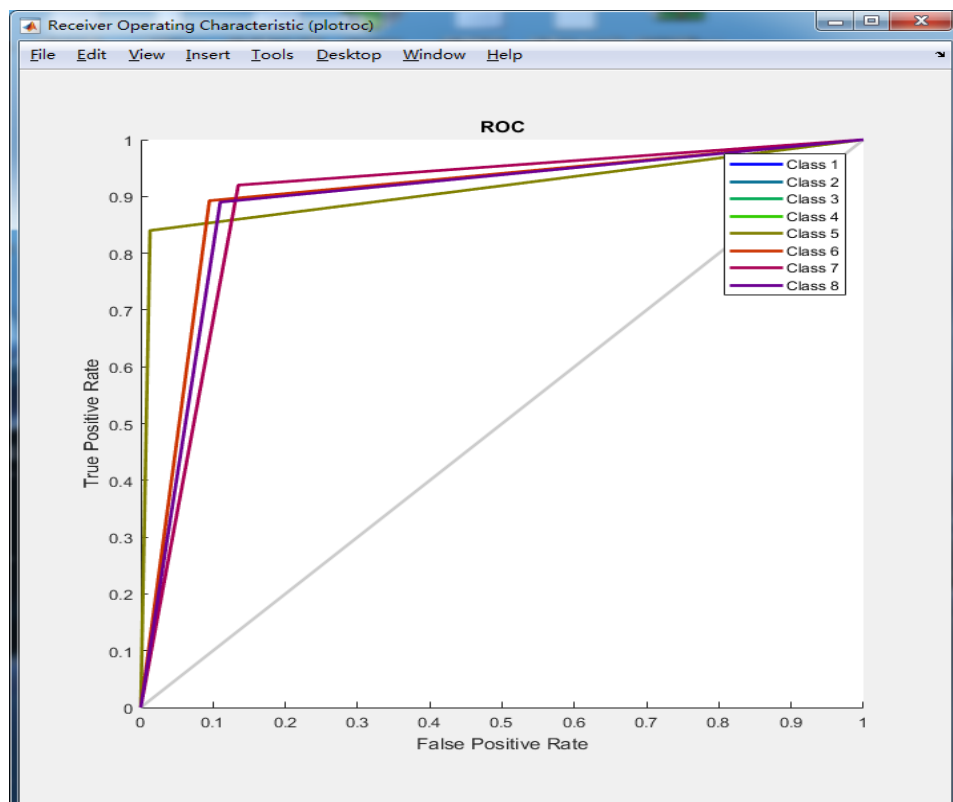


Figure 4.18 The ROC result for the test dataset

4.4.2.3 Experiment result set C

In this experiment, we use the linear, polynomial, and RBF as the kernel functions. We randomly use the Morse codes '5', '7', 'G', 'H', ';', '@', 'X', and 'U'. The confusion matrix and the ROC curve are used to evaluate the classification.

