國立交通大學

電控工程研究所

碩士論文

智慧型字卡影像辨識系統設計

Intelligent Word Card Image Recognition System Design

研究生：孫　齊

指導教授：陳永平 教授

中華民國一百零一年六月

智慧型字卡影像辨識系統設計

Intelligent Word Card Image Recognition System Design

研究生：孫齊　　Student : Chi-Sun

指導教授：陳永平　　Advisor : Yon-Ping Chen

國立交通大學

電控工程研究所

碩士論文

A Thesis
Submitted to Institute of Electrical Control Engineering
College of Electrical and Computer Engineering
National Chiao Tung University
In Part Fulfillment off the Requirements
for the Degree of Master
in
Electrical Control Engineering
June 2012/6/12

Hsinchu, Taiwan, Republic of China

中華民國一百零一年六月

智慧型

字卡影像辨識系統設計

學生：孫齊　　　　　　　　指導教授：陳永平 教授

國立交通大學電機與控制工程學系

摘 要

本篇論文的研究目的是希望能製做出一系統，能使童伴機器人與孩童進行互動，並從中學習單字。系統被設計成三個步驟，包含抽取物件的區域、抽取字元以及字元辨識。本篇論文有主要兩個貢獻，第一部分是特定顏色移動物的辨識。第二部分是在不同大小、傾斜以旋轉中的字元辨識。在特定顏色移動物的辨識中，一般所知的是顏色辨識與移動物辨識兩步驟。本篇論文在此使用類神經演算法將一般所知的兩個步驟改成同時進行。此外在針對文字辨識中，我們使用熟知的標楷體來做辨識，此系統可以在不同位置、大小、傾斜程度以及任意的旋轉角度中，達到成功的辨識。最後，三個實驗將會驗證系統的成功開發。

Intelligent Word Card Image Recognition System Design

Student：Chi Sun　　　Advisor：Prof. Yon-Ping Chen

Institute of Electrical Control Engineering

National Chiao-Tung University

ABSTRACT

The purpose of this thesis is to build up a system for children to learn words in an interactive way. The system is designed in three steps, including potential object localization, character extraction and character recognition. The main contribution of this thesis includes two parts, which are the moving object detection and the image recognition. The moving object is an word card in special color and the image to be recognized is a character. To detect the moving word card from a sequence of images, an artificial neural network is proposed to extract the color and detect the word card simultaneously. After the word card is detected, a scheme based on a set of concentric circles is applied to extract the features of the character on the word card. From the features, another artificial neural network is designed for character recognition invariant to the translation, rotation and scaling of the word card. Finally, the success of the developed system is verified by three experiments.

Acknowledgement

本篇論文得以完成，首先必須誠摯感謝指導教授 陳永平老師在這兩年中的悉心指導與教誨。老師嚴謹的治學態度，理論與實務並重的訓練，使得本論文得以順利完成。除了學術上的指導，在待人處事方面的啟發更是讓我獲益良多，這份師恩會令我永生難忘。同時也感謝　林進燈老師與楊谷洋老師對本論文所提出的珍貴意見與指正，使得本論文得以更佳的完整。

此外，感謝可變結構控制實驗室的世宏學長、桓展、文榜、澤翰與文俊學長在平日攻讀學位之餘，不吝嗇傳授知識與經驗及給予建議。同時，感謝實驗室同學榮哲、振方、崇賢、咨瑋以及學弟谷穎、宣俊、仕政、兆村在課業與研究上一起學習、勉勵以及協助使我收穫良多。

最後，感謝這一路相輔相持的慈珊、詩屏、振綱與玉青，你們在生活中帶給我的許多歡樂，使我生活多采多姿，為我兩年的研究生活帶來許多歡樂及回憶。更感謝我的母親、大哥、二哥，你們的關心與鼓勵，給了我許多的溫暖，由於你們的支持，使我能專心研究。

僅以此篇論文獻給所有關心我、照顧我的人，你們的恩惠我銘感於心，由衷的感謝你們。

孫齊　2012.7

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

In recent years, our laboratory research gradually developed. There are two major parts in our laboratory researches, intelligent machine learning [1],[2] and application of kinder robot.

There are a lot of algorithms used in detection and recognition systems. Expect the higher performance, the algorithm will be more complex and system rules will be harder to find. Artificial neural networks (ANNs) are intelligent machine learning systems that are deliberately constructed to make use of some organizational principles resembling those of the human brain, which can learn to find out the correct results and make the system easier to implement and extent.

The other part in our laboratory research is the application of kinder robot. As the technicalization of children's products gradually extracts attention, how to achieve better educational result when using children's products with electronic technique becomes more important.

Like several thesis in our laboratory research, "Intelligent human detection system design based on depth information" and "Design of Wireless-Based Remote Interaction System Applied to Remote Surveillance", etc. This thesis is to design a system by using the results of the three seniors' work. We combine the color extraction in the face recognition system, the identification of the moving objects in the pedestrian recognition system and the character recognition in the license plate recognition system. The purpose of this thesis is to build up a system for children to learn words in an interactive way.

1.2 System Overview

For hardware architecture, the system shown in Fig. 1.1 is established by setting two cameras on a horizontal line and their lines of vision are parallel and fixed. In addition, the distance between two cameras is set as constant equal to 10 cm and these two cameras, QuickCam^{TM} Communicate Deluxe, have specification listed below. The experimental environment for testing is our laboratory and the deepest depth of the background is 400 cm.

- 1.3-megapixel sensor with RightLight™2 Technology
- Built-in microphone with RightSound™ Technology
- Video capture: Up to 1280 x 1024 pixels (HD quality) (HD Video 960 x 720 pixels)
- Frame rate: Up to 30 frames per second
- Still image capture: 5 megapixels (with software enhancement)
- USB 2.0 certified
- Optics: Manual focus

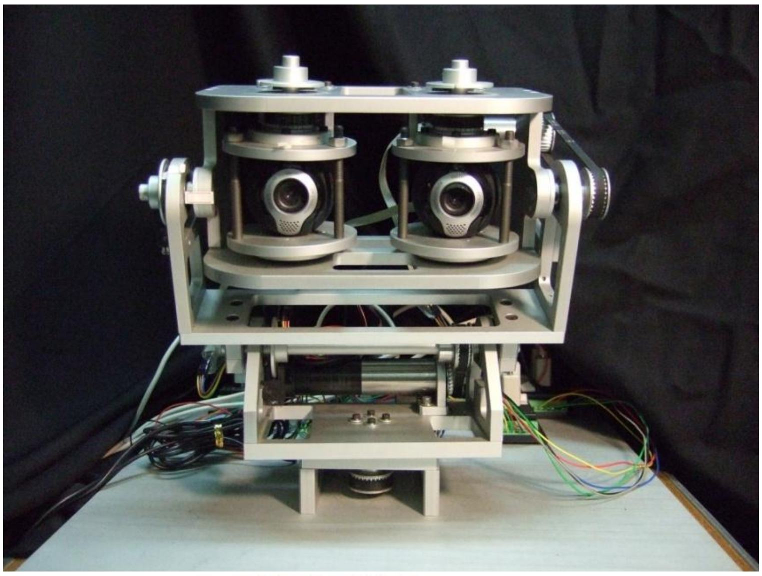


Fig. 1.1 The humanoid vision system.

For software architecture, the image shown in Fig. 1.2 is the flow chart of the proposed system. The system is implemented in three steps, including potential object localization, character extraction and character recognition. The system is designed in three steps, including potential object localization, character extraction and character recognition. In the first step, it is required to detect the moving object, or word card, in special color and then determine the location of the word card in the image. A supervised learning neural network (MCNN) is used to extract the color and detect the moving word card simultaneously. After applying the MCNN, the region of the word card in green color is extracted from a sequence of images; unfortunately, some noise exists therein. Using morphology operation and connected components labeling (CCL), the noise is removed and the region of the word card could be located correctly.

In the second step, use another supervised learning neural network (GNN) to detect the green color of the word card, and then apply the morphology operation to reduce noise. The word card is thus achieved as a binary image with the shape of the character on it. By generating a plain binary card, the character on the word card can be extracted by subtracting the plain binary card. Besides, the total number of pixels of the character is calculated to determine whether the result is a character or not. In the third step, a scheme based on a set of concentric circles is adopted to extract the character features, and then feed the features into the third supervised learning neural network (CRNN) to recognize which word it is, the designed neural networks CRNN can robustly identify characters in different translation, size, tilt and angle of rotation [3]. The overall system processing time is about 0.15s.

Chapter 2 describes the related works of the system. Chapter 3 describes the intelligent word card image recognition system. Chapter 4 shows the experiment results. Chapter 5 is the conclusions of the thesis and the future works.

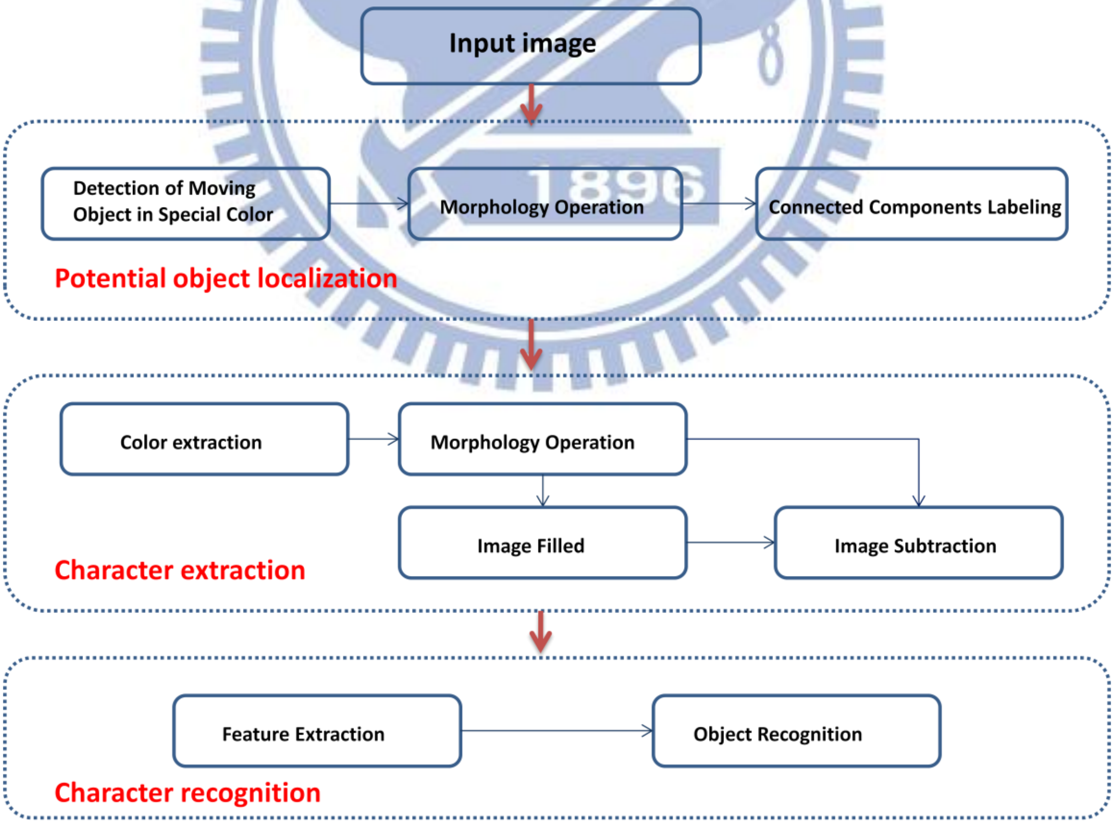


Fig. 1.2 The software architecture.

Chapter 2

Related Work

2.1 Introduction to ANNs

The human nervous system consists of a large amount of neurons, including somas, axons, dendrites and synapses. Each neuron is capable of receiving, processing, and passing signals from one to another. Recently, investigators have developed an intelligent algorithm to mimic the characteristics of the human nervous system, called artificial neural networks (ANNs). In the artificial intelligence field, ANNs have been applied successfully to speech recognition, image analysis and adaptive control. This thesis will apply ANNs to the character recognition in an eyeball system through learning.

-

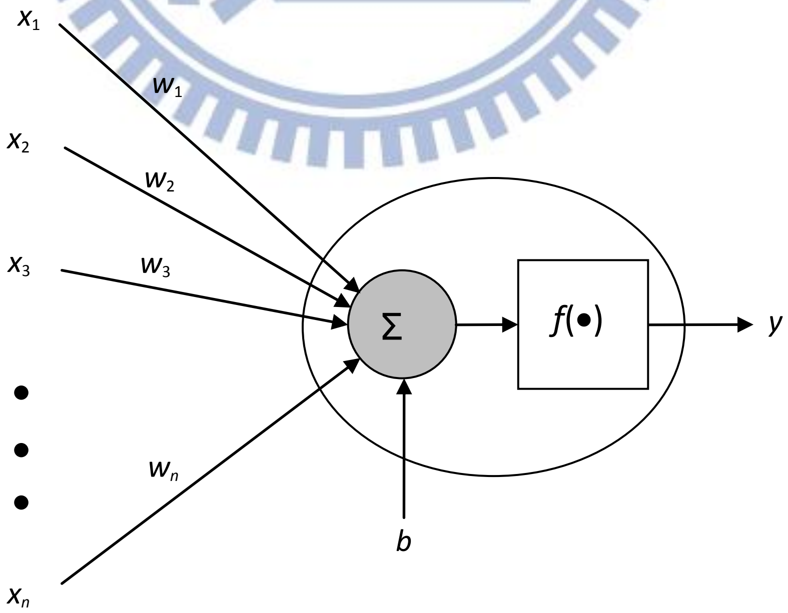


Fig. 2.1 Basic structure of a neuron.

Fig. 2.1 shows the basic structure of a neuron, whose input-output relationship can be described as

$$y = f\left(\sum_{i=1}^{n} w_i x_i + b\right) \qquad (2.1)$$

where x_i and w_i are respectively the i-th input and its weight, b is the bias, and y represents the output. As for the activation function $f(\bullet)$, it can be linear or nonlinear, and its activation level is determined by the sum of $w_i x_i$ and the bias b. Here list three types of the commonest activation functions, called the linear function, log-sigmoid function and tan-sigmoid function, respectively expressed as below:

(1) Linear function

$$f(x) = x \qquad (2.2)$$

(2) log-sigmoid function

$$f(x) = \frac{1}{1+e^{-x}} \qquad (2.3)$$

(3) tan-sigmoid function

$$f(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \qquad (2.4)$$

The structure of a multilayer feed-forward network is generally composed of one input layer, one output layer, and some hidden layers. For example, Fig. 2.2 shows a multilayer feed-forward network with one input layer, one output layer, and two hidden layers. Each layer is formed by neurons with basic structure depicted in Fig. 2.1. The input layer receives signals from the outside world, and then responses through the hidden layers to the output layer. Note that in some cases only the input layer and output layer are required, with the hidden layers omitted.

Compared with the network using single hidden layer, the network with multi-hidden layers can solve more complicated problems. However, its related training process may become more difficult.

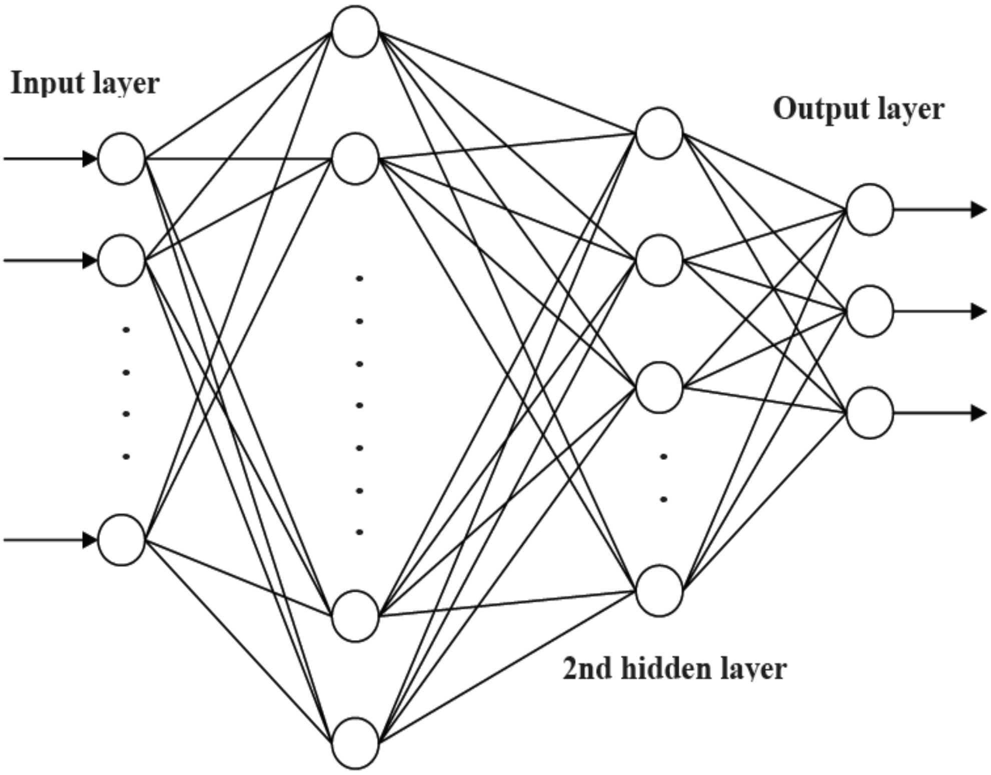


Fig. 2.2 A multilayer feed-forward network with two hidden layers.

In addition to the architecture, the method of setting the values of the weights is important for a neural network, which may be trained via supervised learning or unsupervised learning. Training of supervised learning is mapping a given set of inputs to a specified set of target outputs. The weights are then adjusted according to various learning algorithms. As for the unsupervised learning, the neural network is trained to group similar input vectors together without any training data to specify what a typical member of each group looks like or which group each vector belongs to. In this thesis, the neural network learns the behavior by many input-output pairs, hence that is belongs to supervised learning.