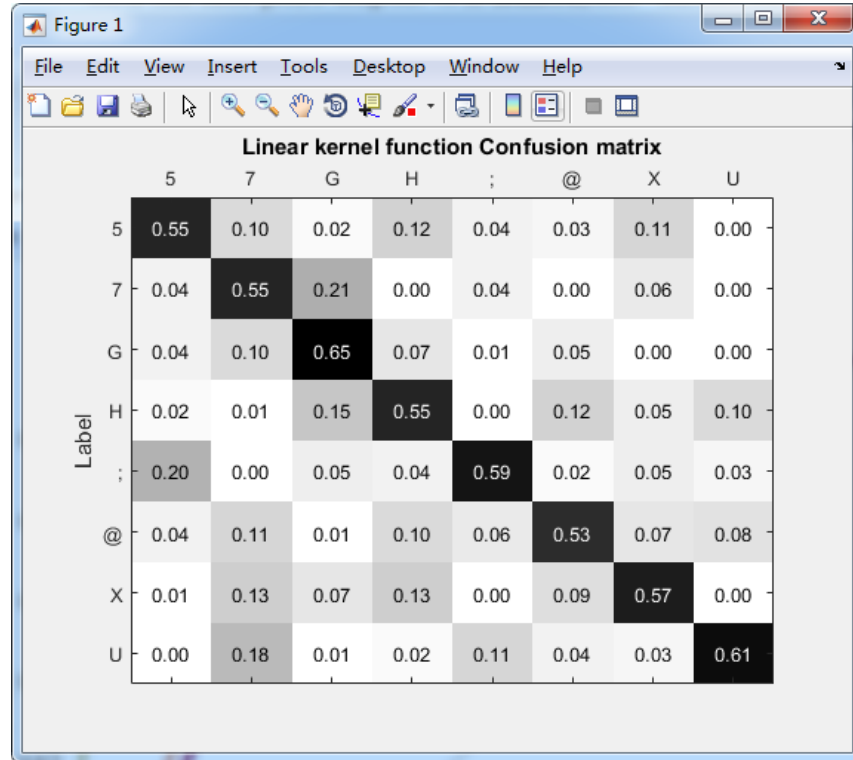


Figure 4.19 The confusion matrix of the test dataset

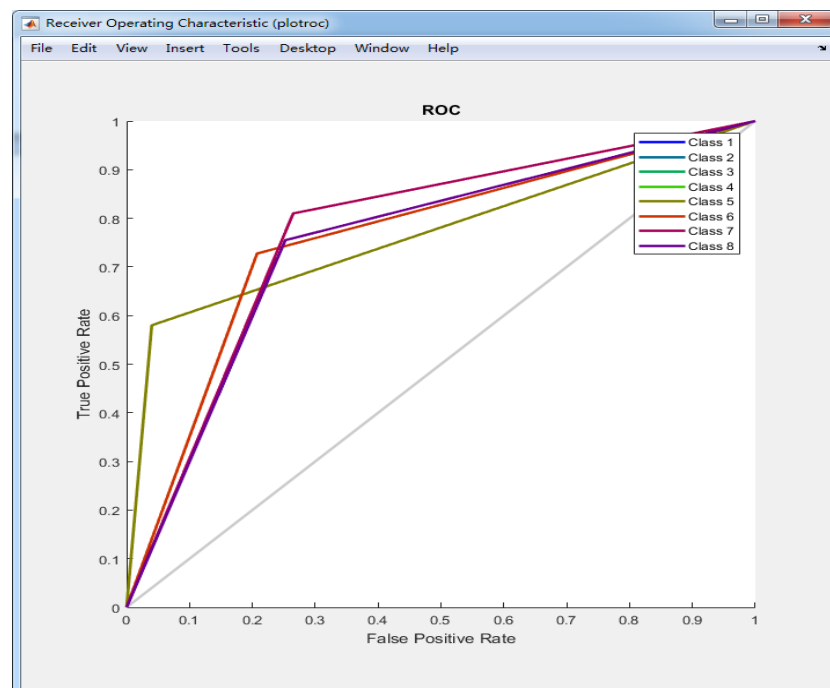


Figure 4.20 The ROC result for the test dataset

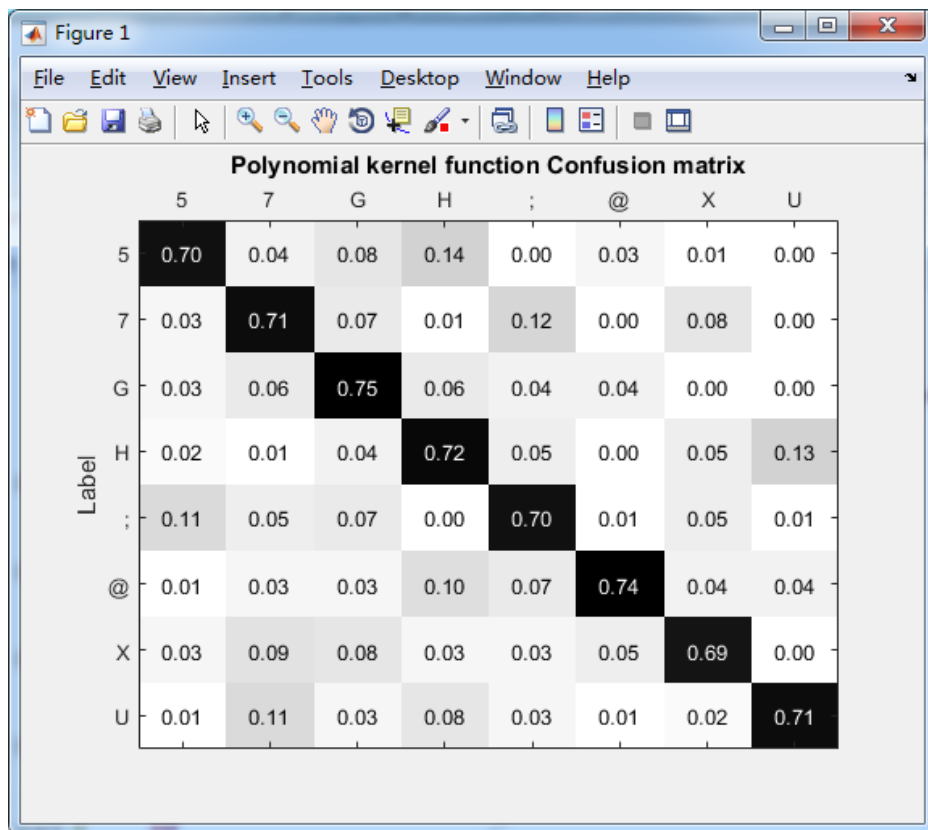


Figure 4.21 The confusion matrix of the test dataset

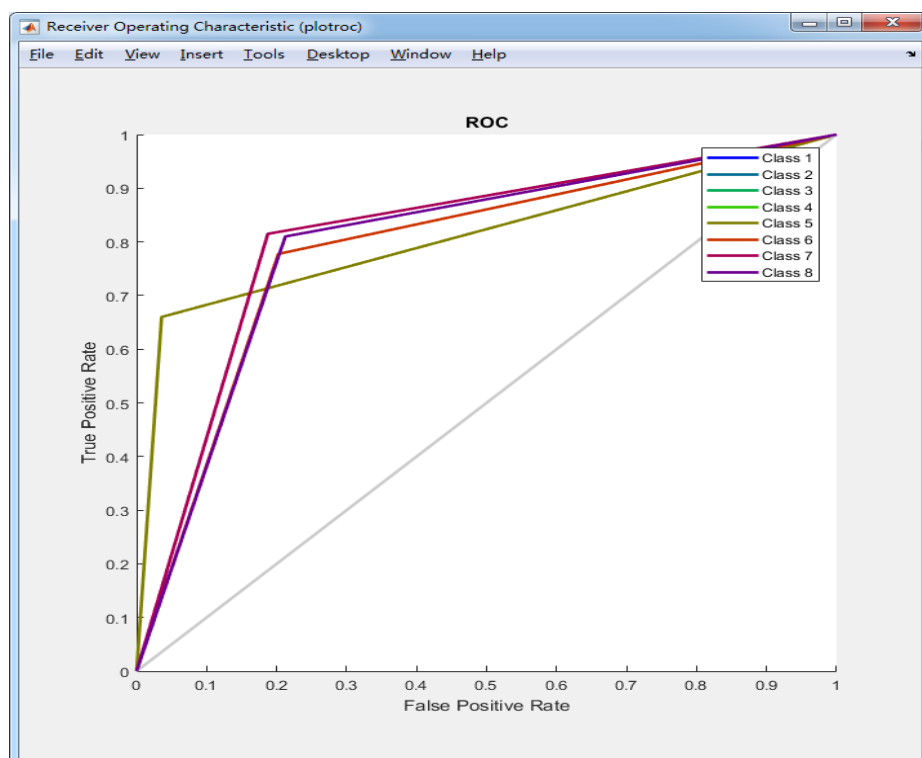


Figure 4.22 The ROC result for the test dataset

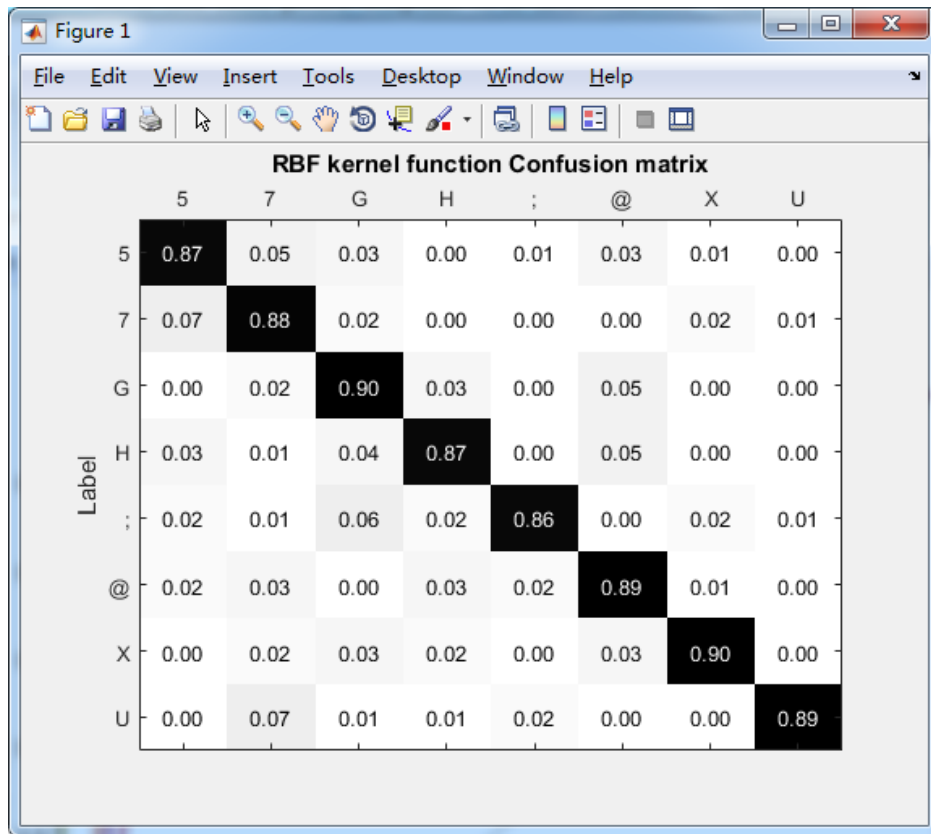


Figure 4.23 The confusion matrix of the test dataset