2.2 Back-Propagation Network

The back propagation, BP in brief, was proposed in 1986 by Werbos, etc. [], which is based on the gradient steepest descent method to update the weights by minimizing the total square error of the output. The BP algorithm has been widely used in a diversity of applications with supervised learning. To clearly explain the BP algorithm, an example is given in Fig. 2.3, a neural network with *I* input nodes, *J* output nodes, and *K* hidden nodes. Let the inputs and outputs be *xi*, and *yj*, where *i*=1,2,…,*I* and *j*=1,2,…,*J*, respective. For the hidden layer, the *k*-th hidden node, *k*=1,2,…,*K*, receives information from input layer and sends out *hk* to the output layer. These three layers are connected by two sets of weights, *vik* and *wkj*. The weight *vik* connects the *i*-th input node to the *k*-th hidden node, while the weight *wkj* connects the *k*-th hidden node to the *j*-th output node.

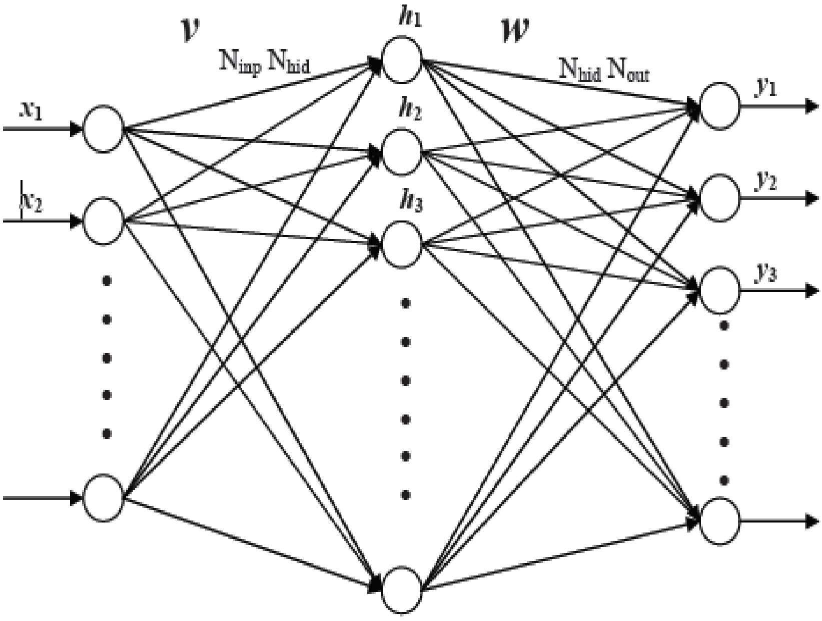


Fig. 2.3 Neural network with one hidden layer.

Based on the neural network in Fig. 2.3, the BP algorithm for supervised learning is generally processed by eight steps as below:

Step 1: Set the maximum tolerable error $E_{\max}$ and then the learning rate η between 0.1 and 1.0 to reduce the computing time or increase the precision.

Step 2: Set the initial weight and bias value of the network randomly.

Step 3: Input the training data, $x = [x_1 \quad x_2 \quad \cdots \quad x_I]^T$ and the desired output $d = [d_1 \quad d_2 \quad \cdots \quad d_J]^T$.

Step 4: Calculate each output of the K neurons in hidden layer

$$h_k = f_h\left(\sum_{i=1}^{I} v_{ik} x_i\right), \qquad k = 1, 2..., K \tag{2.5}$$

where $f_h(\bullet)$ is the activation function, and then each output of the J neurons in output layer

$$y_j = f_y\left(\sum_{k=1}^{K} w_{kj} h_k\right), \qquad j = 1, 2..., J \tag{2.6}$$

where $f_y(\bullet)$ is the activation function.

Step 5: Calculate the following error function

$$E(w) = \frac{1}{2}\sum_{j=1}^{J}(d_j - y_j)^2 = \frac{1}{2}\sum_{j=1}^{J}\left[d_j - f_y\left(\sum_{k=1}^{K} w_{kj} h_k\right)\right]^2 \tag{2.7}$$

where d is the desired output.

Step 6: According to gradient descent method, determine the correction of weights as below:

$$\Delta w_{kj} = -\eta \frac{\partial E}{\partial w_{kj}} = -\eta \frac{\partial E}{\partial y_j} \frac{\partial y_j}{\partial w_{kj}} = \eta \delta_{kj} h_k \tag{2.8}$$

$$\Delta v_{ik} = -\eta \frac{\partial E}{\partial v_{ik}} = -\eta \sum_{j=1}^{J} \frac{\partial E}{\partial y_j} \frac{\partial y_j}{\partial h_k} \frac{\partial h_k}{\partial v_{ik}} = \eta \delta_{ikj} x_i \tag{2.9}$$

where

$$\delta_{kj} = (d_j - y_j)\left[f'_y\left(\sum_{k=1}^{K} w_{kj} h_k \right) \right]$$

$$\delta_{ikj} = \sum_{j=1}^{J} \left[(d_j - y_j) f'_y\left(\sum_{k=1}^{K} w_{kj} h_k \right) w_{kj} \right] f'_h\left(\sum_{i=1}^{I} v_{ik} x_i \right)$$

Step 7: Propagate the correction backward to update the weights as below:

$$\begin{cases} w(n+1) = w(n) + \Delta w \\ v(n+1) = v(n) + \Delta v \end{cases} \tag{2.10}$$

Step 8: Check the next training data. If it exists, then go to Step 3, otherwise, go to Step 9.

Step 9: Check whether the network converges or not. If $E < E_{\max}$, terminate the training process, otherwise, begin another learning circle by going to Step 1.

The maximum tolerable error $E_{\max}$ are the same as error function. Learning rate η is the parameters can change the speed on correction the weights .BP learning algorithm can be used to model various complicated nonlinear functions. Recently, the BP learning algorithm is successfully applied to many domain applications, such as: pattern recognition, adaptive control, clustering problem, etc. In the thesis, the BP algorithm was used to learn the input-output relationship for clustering problem.

2.3 Foreground Segmentation

Dynamic imaging is often the part of interest in the real-time detection system, a good motion detection system to identify moving objects in the picture can get great help for the next classification, or tracking. So a good dynamic detection method can provide more accurate information for follow-up action. There are three common way: background subtraction [7], and optical flow [8], frame difference [9].

Background subtraction is the most common method for segmentation of interesting regions in videos. This method has to build the initial background model firstly. The purpose of training background model is to subtract background image from current image for obtaining interesting foreground regions. Background subtraction method can detect the most complete of feature points of interesting foreground regions and real-time implementation.

Optical flow reflects the image changes due to motion during a time interval, and the optical flow field is the velocity field that represents the three-dimensional motion of foreground points across a two-dimensional image. Compared with other two methods, optical flow can be more accurate to detect interesting foreground region. But optical flow computations are very intensive and difficult to realize in real time.

Frame difference method is to do pixel-based subtraction in successive frames. Its original reasonable is using consistency continuous image background subtraction, image segmentation algorithms such as

$$I_{\mathrm{sub}}(\mathrm{x,y}) = I_{\mathrm{t}}(\mathrm{x,y}) - I_{\mathrm{t-1}}(\mathrm{x,y}) \tag{2.11}$$

$\mathrm{I_{sub}}(\mathrm{x,y})$ is image subtraction matrix， $\mathrm{I_t}(\mathrm{x,y})$ and $\mathrm{I_{t-1}}(\mathrm{x,y})$ are representatives time for RGB color in $\mathrm{I_t}$ and $\mathrm{I_{(t-1)}}$. Application to set up image subtraction threshold

(Sub_{th}) for change detection, when I_{sub}(x,y) level of abnormal larger than this threshold can be regarded as a dynamic pixel, otherwise identified as the background.

$$I_{sub}(x,y)=\begin{cases} 1 & \left|I_t(x,y)-I_{t-1}(x,y)\right| \geq Sub_{th} \\ 0 & \left|I_t(x,y)-I_{t-1}(x,y)\right| < Sub_{th} \end{cases} \tag{2.12}$$

This method can quickly adapt to change of illumination and camera motion and lower computation. This study considers the environment and the processing time and other factors, the use of frame difference as the prospect of capture method.

2.4 Introduction to Morphology Operation

Morphology has two simple functions, dilation and erosion [10]. Dilation is defined as:

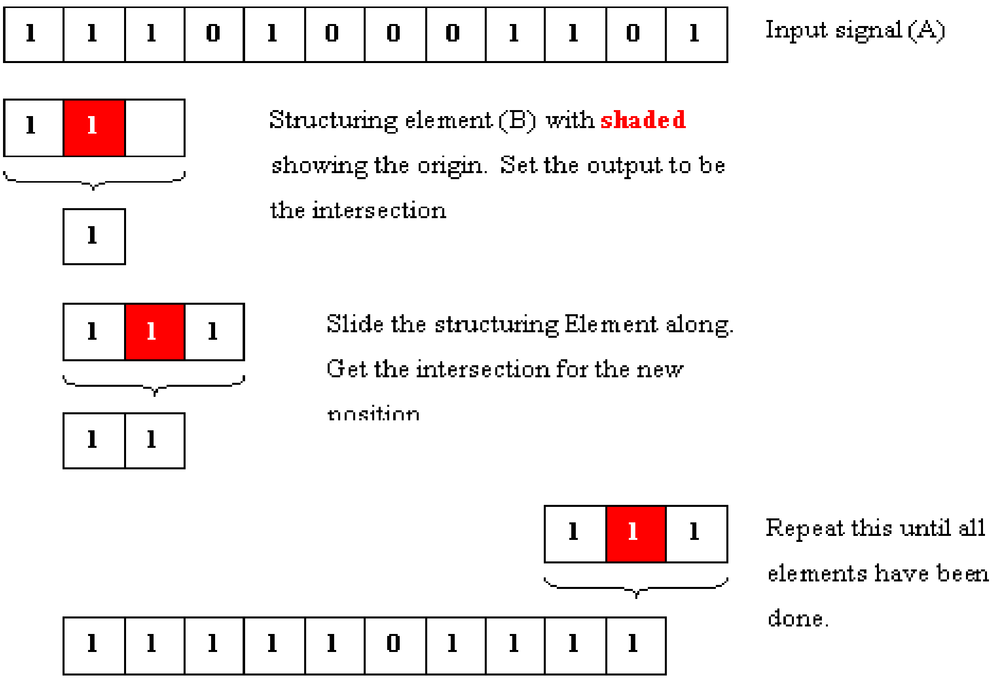
$$A \oplus B = \left\{ x : (\hat{B})_x \cap A \neq \phi \right\} \tag{2.13}$$

where *A* and *B* are sets in *Z*. This equation simply means that *B* is moved over *A* and the intersection of *B* reflected and translated with *A* is found. Usually *A* will be the signal or image being operated on and *B* will be the structuring element. Fig. 2.4 Shows how dilation works.

The opposite of dilation is known as erosion. This is defined as:

$$A \Theta B = \left\{ x : (B)_x \subseteq A \right\} \tag{2.14}$$

which simply says erosion of *A* by *B* is the set of points *x* such that *B*, translated by *x*, is contained in *A*. Fig. 2.5 shows how erosion works. This works in exactly the same way as dilation. However equation (2.2) essentially says that for the output to be a one, all of the inputs must be the same as the structuring element. Thus, erosion will remove runs of ones that are shorter than the structuring element. This thesis will applied two kind of this operation to process the image.

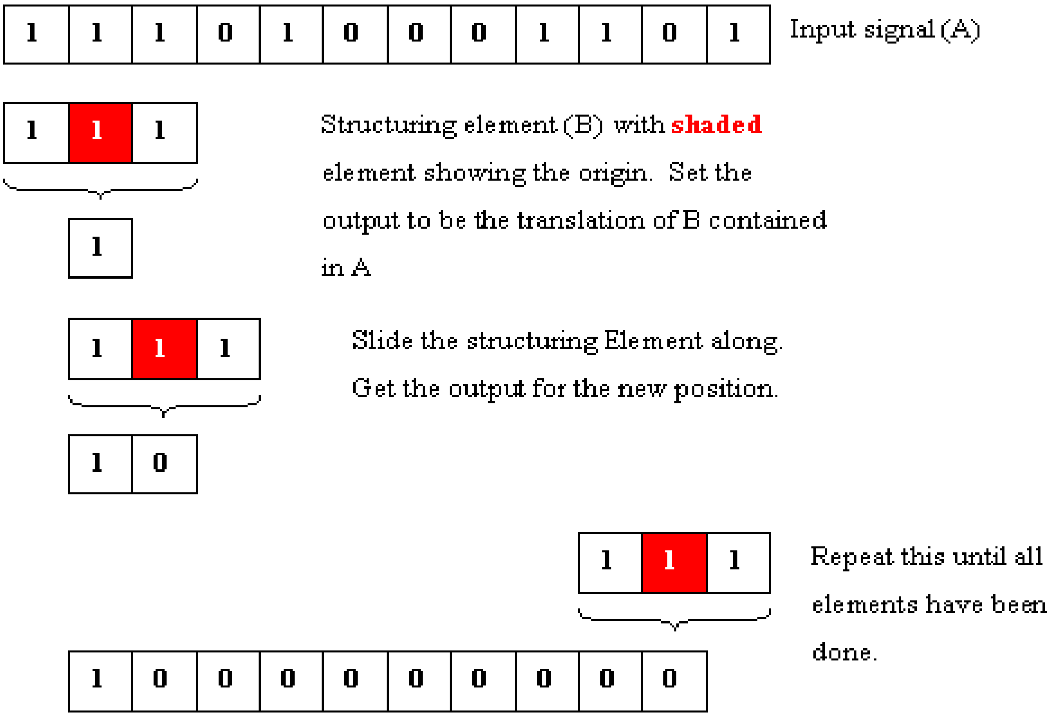


Structuring element (B) with **shaded** showing the origin. Set the output to be the intersection

Slide the structuring Element along. Get the intersection for the new position

Repeat this until all elements have been done.

Fig. 2.4 Example of dilation.



Structuring element (B) with **shaded** element showing the origin. Set the output to be the translation of B contained in A

Slide the structuring Element along. Get the output for the new position.

Repeat this until all elements have been done.

Fig. 2.5 Example of erosion.

2.5 Color Detection

Color is an important source of information during the human visual perception activities. There are some popular research topic like detecting and tracking human faces and gestures. Different color detection has applied to a variety of tasks, we can chose the color we want and using filter in web image contents, for examples about skin color like detecting and tracking human faces and gestures, and diagnosing disease [11],[12],[13].

As the first task in detection of moving object in special color and character extraction technique in our schemes, color detection can highly reduce the computational cost [14], and then extracts the potential object regions and character. Furthermore, color image segmentation is computationally fast while being relatively robust to changes in scale, viewpoint, and complex background.

According to the characteristics of module in color space distribution, the color of pixel can be detected quickly by a module's color model. However the use of different color spaces for different races and different illuminations often results in different detection accuracy [15]. In this thesis, the experimental environment is our laboratory and the lighting condition is fixed.

Usually, the color detection should be considered two aspects: color space selection and how to use the color distribution to establish a good color model. Nowadays main color spaces include RGB, HSV, HSI, YC_bC_r, some of their variant, etc, while RGB is the foundational method to represent color.

2.6 Character recognition

In character recognition applications, it can be divided into two categories, Optical Character Recognition (OCR) and On Line Character Recognition (OLCR). The OLCR uses a handwritten board or digital pen as an input tool to get the characters and then implement the character recognition. Different from the OLCR, The OCR uses a scanner to scan a document and save it as an image file, and then identifies the characters in the image file. This thesis will adopt the OCR for character recognition since the characters to be recognized are attained from a sequence of images.

The basic flow chart of the character recognition is shown in Fig.2.6. In general, the pre-processing of an image contains the object location, size normalization, binarization, angle of rotation, tilt, etc. There are two main parts in the character recognition system, which are feature extraction and feature classification. They are related to the speed and accuracy in text recognition. In these steps, many methods have been proposed and they can be divided into three types: Statistics method, Structure method, Merger Statistics and Structure method.

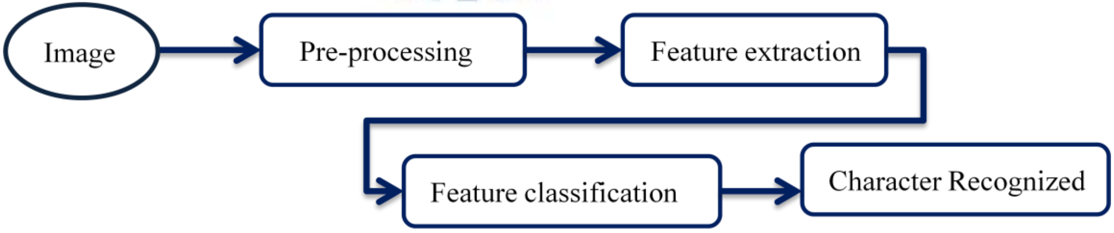


Fig. 2.6 Flow chart of character recognition

The main part in statistical method is to measure the composition of some particular physical quantities in the image. It's usually extracted text features or characteristics, to classify and by matching the pattern in built-in database. Basically, it's easy to make such character, have fast calculation, and often organized into a vector. Therefore, the feature space of the text image can be mapped to the point, when the point is closest to the one-word characteristic distribution, this image is judged as a text.

Structure method is using the geometric structure of the text, and setting description language to represent text which usually based on the structure of the characters. Word is split into several parts and compare with built-in database in order to determine the most similar result. In general, the structure method can tolerate its own variability. But the reaction with interference of noise is unstable. For example: template matching method.

Merger Statistics and Structure method combined the advantages in two ways. This thesis used merger statistics and structure method and mainly refers to this literature on the feature extraction, Torres-Mendez, L.A. "Translation, Rotation, and Scale-Invariant Object Recognition" [16].The paper presents a method for object recognition that achieves excellent invariance under translation, rotation, and scaling. In the feature extraction, it takes into account the invariant properties of the normalized moment of inertia [17] and a novel coding that extracts topological object characteristics [18].

The feature extraction is based on a set of concentric circles which are naturally and perfectly invariant to rotation (in 2-D). Fig. 2.7(a) shows an example with 8 concentric circles. Each circle is cut into some arcs by the character. Heuristically, the number of arcs of the *i*-th outside the character can be used as the first feature, denoted as M_i. This simple coding scheme extracts the topological characteristics of the object regardless of its position, orientation, and size. However, in some cases, two different objects could have the same or very similar M_i value (for example, letters 2 and 5). For the second feature, we take into account the difference of the two largest arcs for each circle outside the object and normalize the difference by the circumference, denoted as

$$D_i = \frac{d_{i2} - d_{i1}}{2\pi r_i} \tag{2.15}$$

for the *i*-th circle. Fig. 2.7(b) shows d_{31} and d_{32} of the third circle as an example.