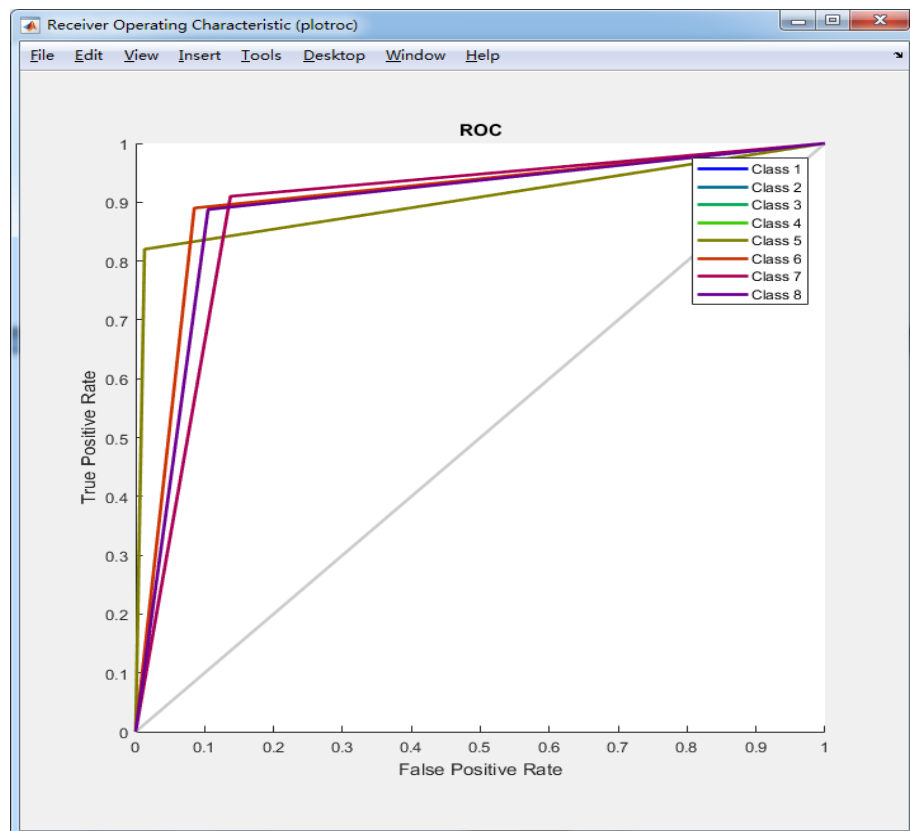


Figure 4.24 The ROC result for the test dataset

4.4.2.4 Experiment result set D

In this experiment, we also used the BPNN algorithm to compare with the SVM-based classifier. We chose the same samples of the 3 groups with eight letters as our training and testing samples. We randomly use the Morse codes '3', '6', 'A', 'B', '?', 'T', '.' and 'I'. Morse code '9', '0', 'C', 'O', '!', 'W', ',' and 'U'. Morse code '5', '7', 'G', 'H', ';', '@', 'X', and 'U'.

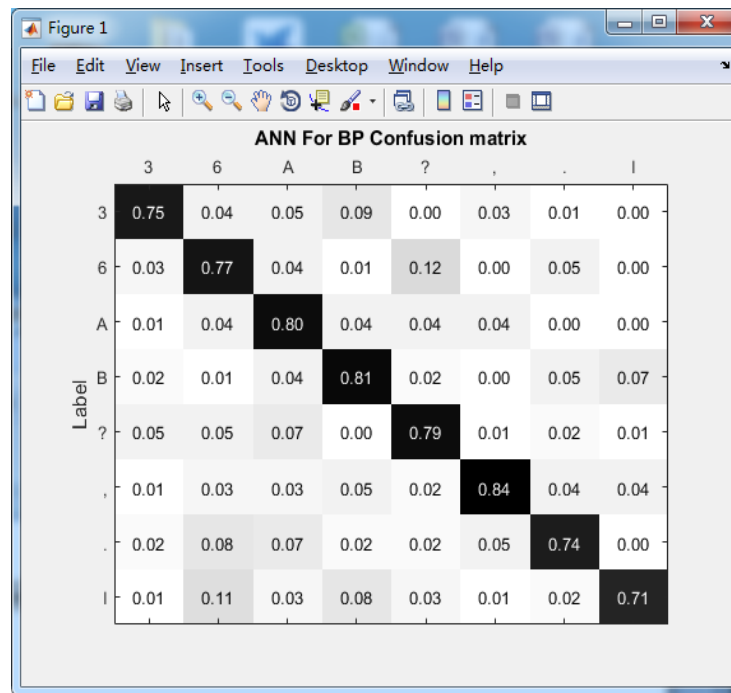


Figure 4.25 The confusion matrix of the test dataset

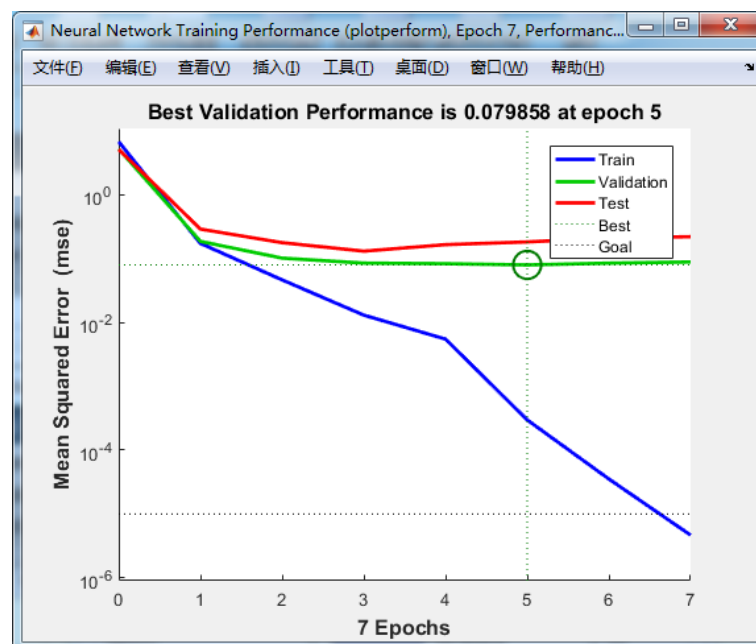


Figure 4.26 The performance of training dataset

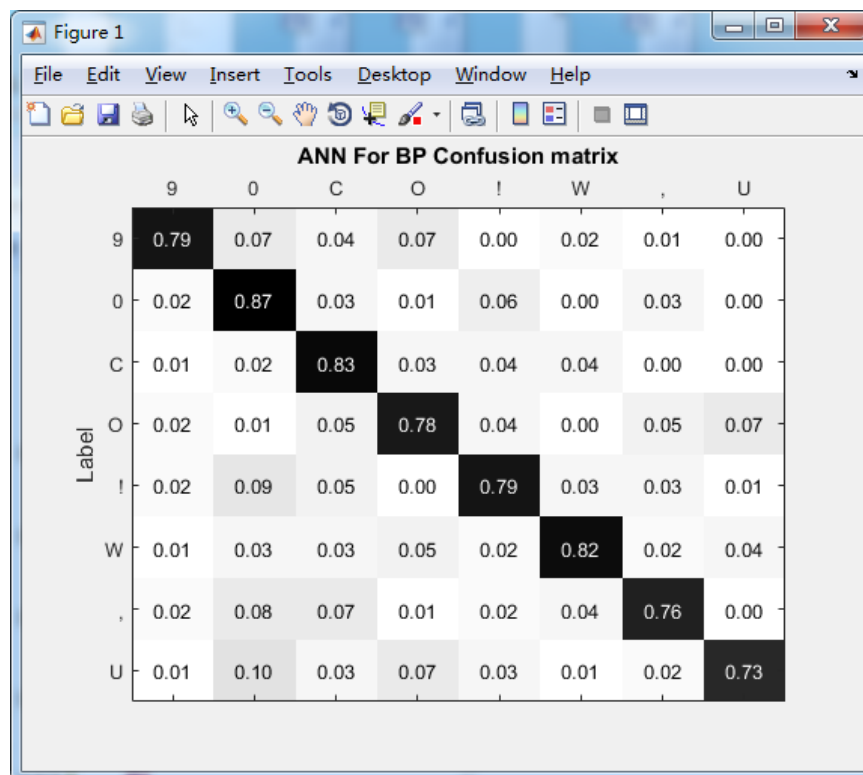


Figure 4.27 The confusion matrix of the test dataset

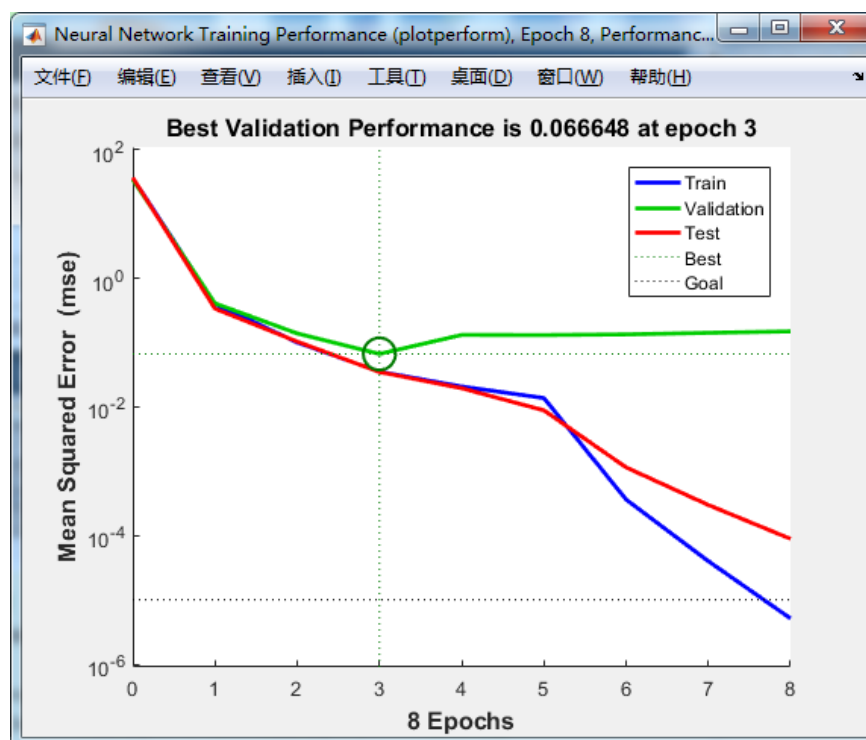


Figure 4.28 The performance of training dataset

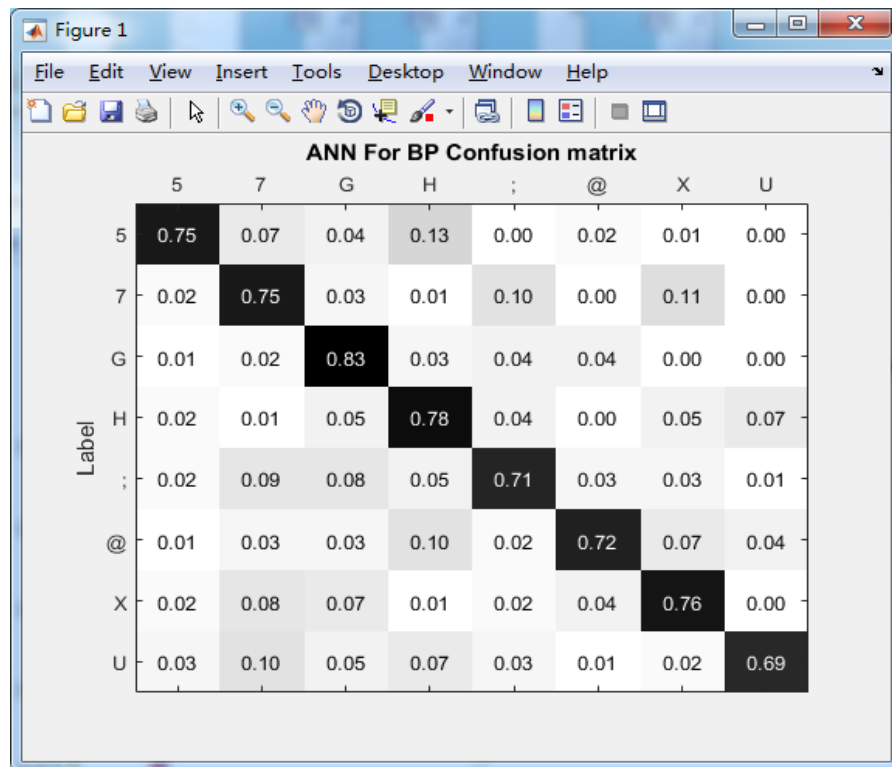


Figure 4.29 The confusion matrix of the test dataset

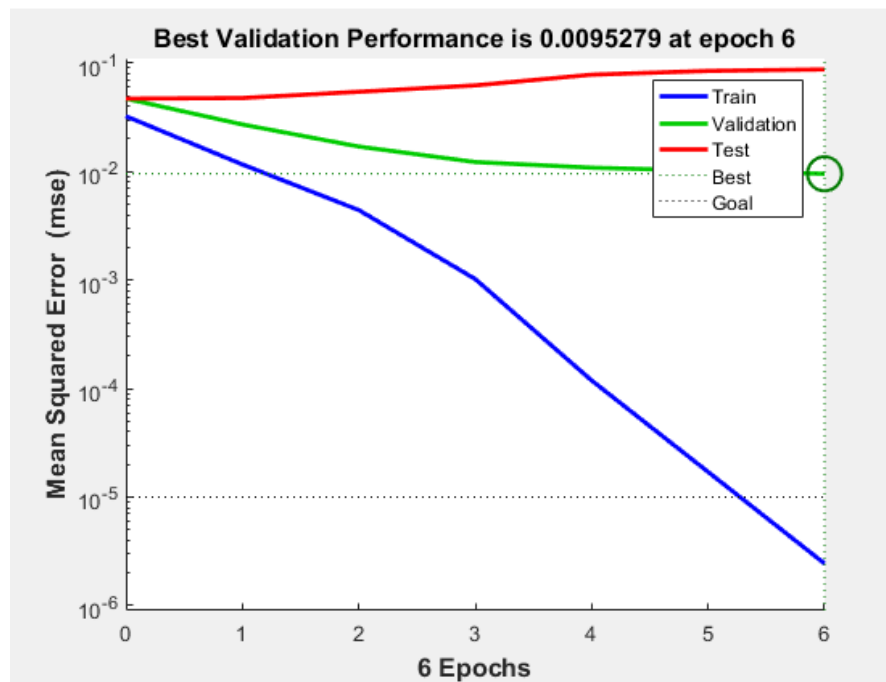


Figure 4.30 The performance of training dataset

Table 4.2 Overall average accuracy for all characters

Morse Codes	Linear	Polynomial	RBF	BPNN
Alphabets	0.6	0.74	0.89	0.79
Arabic Numbers	0.57	0.72	0.88	0.79
Punctuations	0.57	0.7	0.88	0.76
SMS and Emoticons	0.52	0.68	0.84	0.78
Mathematics Symbols	0.55	0.67	0.87	0.78
Chinese Characters	0.55	0.67	0.85	0.76

Table 4.2 shows the overall average results of accuracy of finger gesture recognition for all characters, detailed result refed in appendix.

Chapter 5 Discussions and Analysis

5.1 Introduction

In this chapter, we mainly analyse the experimental results, the performance of the algorithm as well as discuss our research questions. In Section 5.2, we evaluate the Morse code recognition based on the test results. Section 5.3 will discuss exhaustive details of the research question that was raised in Chapter 1. In Section 5.4, 5.5 and 5.6, limitations, discussions and recommendations will be presented.

5.2 Analysis and Evaluation

The lowest recognition rate of our proposed method is 60%. Among all tests, the highest recognition rate is 80%. We found that during the tests that the recognition rate is mainly affected by the finger sliding, the shadow region, and lighting conditions.

The experimental results show the recognition rate significantly reduces if the finger sliding exceeds our pre-set values, which often identifies a wrong Morse code and causes errors. Furthermore, if some obvious overlapped shadows appear, then the recognition rate is dropped dramatically as well. According to our identification method, while the hand and its shadow overlap, it will be recognised as the gesture that a finger touches the desk top. In our four experimenters, all fingers have very similar skin tones, the colour differences are not significant. Experimenting with multiple skin tones has not been considered in our experiments.