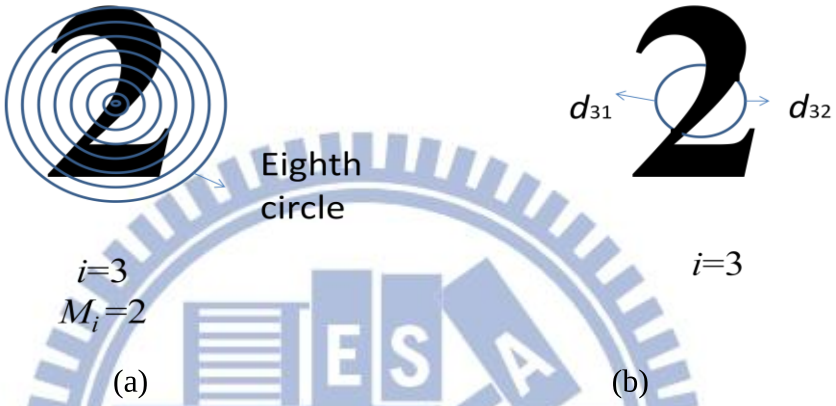


Fig. 2.7 Example of feature extraction.

(a) Example with 8 concentric circles. (b) d_{31} and d_{32} of the third circle.

Chapter 3 Intelligent Word Card Image Recognition System

The intelligent recognition system is implemented in several steps as shown in Fig. 3.1, including potential object localization, character extraction and character recognition. Each step adopts some schemes of image processing, such as image subtraction, morphology operation and connected components labeling (CCL) are used in the first step to extract moving objects. Different to the conventional image processing, this thesis will adopt the intelligent neural network on the basis of supervised learning to accomplish part of the schemes, detection of moving object in special color, color extraction and object recognition.

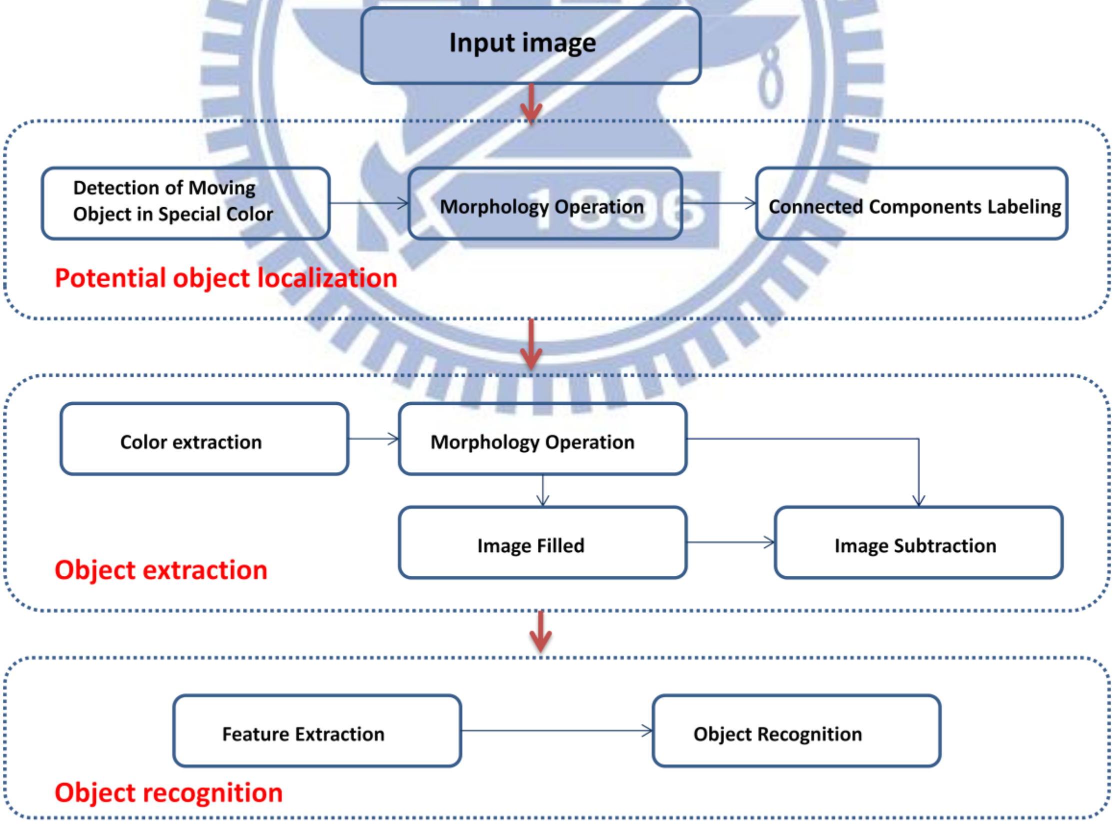


Fig. 3.1 The flow chart of the intelligent system.

3.1 Detection of Moving Object in special color

This is the first part of potential object location. In usual, there are two fundamental steps to detect a moving object of special color, which include moving object detection and color extraction. Both steps are often processed separately, but this thesis presents a scheme based on the artificial neural network to extract the color and detect the moving object simultaneously.

To detect a moving object from a sequence of images, the algorithm is shown as below:

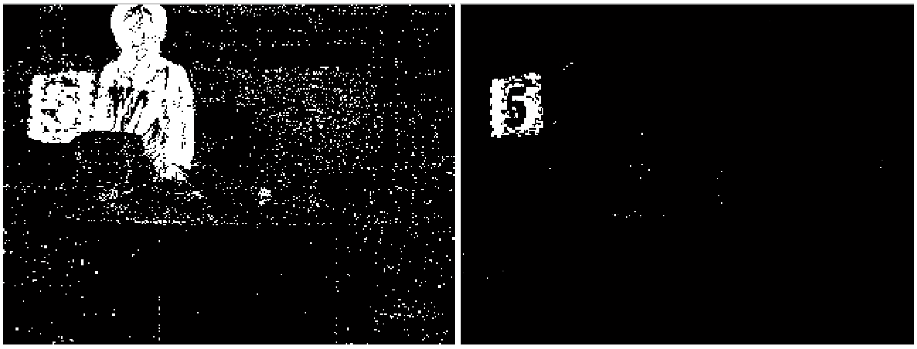
$$I_m(x, y) = \begin{cases} 1 & \left|I_t(x, y) - I_{t-1}(x, y)\right| \geq I_{th} \\ 0 & \left|I_t(x, y) - I_{t-1}(x, y)\right| < I_{th} \end{cases} \tag{3.1}$$

where $I_t(x,y)$ and $I_{t-1}(x,y)$ represent the images at the time t and $t-1$ and I_{th} is the threshold. It is clear that $I_m(x,y)$ is a binary image. To detect a special color in the following ranges:

$$I_{mosc}(x, y) = I_c(x, y) \mid B_t(\tag{3.2}$$

where I_c represent the color in the image and we choose green for example, I_{mosc} is the result of moving object in special color is shown in Fig. 3.2(d),

(a) (b)



(c) (d)

Fig. 3.2 (a) Input image (T=t-1). (b) Input image (T=t).

(c) Moving detection (I_m). (d) Detection of moving object in special color (I_{mosc}).

In supervised learning, the training data I_{mosc} are required as shown in Fig. 3.2(d).The RGB information is learned by the neural network structure in Fig. 3.3 based on the back-propagation. After learning, moving object of special color can be distinguished from the background according to the output value of neural network. Usually, a pixel of moving object of special color has an output value near to 1, while a pixel in the background has an output value near to 0. To efficiently extract the moving object of special color in an image, a threshold value should be carefully selected under the lighting condition of the environment being properly controlled.

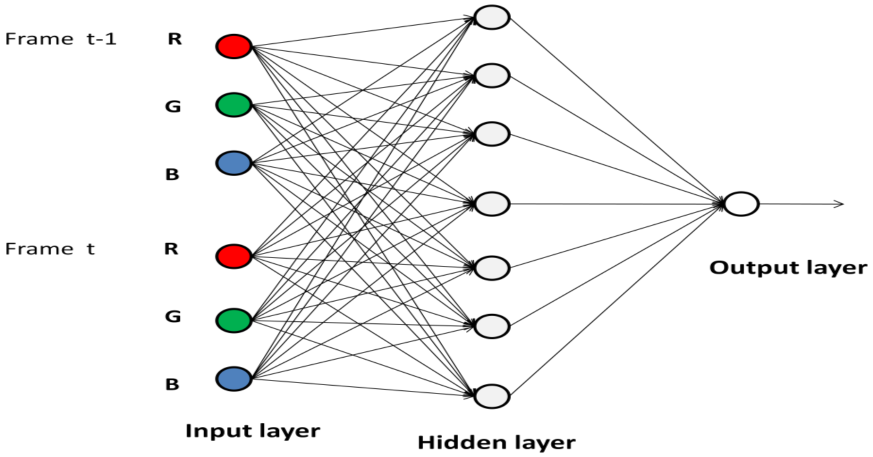


Fig. 3.3 MCNN's structure.

The neural network MCNN extracts a moving object of special color is shown in Fig. 3.3, which is composed of one input layer with 6 neurons, one hidden layer with 7 neurons, and one output layer with 1 neuron. The RGB values are sent into the 6 neurons of the input layer, represented by *MC(p)*, where *p*=1,2,3 for frame *t*-1 and *p*=4,5,6 for frame *t*. The *p*-th input neuron is connected to the *q*-th neuron, *q*=1,2,…,7, of the hidden layer with weighting $W_{MC1}(p,q)$, which is a weighting array of dimension 6×7. Besides, the *q*-th neuron of the hidden layer is also with an extra bias $b_{MC1}(q)$. Finally, the *q*-th neuron of the hidden layer is connected to the output neuron with weighting $W_{MC2}(q)$, *q*=1,2,…,7, and a bias b_{MC2} is added to the output neuron.

Let the activation function of the hidden layer be the hyperbolic tangent sigmoid transfer function, then the *q*-th output neuron $O_{MC1}(q)$ is expressed as:

$$O_{MC1}\left(q\right) = tansig(n_1(q)) = \frac{2}{1+exp\left(-2n_1\left(q\right)\right)} - 1, \quad q = 1, 2, ..., 7. \quad (3.3)$$

where

$$n_1\left(q\right) = \sum_{p=1}^{6} W_{MC1}(p,q) MC(p) + b_{MC1}(q) \quad (3.4)$$

Let the activation function of the output layer be the log-sigmoid transfer function, then the single output neuron O_{MC2} is expressed as:

$$O_{MC2} = logsig(n_2) = \frac{1}{1+exp\left(-n_2\right)} \quad (3.5)$$

where

$$n_2 = \sum_{q=1}^{7} W_{MC2}(q) O_{MC1}(q) + b_{MC2} \quad (3.6)$$

The above operations are shown in Fig. 3.4.

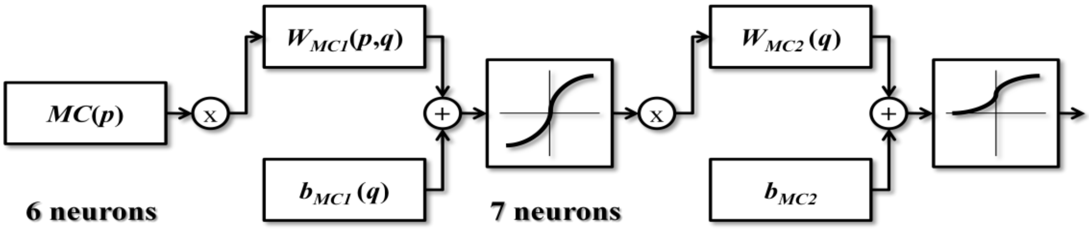


Fig. 3.4.MCNN

3.2 Morphology operation

Three parts are using in this section, erosion; dilation and holes filling.

3.2.1 Erosion and Dilation

After applying color extraction, color regions are extracted from the original image, but some noise still exists therein. One of the conventional ways to eliminate noise regions is using the morphology operations. In the thesis, the noises are eliminated by the morphology erosion operation (2.14) expressed as

$$A \Theta B = \{x : (B)_x \subseteq A\} \qquad (3.7)$$

where *B* is a disk-shaped structuring element with radius 4 as shown in Fig. 3.5(a) and the noises in image *A* with region smaller than *B* are erased after operation. However, some gaps may be also generated in isolated regions after erosion. In order to repair these gaps, further employ the morphology dilation operation (2.13) expressed as

$$A \oplus C = \{x : (\hat{C})_x \cap A \neq \phi\} \qquad (3.8)$$

where *C* is a disk-shaped structuring element with radius 6 as shown in Fig. 3.5(b) and the gaps in image *A* are repaired after operation. Fig. 3.6 shows an example of erosion and dilation using the structuring elements *B* and *C*.

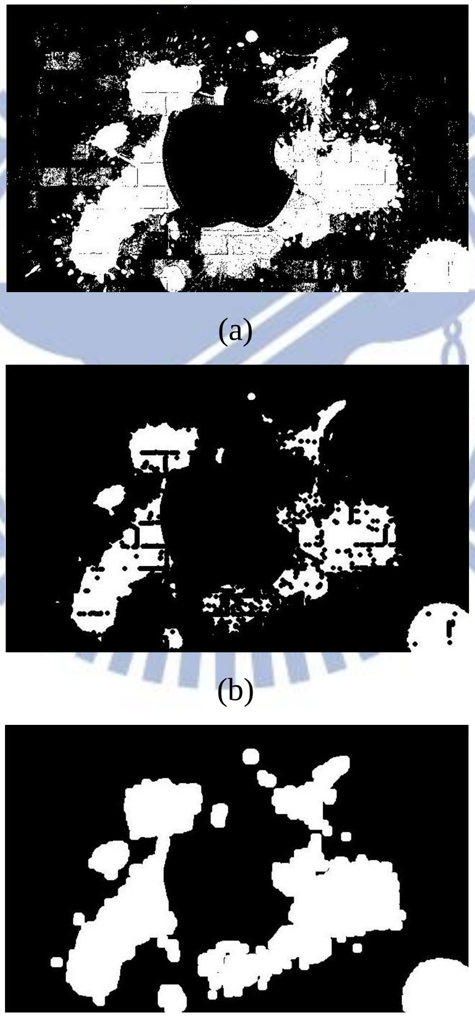
0	0	1	1	1	1	0	0
0	1	1	1	1	1	1	0
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
0	1	1	1	1	1	1	0
0	0	1	1	1	1	0	0

(a)

0	0	1	1	1	1	1	1	1	1	0	0
0	1	1	1	1	1	1	1	1	1	1	0
1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1
0	1	1	1	1	1	1	1	1	1	1	0
0	0	1	1	1	1	1	1	1	1	0	0

(b)

Fig. 3.3 (a) Structuring element B (b) Structuring element C



(a)

(b)

(c)

Fig. 3.4 Steps for morphology operations (a) Initial image (b) Result of erosion using structuring element B (c) Result of dilation using structuring element C

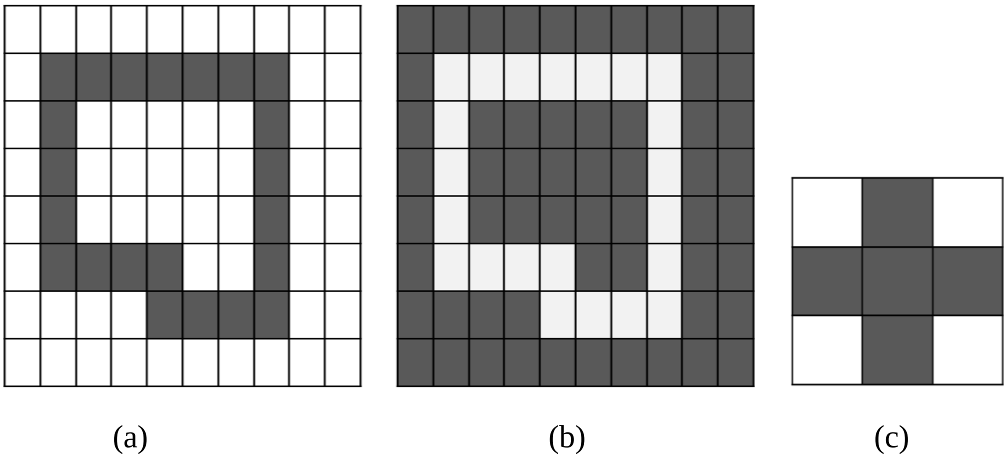
3.2.2 Holes filling

In the current application, it is appropriately called conditional dilation or inside connected components. A hole may be defined as a background region surrounded by connected border of foreground pixels. In this section, we develop an algorithm based on set dilation, complementation, and intersection for filling holes in an image. Let A denote a set whose elements are 8-connected boundaries as in Fig. 3.7(a), each boundary enclosing a background region. Given a point in each hole, the objective is to fill all the holes with 1.

We begin by forming an array, X_0, of 0s, expect at the locations in $X_{\mathbf{0}}$ corresponding to given point in each hole, which we set to 1. Then, the following procedure fills all holes with 1s:

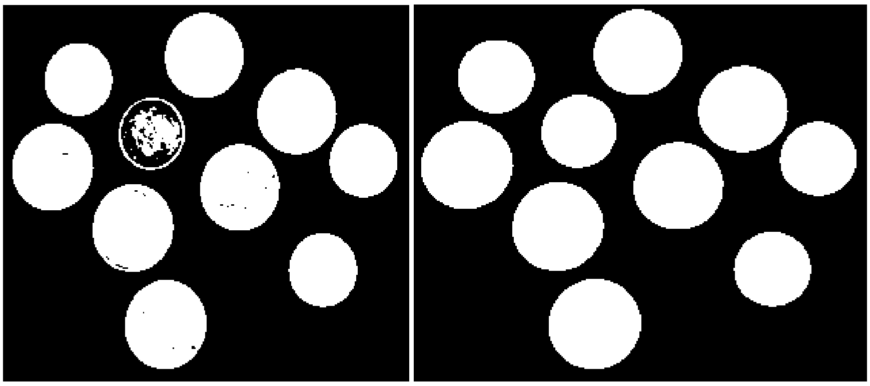
$$X_k = (X_{k-1} \oplus B) \cap A^c \qquad k = 1,\ 2,\ 3 \qquad (3.9)$$

where *B* is the symmetric structuring element in Fig. 3.7(c). The algorithm terminates at iteration step k if X_k=X_{k-1}.The set then X_k contains all the filled holes. The set union of X_k and *A* contains all the filled holes and their boundaries. Fig. 3.8(a) shows an example image, and Fig. 3.8(b) shows the result of the hole filling.