Our algorithm has been optimised. Several failing frames appear in the process of image binarization compared with the normal segmentation. Therefore, it is necessary to set a threshold to improve the recognition rate by removing those video frames. Overall, the proposed method is not “smart” enough to achieve high recognition accuracy.

We noticed that the longer the Morse code, the lower the success rate of input, because each Morse code enter is entirely recorded. However, each group of Morse codes was recognised only after the finger had left the screen. A long interleave indicates that the finger has left the field of view for a long time, resulting in a decline in the recognition

rate. For the procedure of fingertip tracking, the fingertip needs to be positioned every time. If not, it leads to track failure, thus affects the accuracy of tracking.

In the SVM-based Morse code recognition, in Figure 4.7, Figure 4.13 and Figure 4.19, we see that the performance of the linear kernel function is ordinary, the recognition rate is from 52% to 68%. From the overall ROC curve in Figure 4.8, Figure 4.14 and Figure 4.20, the classification is not good. In Figure 4.9, Figure 4.15 and Figure 4.21, we see that the recognition rate is 75% by using a polynomial kernel function. From the ROC curve in Figure 4.10, Figure 4.16 and Figure 4.22, the whole curve is close to the middle and upper left, this indicates that the classification is better than using the linear kernel function. From Figure 4.11, Figure 4.17 and Figure 4.23, we see that the SVM classifier of the RBF kernel function has a recognition rate up to 90%, which has the best precision of all three methods. With regard to BPNN classifier, the overall recognition rate is 80%, the result is lower than SVM, the reason is that we did not train enough features.

For the SVM-based recognition, the recognition rate highly depends on the selection of training samples. The training samples therefore are possible to improve the recognition rate. If the sample selection is poor to calculate the cross-correlation coefficient, this appears when there has not a touch on the desk surface, after the features are extracted, the identification will then be failed. In order to avoid this situation, we add a tag in the SVM so as to improve the classification, such as a genetic algorithm for global optimisation which will improve the recognition rate.

If we compare the threshold-based recognition algorithm and the SVM/BPNN algorithms, the threshold-based recognition algorithm in our experiments reached up to a 60% recognition rate, meanwhile the SVM/BPNN algorithm obtained the rate of 90%. We have a lot of experimental evidence to show that machine learning algorithms are more accurate and effective than the traditional algorithms.

5.3 Justifications

The primary purpose of this thesis is to design a finger gesture recognition. For finger gesture recognition, we use our finger to tap or slide on the desk top to represent Morse code “dots” and “dashes”. In the finger gestures, we came up with two sets of gesture

segmentation tests. In comparison, one is the segmentation-based algorithm in YCbCr colour space, and the other is ROI based by using a Gaussian pyramid model with an SVM. Experimental results showed the Gaussian pyramid model with the SVM has better performance than segment-based algorithm in YCbCr colour space. Thus, we adopted the Gaussian pyramid model for gesture feature extraction and SVM/BPNN for classification. The overall recognition rate reached up to 90% which is a satisfactory result that meets our expectation.

We thus answer the research questions of this thesis as follows:

Question 1. Can single camera-based precise gesture recognition be achieved?

The answer is yes, but with conditions. Our experiments show that in the acceptance environment, the recognition rate of a single camera on the finger gestures can reach more than 90%. But because of limitations, we did not work for any experiments on the gesture recognition with the complex background. If this research project is possible to be continued, we will conduct a series of experiments on gesture recognition for Morse codes in a complex environment.

Question 2 Is machine learning better than traditional methods?

Machine learning is better than those traditional methods. In our experiments, the recognition rate of the traditional methods of is about 60% in a fixed testing environment. As long as the training set is well selected, the recognition rate can reach more than 90% in machine learning.

5.4 Limitations

In this project, we have three restrictions. Firstly, in our gesture recognition based on a simple background, we do not take account gesture recognition in sophisticated background. In a stable settings, the lighting and background will not be changed.

Secondly, the angle of a camera pointed in the finger is fixed, which means that we did not test further multiple perspectives of the gesture recognition. If so, we need create a very big training set.

Thirdly, we only used the basic 26 letters (A-Z), 10 numbers (0-9), 18 punctuations and six SMS/emoticons for the finger gesture recognition. We did not take a completed character set, such as Chinese characters, SMS, Emoticons, and SOS symbols into consideration because it will need a very large character set.

5.5 Discussions

We have processed the video frames by using two kinds of algorithms for Morse code recognition. We see that the SVM algorithm has better adaptability and has improved the recognition accuracy, but there are still some shortcomings:

- In the selection of training samples or training features, there is no alternative way to choose; each time they need to be manually selected, and the results selected will affect the accuracy of finger gesture recognition.
- In the classifier selection, we use the SVM algorithm to achieve the nonlinear classification. The kernel function is chosen to use the strong RBF kernel function, but whether it is suitable for use in another kind of scene is worth discussing.
- In this thesis, we do not optimise the parameters of the SVM. Although the algorithm is converged to get the two most important parameters of the hyperplane, we do not know whether the global optimal solution or local optimal solution is adopted. In considering whether to optimise the SVM parameters with global optimisation algorithms, this would be used to improve the recognition rate and reduce training time.

5.6 Recommendations

Although a machine learning algorithm is utilised to solve the problem of image segmentation, there is still a gap in real-time algorithms from the experimental results. The recognition depends on the choice of samples, so the algorithm still has limitations. A lot of research work has been carried out in human behaviour recognition, such as 3D SIFT-based recognition algorithms, where these algorithms are based on the ideal environment; the actual situation is not practical, The future work is to study a realtime-based gesture recognition algorithm which is strongly robust.

The recommendation is the fully convolutional networks (FCN). The convolution neural network (CNN) has made great achievements and there are extensive applications in image classification and image detection since 2012. CNN in multi-layer structures can automatically learn several layers of features. The convolutional layer in perception domain is small, and the CNN can learn local area characteristics; the deep convolution layer has a larger perceptual domain that can learn more abstract features. These features are less sensitive to the size, position, and orientation of the object, thereby helping to identify performance improvements. These features are very helpful in classification to determine what kind of object is included in an image because the details of objects are missing. The specific contour belonging to the object cannot be distinguished well, so it is very difficult to achieve an accurate division. In view of this problem, Jonathan Long et al. presented the FCN (Long, Shelhamer & Darrell, 2015) for image segmentation. The network covers the category that each pixel belongs from the abstract features. Also, known as the classification from the image level, it is further extended to the pixel level classification. Due to the characteristics of the FCN, using this method to segment finger gestures from complex environments is possible.

Chapter 6 Conclusion and Future Work

6.1 Conclusion

The purpose of this thesis is to identify the gestures represented Morse codes from a single camera. Although the Morse code is a bit slow dropping out of the communications, Morse codes are still used by people in some circumstances today. Although there are many algorithms in gesture recognition, the proposed approach of this thesis is rarely found. In this thesis, the finger gesture recognition of Morse code based on a single camera is developed, with more than 85% recognition rate, which is a very decent achievement. During the experiments, we have introduced two different algorithms to achieve gesture recognition, mainly studying a SVM-based gesture recognition, where three kernel functions were used and compared in the SVM. The results show that by using the RBF kernel function that the classification rate is up to 92%. By using BPNN algorithm, the best classification rate is up to 82%.

6.2 Future Work

The future development is infinite. For an example, we can use the most popular deep learning technology to identify finger gestures. This can achieve the recognition in a more complex environment, such as real-time gestures used in any context. The semantic segmentation algorithm based on image understanding can solve the problem effectively. That is simply to say, let the machine know the exact meaning of the finger gestures happened before the camera.

References

- Adachi, T., Furuya, R., Greene, S., & Mikuriya, K. (1991). Feature selection for neural network recognition. *International Joint Conference on In Neural Networks*, (pp. 696-701).
- Amma, K., Yaguchi, Y., Niitsuma, Y., Matsuzaki, T., & Oka, R. (2013). A comparative study of gesture recognition between RGB and HSV colors using time-space continuous dynamic programming. *International Joint Conference on Awareness Science and Technology and Ubi-Media Computing*, (pp. 185-191). IEEE.
- Anderson, B. D., & Moore, J. B. (1979). Optimal filtering. Englewood Cliffs, 21, pp.22-95.
- Anthony, D., Hines, E., Barham, J., & Taylor, D. (1990). A comparison of image compression by neural networks and principal component analysis. *In International Joint Conference on Neural Networks*, (pp. 339-344). IEEE.
- Azoz, Y., Devi, L., & Sharma, R. (1998). Reliable tracking of human arm dynamics by multiple cue integration and constraint fusion. *In Conference on Computer Vision and Pattern Recognition, Computer Society*, (pp. 905-910). IEEE.
- Bérci, N., & Szolgay, P. (2009). Towards a gesture based human-machine interface: Fast 3D tracking of the human fingers on high speed smart camera computers. *In International Symposium on Circuits and Systems*, (pp. 1217-1220). IEEE.
- Bergerman, M., Lee, C., & Xu, Y. (1995). Experimental study of an underactuated manipulator. *In International Conference on Intelligent Robots and Systems 95. 'Human Robot Interaction and Cooperative Robots', Proceedings. (2)*, (pp. 317-322). IEEE.
- Billon, R., Nedelec, A., & Tisseau, J. (2008). Gesture recognition in flow based on PCA analysis using multiagent system. *In Proceedings of the 2008 International Conference on Advances in Computer Entertainment Technology*, (pp. 139-146). ACM.

- Birk, H., Moeslund, T. B., & Madsen, C. B. (1997). Real-time recognition of hand alphabet gestures using principal component analysis. *In Proceedings of the Scandinavian Conference on Image Analysis*, (pp. 261-268).
- Bo, K. (1982). Human-Computer Interaction: Guest Editor's Introduction. *In Computer* 15(11).
- Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (6), pp. 679-698.
- Cao, W. M., Lu, F., Gu, Y. B., Peng, H., & Wang, S. (2004). Study of human face recognition based on principal component analysis (PCA) and direction basis function neural networks. *In Fifth World Congress on Intelligent Control and Automation*, (5, pp. 4195-4198). IEEE.
- Chan, T. M. (2000). Approximating the diameter, width, smallest enclosing cylinder, and minimum-width annulus. *In Proceedings of the sixteenth annual symposium on Computational geometry*, (pp. 300-309). ACM.
- Charniak, E. (1993). Statistical Language Learning. *In Cambridge: MIT press*. (pp. 39-53) Cambridge, Massachusetts, London, England.
- Chelali, F. Z., Cherabit, N., & Djeradi, A. (2015). Face recognition system using skin detection in RGB and YCbCr color space. *In World Symposium on Web Applications and Networking* (pp. 1-7). IEEE.
- Cootes, T. F., Taylor, C. J., Cooper, D. H., & Graham, J. (1995). Active shape models-their training and application. *Computer Vision and Image Understanding*, 61(1), pp. 38-59.
- Cootes, T. F. & Taylor, C. J. (1992). Active Shape Models — ‘Smart Snakes’. *In Proceedings of the British machine vision conference* (pp. 266-275).
- Copot, C., Syafiie, S., Vargas, S., De Keyser, R., Van Langenhove, L., & Lazar, C. (2009). Carpet wear classification based on support vector machine pattern recognition approach. *In International Conference on Intelligent Computer Communication and Processing*, (pp. 161-164). IEEE.

- Cristianini, N. & Shawe, T. J. (2000). An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods. In *Cambridge University Press*, Cambridge.
- Davis, J., & Shah, M. (1994). Recognizing hand gestures. In *Computer Vision—ECCV'94*, (pp. 331-340).
- Davis, J. & Shah, M (1993) Gesture recognition. In *University of central*.
- Deepak, R., Nayak, A. V., & Manikantan, K. (2016). Ear detection using active contour model. In *International Conference on Emerging Trends in Engineering, Technology and Science*, (pp. 1-7). IEEE.
- Dix, A. (2009). Human-computer interaction. In *Encyclopedia of database systems*, (pp. 1327-1331). Springer US.
- Erden, F., & Çetin, A. E. (2014). Hand gesture based remote control system using infrared sensors and a camera. *IEEE Transactions on Consumer Electronics*, 60(4), pp. 675-680.
- Fels, S. S., & Hinton, G. E. (1993). Glove-talk: A neural network interface between a data-glove and a speech synthesizer. *IEEE Transactions on Neural Networks*, 4(1), pp. 2-8.
- Feynman, R. Leighton, R. & Sands, M. (1970) The Feynman Lectures on Physics. In *Electrostatic Analogs*, 12(2).
- Gadea, C., Ionescu, B., Ionescu, D., Islam, S., & Solomon, B. (2012). Finger-based gesture control of a collaborative online workspace. In *International Symposium on Applied Computational Intelligence and Informatics*, (pp. 41-46). IEEE.
- Gonzalez, R. C. & Wood R. E. (2005) Digital Image Processing. In *House of Electronics Industry*, (pp. 134-137)
- Grobel, K., & Assan, M. (1997). Isolated sign language recognition using hidden Markov models. In Systems, Man, and Cybernetics, In *International Conference on Computational Cybernetics and Simulation*. (pp. 162-167). IEEE.

- Ham, Y. C., & Shi, Y. (2009). Developing a smart camera for gesture recognition in HCI applications. *In International Symposium on Consumer Electronics*, (pp. 994-998). IEEE.
- Heap, T., & Samaria, F. (1995). Real-time hand tracking and gesture recognition using smart snakes. *Proc. Interface to Human and Virtual Worlds*, Montpellier, France, 50.
- Jia, J., Jiang, J., & Wang, D. (2008). Recognition of hand gesture based on Gaussian mixture model. *In International Workshop on Content-Based Multimedia Indexing*, (pp. 353-356). IEEE.
- Ji, S., Xu, W., Yang, M., & Yu, K. (2013). 3D convolutional neural networks for human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1), pp. 221-231.
- Kasson, J. M., & Plouffe, W. (1992). An analysis of selected computer interchange color spaces. *ACM Transactions on Graphics (TOG)*, 11(4), pp. 373-405.
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *In Advances in neural information processing systems*, (pp. 1097-1105).
- Kim, H. J., Lee, J. S., & Park, J. H. (2008). Dynamic hand gesture recognition using a CNN model with 3D receptive fields. *In International Conference on Neural Networks and Signal Processing*, (pp. 14-19). IEEE.
- Kim, M. S., Kim, D., & Lee, S. Y. (2003). Face recognition using the embedded HMM with second-order block-specific observations. *Pattern Recognition*, 36(11), pp. 2723-2735.
- Kumar, S., & Segen, J. (1999). Gesture based 3d man-machine interaction using a single camera. *In International Conference on Multimedia Computing and Systems*, (pp. 630-635). IEEE.
- LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), pp. 2278-2324.