(a) (b) (c)

Fig. 3.7(a) A (b) A^c (c) B



(a) (b)

Fig.3.8

Fig.3.8 (a) Original frame. (b) Result of holes filling.

3.3 Connected Components Labeling

After morphology operation different components are identified by using Connected Components Labeling (CCL), which is often used in computer vision to detect connected regions containing 4 or 8 pixels in the binary digital image. In this thesis, the 4-pixel connected component will be used to label potential face regions.

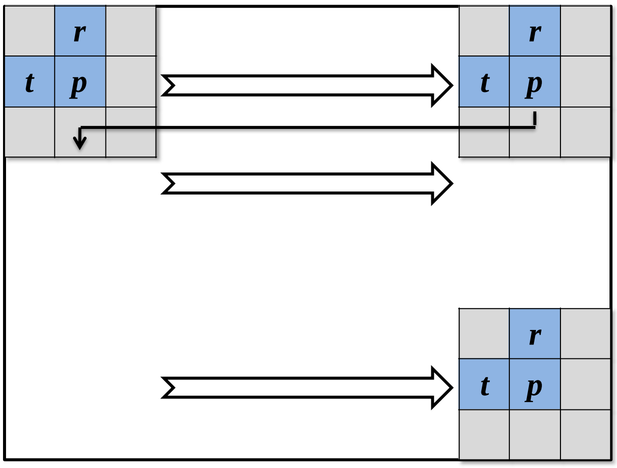


Fig. 3.9 Scanning the image.

The 4-pixel connected CCL algorithm can be partitioned into two processes, labeling and componentizing. During the labeling, the image is scanned pixel by pixel, from left to right and top to bottom as shown in Fig. 3.9, where p is the pixel being processed, and r and t are respectively the upper and left pixels to p. Defined $v(\cdot)$ and $l(\cdot)$ as the binary value and the label of a pixel. If $v(p)$=0, then move on to next pixel, otherwise, i.e., $v(p)$=1, the label $l(p)$ is determined by following rules:

R1. For $v(r)$=0 and $v(t)$=0, assign a new label to $l(p)$.

R2. For $v(r)$=1 and $v(t)$=0, assign $l(r)$ to $l(p)$, i.e., $l(p)=l(r)$.

R3. For $v(r)$=0 and $v(t)$=1, assign $l(t)$ to $l(p)$, i.e., $l(p)=l(t)$.

R4. For $v(r)$=1, $v(t)$=1 and $l(t)=l(r)$, then assign $l(r)$ to $l(p)$, i.e., $l(p)=l(r)$.

R5. For $v(r)$=1, $v(t)$=1 and $l(t) \neq l(r)$, then assign $l(r)$ to both $l(p)$ and $l(t)$, i.e., $l(p)=l(r)$ and $l(t)= l(r)$.

For example, after the labeling process, Fig. 3.10(a) is changed into Fig. 3.10(b). It is clear that some connected components contain pixels with different labels. Hence, it is required to further execute the process of componentizing, which sorts all the pixels connected in one component and assign them by the same label, the smallest number among the labels in that component. Fig. 3.10(c) is the result of Fig. 3.10(b) after componentizing.

0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	1	0	1	0	0
0	0	0	0	1	1	0	1	0	0
0	0	0	1	1	0	0	0	0	0
0	0	0	1	1	0	0	0	1	0
0	1	1	1	0	0	1	0	1	0
1	1	1	1	0	0	1	1	1	0
0	0	0	0	0	0	0	0	0	0

(a)

0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	1	0	2	0	0
0	0	0	0	1	1	0	2	0	0
0	0	0	1	1	0	0	0	0	0
0	0	0	1	1	0	0	0	3	0
0	4	1	1	0	0	5	0	3	0
4	1	1	1	0	0	5	3	3	0
0	0	0	0	0	0	0	0	0	0

(b)

0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	1	0	2	0	0
0	0	0	0	1	1	0	2	0	0
0	0	0	1	1	0	0	0	0	0
0	0	0	1	1	0	0	0	3	0
0	1	1	1	0	0	3	0	3	0
1	1	1	1	0	0	3	3	3	0
0	0	0	0	0	0	0	0	0	0

(c)

Fig. 3.10 Example of 4-pixel connected CCL.

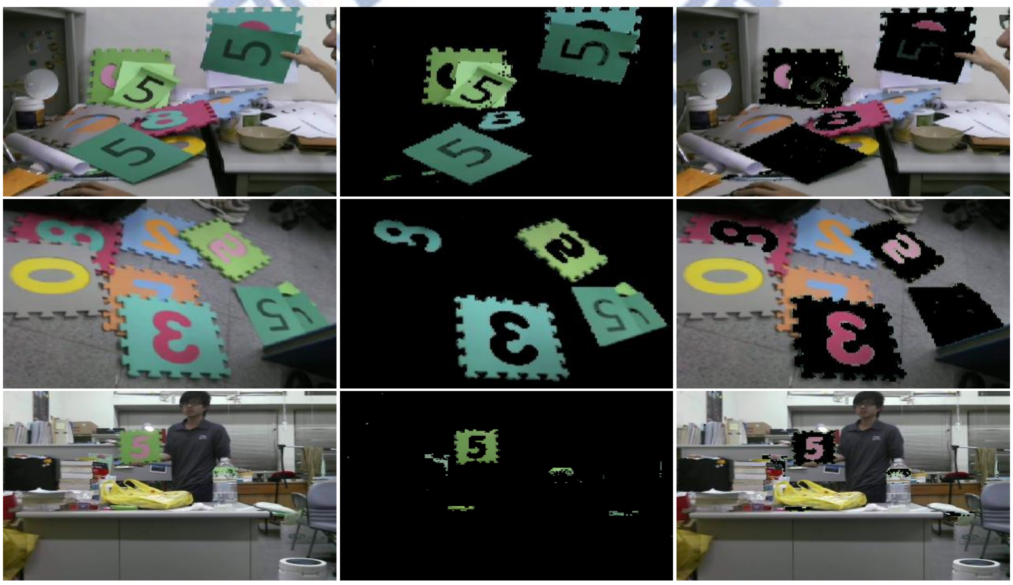
(a) Digital image. (b) Labeling. (c) Componentizing.

3.4 Character extraction from Potential Region

Two main parts in this section, color extraction and character extraction.

3.4.1 Color extraction

This system presents the color extraction based on artificial neural network. In supervised learning, the training data of the colors are required and obtained from images composed of object and background. Examples of training data are shown in Fig. 3.11, where each original image is separated into object region and background. The RGB information is learned by the neural network structure in Fig. 3.12 based on the back-propagation. After learning, the pixels of green color can be distinguished from the background according to the output value of neural network. Usually, a pixel of color has an output value near to 1, while a pixel in the background has an output value near to 0. To efficiently extract the color in an image, a threshold value should be carefully selected under the lighting condition.



(a) (b) (c)

Fig. 3.11 Examples of training data for skin color extraction.

(a) Original image (b) Skin color region (c) Background

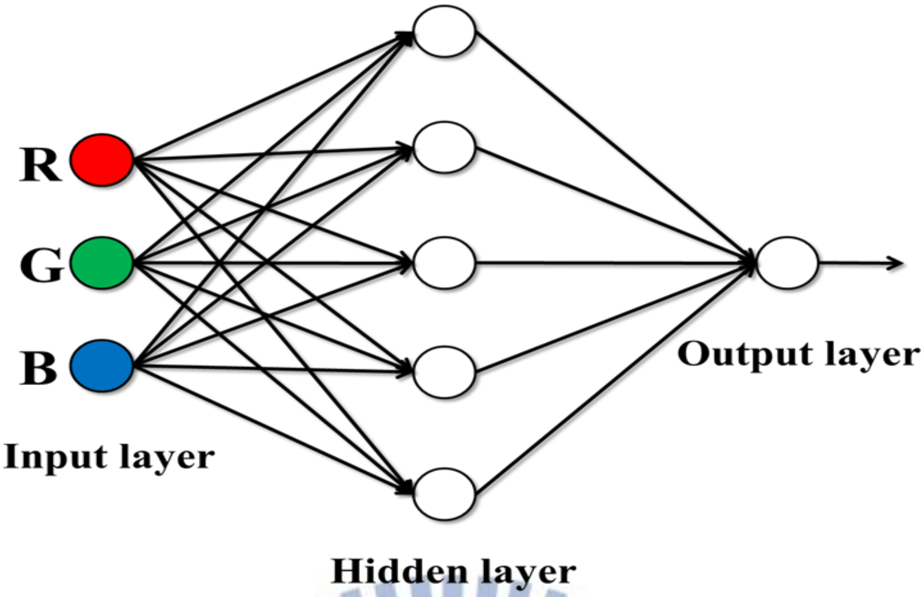


Fig. 3.12 Neural network structure for color extraction.