- Lee, L. K., & Liew, S. C. (2015). Breast ultrasound automated ROI segmentation with region growing. *In International Conference on Software Engineering and Computer Systems*, (pp. 177-182). IEEE.
- Luo, D., & Ohya, J. (2010). Study on human gesture recognition from moving camera images. *In International Conference on Multimedia and Expo*, (pp. 274-279). IEEE.
- Liang, R. H., & Ouhyoung, M. (1995). A Real - time Continuous Alphabetic Sign Language to Speech Conversion VR System. *In Computer Graphics Forum* 14(3), (pp. 67-76).
- Liu, L., & Fan, G. (2003). A new JPEG2000 region-of-interest image coding method: Partial significant bitplanes shift. *IEEE Signal Processing Letters*, 10(2), pp. 35-38.
- Lindeberg, T. (1994). Scale-space theory: A basic tool for analyzing structures at different scales. *Journal of applied statistics*, 21(1-2), pp. 225-270.
- Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 3431-3440).
- Lowe, D. G. (1999). Object recognition from local scale-invariant features. *In International Conference on Computer Vision, The proceedings of the seventh IEEE* (pp. 1150-1157). IEEE.
- Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2), pp. 91-110.
- Lu, D., Yu, Y., & Liu, H. (2016). Gesture recognition using data glove: An extreme learning machine method. *In International Conference on Robotics and Biomimetics* (pp. 1349-1354). IEEE.
- Marr, D. & Hildreth, E. C. (1980) Theory of edge detection. *In Proceedings of the Royal Society B*, pp. 187-217

- Martin, J., & Crowley, J. L. (1997). An appearance-based approach to gesture-recognition. *In International Conference on Image Analysis and Processing* (pp. 340-347). Springer, Berlin, Heidelberg.
- Megiddo, N. (1983). Linear-time algorithms for linear programming in R^3 and related problems. *SIAM journal on computing*, 12(4), pp. 759-776.
- Mitchell, T. M. (1997). Machine learning. 1997. Burr Ridge, IL: McGraw Hill, 45(37), pp. 870-877.
- Murakami, K., & Taguchi, H. (1991). Gesture recognition using recurrent neural networks. *In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 237-242). ACM.
- Narkhede, P. R., & Gokhale, A. V. (2015). Color image segmentation using edge detection and seeded region growing approach for CIELab and HSV color spaces. *In International Conference on Industrial Instrumentation and Control* (pp. 1214-1218). IEEE.
- Negri, R. G., Sant'Anna, S. J. S., & Dutra, L. V. (2013). A new contextual version of Support Vector Machine based on hyperplane translation. *In International Geoscience and Remote Sensing Symposium* (pp. 3116-3119). IEEE.
- Noda, H., Niimi, M., & Korekuni, J. (2006). Simple and efficient colorization in YCbCr color space. *In 18th International Conference on Pattern Recognition*, (3, pp. 685-688). IEEE.
- O'Toole, A. J., Millward, R. B., & Anderson, J. A. (1988). A physical system approach to recognition memory for spatially transformed faces. *Neural Networks*, 1(3), pp. 179-199.
- Ouhyoung, M., & Liang, R. H. (1996). A sign language recognition system using hidden markov model and context sensitive search. *In Conference Procs. of ACM Virtual Reality Software and Technology* (pp. 59-66).

- Osher, S., & Sethian, J. A. (1988). Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations. *Journal of computational physics*, 79(1), pp. 12-49.
- Pavlovic, V. I., Sharma, R., & Huang, T. S. (1997). Visual interpretation of hand gestures for human-computer interaction: A review. *IEEE Transactions on pattern analysis and machine intelligence*, 19(7), pp. 677-695.
- Popescu, B., Iancu, A., Brezovan, M., & Burdescu, D. D. (2010). Boundary-Based Measures for Evaluation of Color Image Segmentation. *In International Conferences on Advances in Multimedia* (pp. 162-167). IEEE.
- Russell, S., Norvig, P., & Intelligence, A. (1995). A modern approach. *Artificial Intelligence*. Prentice-Hall, Englewood Cliffs, 25, 27.
- Rautaray, S. S., & Agrawal, A. (2015). Vision based hand gesture recognition for human computer interaction: a survey. *Artificial Intelligence Review*, 43(1), pp. 1-54.
- RongQing, Y., WenHui, L., Duo, W., & Hua, Y. (2008). Feature Recognition Based on Graph Decomposition and Neural Network. *In Third International Conference on Convergence and Hybrid Information Technology* (pp. 864-868). IEEE.
- Sahoo, P. K., Soltani, S. A. K. C., & Wong, A. K. (1988). A survey of thresholding techniques. *Computer vision, graphics, and image processing*, 41(2), pp. 233-260.
- Sapaico, L. R., & Sato, M. (2011). Analysis of vision-based Text Entry using Morse code generated by tongue gestures. *In 4th International Conference on Human System Interactions*, (pp. 158-164). IEEE.
- Schlenzig, J., Hunter, E., & Jain, R. (1995). Recursive spatio-temporal analysis: Understanding gestures. Technical Report VCL-95-109, Visual Computing Laboratory, University of California, San Diego.
- Sharma, R., Pavlovic, V. I., & Huang, T. S. (1998). Toward multimodal human-computer interface. *Proceedings of the IEEE*, 86(5), pp. 853-869.

- Shi, Y., Taib, R., & Lichman, S. (2006). GestureCam: a smart camera for gesture recognition and gesture-controlled web navigation. *In 9th International Conference on Control, Automation, Robotics and Vision*, (pp. 1-6). IEEE.
- Sirovich, L., & Kirby, M. (1987). Low-dimensional procedure for the characterization of human faces. 4(3), pp. 519-524.
- Soni, N. S., Nagmode, M. S., & Komati, R. D. (2016). Online hand gesture recognition & classification for deaf & dumb. *In International Conference on Inventive Computation Technologies*. (pp. 1-4). IEEE.
- Starner, T. E. (1995). Visual Recognition of American Sign Language Using Hidden Markov Models.
- Suarez, J., & Murphy, R. R. (2012). Hand gesture recognition with depth images: A review. *In Ro-man* (pp. 411-417). IEEE.
- Sun, Y., Chen, Y., Wang, X., & Tang, X. (2014). Deep learning face representation by joint identification-verification. *In Advances in neural information processing systems* (pp. 1988-1996).
- Tairi, Z. H., Rahmat, R. W., Saripan, M. Q. & Sulaiman, P. S. (2014) Skin Segmentation Using YUV and RGB Color Spaces. *Korea Information Processing Society*. 10(2) (pp. 283-299).
- Takahashi, T., & Kishino, F. (1991). Hand gesture coding based on experiments using a hand gesture interface device. *ACM SIGCHI Bulletin*, 23(2), pp. 67-74.
- Triantafyllidou, D., & Tefas, A. (2016,). Face detection based on deep convolutional neural networks exploiting incremental facial part learning. *In 23rd International Conference on Pattern Recognition* (pp. 3560-3565). IEEE.
- Tremeau, A., & Borel, N. (1997). A region growing and merging algorithm to color segmentation. *Pattern recognition*, 30(7), pp. 1191-1203.
- Usabiaga, J., Erol, A., Bebis, G., Boyle, R., & Twombly, X. (2009). Global hand pose estimation by multiple camera ellipse tracking. *Machine Vision and Applications*, 21(1), pp. 1-15.

- Vapnik, V. N. (1995). The Nature of Statistical Learning Theory. In *Statistics for Engineering and Information Science*. pp. 227-229. New York.
- Vapnik, V. (1963). Pattern recognition using generalized portrait method. *Automation and remote control*, 24, pp. 774-780.
- Wang, P., Li, W., Liu, S., Zhang, Y., Gao, Z., & Ogunbona, P. (2016). Large-scale continuous gesture recognition using convolutional neural networks. In *International Conference on Pattern Recognition* (pp. 13-18). IEEE.
- WANG, Y. J., CHEN, X. H., & ZOU, L. (2007). Automatic Extraction of Regions of Interest [J]. *Science Technology and Engineering*, 12, pp. 28.
- Weissmann, J., & Salomon, R. (1999). Gesture recognition for virtual reality applications using data gloves and neural networks. In *International Joint Conference on Neural Networks* (pp. 2043-2046). IEEE.
- Wren, C. R., Azarbayejani, A., Darrell, T., & Pentland, A. P. (1997). Pfindex: Real-time tracking of the human body. *IEEE Transactions on pattern analysis and machine intelligence*, 19(7), pp. 780-785.
- Wu, Y., & Huang, T. S. (1999). Vision-based gesture recognition: A review. In *Gesture Workshop* (pp. 103-115).
- Yu, H., Yuanxin, Z., Guangyou, X., Hui, Z., Zhen, W., & Haibin, R. (1998). Video camera-based dynamic gesture recognition for HCI. In *Fourth International Conference on Signal Processing Proceedings* (pp. 904-907). IEEE.
- Zaslow, J. & Pausch, R. (2008) The Last Lecture. United States.
- Zhang, Y. J. (2001). A review of recent evaluation methods for image segmentation. In *Sixth International Symposium on Signal Processing and its Applications*, (pp. 148-151). IEEE.
- Zivkovic, Z., Kliger, V., Kleihorst, R., Danilin, A., Schueler, B., Arturi, G., & Aghajan, H. (2008). Toward low latency gesture control using smart camera network. In *Computer Society Conference on Computer Vision and Pattern Recognition Workshops* (pp. 1-8). IEEE.

Appendix

Table I Morse codes of English letters

English Letters	Morse Codes	SVM-based kernel functions			BPNN Algorithm
		Linear	Polynomial	RBF	
A	.-	0.68	0.82	0.92	0.8
B	0.52	0.67	0.85	0.81
C	0.65	0.75	0.92	0.83
D	...	0.69	0.71	0.9	0.76
E	.	0.64	0.69	0.89	0.79
F	0.68	0.76	0.89	0.77
G	...	0.65	0.75	0.9	0.83
H	0.55	0.72	0.87	0.78
I	..	0.64	0.77	0.93	0.71
J	0.57	0.71	0.88	0.85
K	...-	0.68	0.8	0.91	0.82
L	0.55	0.69	0.86	0.82
M	--	0.59	0.73	0.88	0.81
N	.-	0.64	0.78	0.88	0.82
O	---	0.55	0.72	0.85	0.78
P	0.51	0.81	0.87	0.81
Q	0.56	0.76	0.86	0.83
R	...-	0.59	0.71	0.9	0.81
S	...	0.57	0.78	0.89	0.76
T	-	0.53	0.68	0.88	0.69
U	...-	0.61	0.71	0.91	0.73
V-	0.68	0.74	0.91	0.75
W	...-	0.55	0.74	0.88	0.82
X-	0.57	0.69	0.9	0.76
Y-	0.57	0.77	0.89	0.77
Z	0.64	0.8	0.91	0.73

Table II Morse codes of Arabic numbers

Arabic Numbers	Morse Codes	SVM-based kernel functions			BPNN Algorithm
		Linear	Polynomial	RBF	
1	-----	0.6	0.7	0.89	0.82
2	0.61	0.72	0.87	0.8
3	0.55	0.76	0.86	0.75
4--	0.58	0.72	0.89	0.85
5	0.55	0.7	0.87	0.75
6	0.6	0.7	0.9	0.77
7--	0.55	0.71	0.88	0.73
8	0.56	0.73	0.9	0.82
9	0.54	0.7	0.86	0.79
0	-----	0.55	0.71	0.88	0.81

Table III Morse codes of punctuations

Punctuations	Morse Codes	SVM-based kernel functions			BPNN Algorithm
		Linear	Polynomial	RBF	
Period (.)	0.54	0.75	0.9	0.74
Question mark (?)	0.6	0.71	0.83	0.79
Exclamation mark (!)	0.57	0.7	0.86	0.79
Parenthesis (())	0.54	0.69	0.91	0.76
At (@)	0.53	0.74	0.89	0.72
Double quotes (")	0.57	0.7	0.85	0.76
Comma (,)	0.57	0.69	0.9	0.84
Apostrophe (')	0.61	0.77	0.88	0.81
Underline (_)	0.5	0.69	0.9	0.79
Colon (:)	0.56	0.6	0.86	0.75
Equal sign (=)	0.55	0.73	0.86	0.8
Hyphen (-)	0.58	0.71	0.91	0.82
After parenthesis ())	0.55	0.69	0.87	0.8
Semicolon (;)	0.59	0.7	0.86	0.71
And (&)	0.61	0.73	0.88	0.8
Dollar (\$)	0.6	0.66	0.87	0.73
Slash (/)	0.53	0.6	0.86	0.79
Space	..--	0.62	0.72	0.92	0.76

Table IV Morse codes of SMS and emoticons

SMS Emoticons	Morse Codes	SVM-based kernel functions			BPNN Algorithm
		Linear	Polynomial	RBF	
SOS	·····	0.52	0.69	0.8	0.76
How are you?	·····	0.5	0.6	0.82	0.8
See you	·····	0.49	0.6	0.86	0.81
LOL	·····	0.51	0.73	0.86	0.82
😞	·····	0.56	0.76	0.88	0.76
😞	·····	0.55	0.68	0.8	0.75
😏	·····	0.51	0.66	0.87	0.8
🙄	·····	0.53	0.69	0.86	0.77

Table V Morse codes of mathematical symbols

Mathematical Symbols	Morse Codes	SVM-based kernel functions			BPNN Algorithm
		Linear	Polynomial	RBF	
$\sqrt{\quad}$.-.-.	0.59	0.68	0.89	0.74
\times	-.-..	0.54	0.69	0.86	0.76
Σ	..-..	0.55	0.73	0.89	0.74
α	-.-..	0.65	0.73	0.81	0.76
Π	.-.-.	0.48	0.6	0.9	0.78
∞	-...-.-.	0.54	0.69	0.89	0.71
\approx	.-.-.	0.55	0.65	0.9	0.81
\div	.-.-.	0.56	0.61	0.88	0.8
\approx	..-.-.	0.51	0.63	0.8	0.76
\int	-..-.-.	0.54	0.64	0.86	0.77
β	-.-.-.	0.55	0.66	0.85	0.76
\bowtie	-.-..	0.53	0.66	0.89	0.76
ω	...-.-.	0.58	0.6	0.88	0.79
\prec	..-.-.	0.56	0.67	0.84	0.77
\emptyset	----.-.	0.54	0.68	0.86	0.78
\propto	----.-.	0.52	0.68	0.87	0.75
\exists	.-.-.-.-.	0.49	0.67	0.89	0.79
∇	-.-.-.-.	0.58	0.69	0.87	0.8
\cup	-.-.-..	0.55	0.64	0.9	0.8
\in	.-.-.-.-.	0.53	0.71	0.86	0.77
\equiv	-...-.-.-.	0.53	0.71	0.9	0.76

Table VI Morse codes of some Chinese characters

Chinese Characters	Codes of Chinese Characters	Morse Codes	SVM-based kernel functions			BPNN Algorithm
			Linear	Polynomial	RBF	
啊	0001	.-----	0.45	0.66	0.86	0.73
阿	0002	..-----	0.59	0.68	0.86	0.77
鱈	0087	-----...	0.49	0.65	0.8	0.73
藹	0100	-----.	0.57	0.67	0.81	0.8
矮	0101	.----.---	0.52	0.6	0.85	0.71
艾	0102	..---.---	0.64	0.72	0.88	0.76
鞅	0187	---..---.---	0.48	0.71	0.83	0.73
按	0200	-----.---	0.62	0.68	0.86	0.79
暗	0201	.----.---	0.52	0.69	0.88	0.71
岸	0202	..---.---	0.53	0.65	0.9	0.78
骷	0287	--..---.---	0.61	0.69	0.82	0.79
鰲	8700	-----..--.	0.55	0.66	0.87	0.77
鰻	8701	.-----..--.	0.61	0.69	0.82	0.78
鰻	8702	..-----..--.	0.48	0.6	0.86	0.8
黜	8787	---..---.---	0.52	0.69	0.81	0.73