For green color extraction neural network (GNN) based on the structure in Fig. 3.12, there include one input layer with 3 neurons, one hidden layer with 5 neurons, and one output layer with 1 neuron. The RGB values are sent into the 3 neurons of the input layer, represented by $G(p)$, p=1,2,3, correspondingly. The p-th input neuron is connected to the q-th neuron, q=1,2,…,5, of the hidden layer with weighting $W_{G1}(p,q)$. Hence, there exists a weighting array $W_{G1}(p,q)$ of dimension 3x5. Besides, the q-th neuron of the hidden layer is also with an extra bias $b_{G1}(q)$. Finally, the q-th neuron of the hidden layer is connected to the output neuron with weighting $W_{G2}(q)$, q=1,2,…,5, and a bias b_{G2} is added to the output neuron.

Let the activation function of the hidden layer be the hyperbolic tangent sigmoid transfer function and the q-th output neuron $O_{G1}(q)$ is expressed as:

$$O_{G1}(q) = tansig(n_1(q)) = \frac{2}{1+exp(-2n_1(q))} - 1, \quad q = 1,2,...,5. \qquad (3.10)$$

where

$$n_1(q) = \sum_{p=1}^{3} W_{G1}(p,q)G(p) + b_{G1}(q) \tag{3.11}$$

Let the activation function of the output layer be the log-sigmoid transfer function and the single output neuron O_{G2} is expressed as:

$$O_{G2} = logsig(n_2) = \frac{1}{1+exp(-n_2)} \tag{3.12}$$

where

$$n_2 = \sum_{q=1}^{5} W_{G2}(q)O_{G1}(q) + b_{G2} \tag{3.13}$$

The above operations are shown in Fig. 3.13.

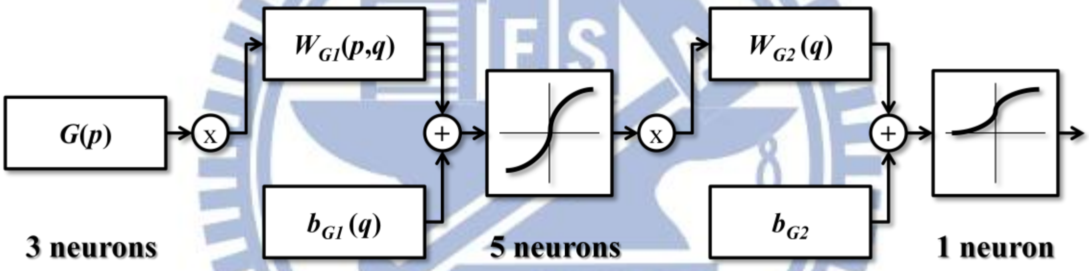


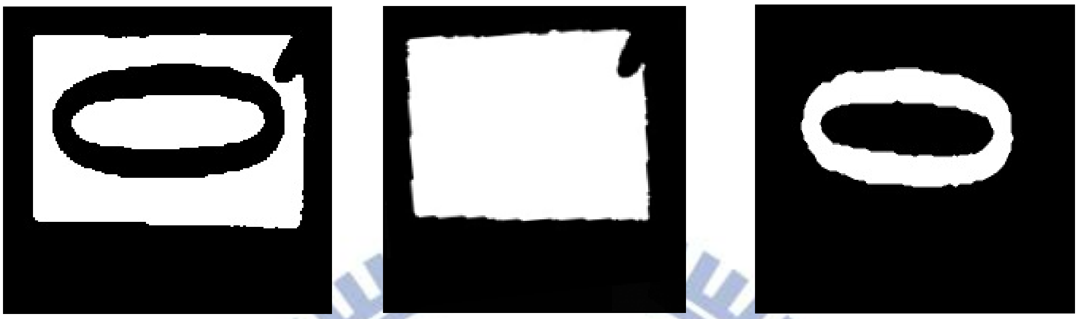
Fig. 3.13 GNN.

3.4.2 Character extraction

After detect the green color of the word card, the word card will be abstract. Here are three steps in this chapter to extract character from the word card. First, some noise still exists therein. Using erosion (3.7) structuring element with radius 2 and dilation (3.8) structuring element with radius 3 will reduce noises and get the frame as shown in Fig. 3.14(a).

The word card is thus achieved as a binary image with the shape of the character on it as shown in Fig. 3.14(a).The character will be extracted by using holes filling and image subtraction.

In the first, generate a plain binary card with holes filling, the holes of the word card will be filled as shown in Fig. 3.14(b). Then, by generating a plain binary card, the character on the word card can be extracted by subtracting the plain binary card. The result is shown in Fig. 3.14(c).



(a) (b) (c)

Fig. 3.14 (a) Result after reduce the noise.
(b)Image with hole filled. (c) Image subtraction.

3.5 Character recognition

Two parts are using in this section, Feature extraction and classification.

3.5.1 Feature extraction

After object extraction, the character recognition is executed as the following step, whose performance will directly affect the overall accuracy rate. There are two main parts of the character recognition, one is feature extraction and the other is classification. This section will focus on the feature extraction [16]. In this thesis, we create a system which can deal with not only translation and rotation problem but also geometric deformation, like affine deformation.

The features to be extracted are set to be I_1 to I_{12} which will be used for character recognition and defined in the followings. The first feature is the normalization of the moment of inertia of the image, defined as [17]

$$I_1 = \frac{1}{N^2}\sum_{i=1}^{N}\left((x_i - C_x)^2 + (y_i - C_y)^2\right) \tag{3.14}$$

where C_x and C_y are the location of centroid, xi; yi are the image pixel coordinates of the object, and N is the total number of pixels in the object. Clearly, this feature is scaling invariant.

The second feature I_2 is similar with I_1, another feature possessing scaling invariance, and defined as

$$I_2 = \left(\frac{N}{L_{\min} * L_{maj}}\right)^2 * 10 \tag{3.15}$$

where L_{maj} is the major axis length and $L_{\min}$ is the minor axis length, which are the length if the smallest ellipse as shown in Fig.3.15. Using equation (3.15) will let the value of the percentage of pixel in the area separated clearly.

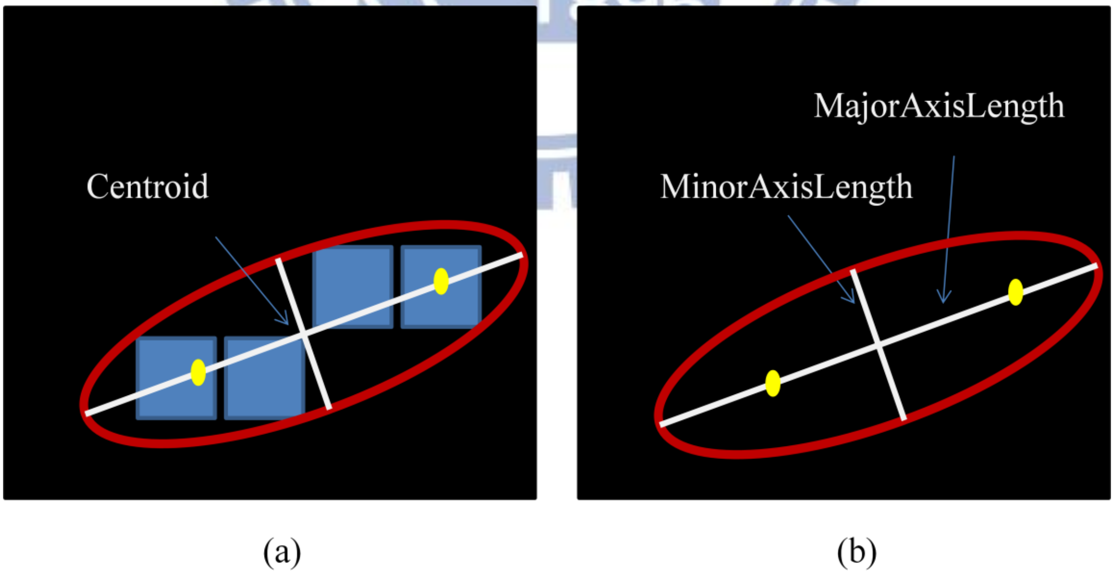


Fig. 3.15(a)(b) Example of major axis length and minor axis length.

The third feature I_3 is called the Euler number, which is a measure of the topology of an image and defined as the total number of objects in the image minus the number of holes in those objects. For example, 0's Euler number is 0 and 8's Euler number is −1.

For the features I_4 and I_5, which are defined as the normalized difference of arcs outside the character of the fourth and the fifth circles. From (2.15), both can be expressed as

$$I_4 = \frac{d_{42} - d_{41}}{2\pi r_4} \qquad (3.17)$$

$$I_5 = \frac{d_{52} - d_{51}}{2\pi r_5} \qquad (3.18)$$

For the rest of features from I_6 to I_{12}, they are defined as

$$I_k = M_{k-5}, \qquad k=6,7,...,12 \qquad (3.19)$$

which adopts the definition Mi to represent the number of arcs of i-th circle outside the character shown in section 2.6. Clearly, these features are related to 1st to 7th circles. As for the 8th circle, it is neglected since the 8th circle in all the characters are almost the same.

3.5.2 Classification with Neural Networks

In this thesis, there are 43 neurons used for the ANN structure shown in Fig. 3.16, called the Character-recognition neural network or CRNN in short, which contains 12 neurons from m_1 to m_{12} for input layer, 17 neurons for hidden layer, and 4 neurons from e_1 to e_4 for output layer. The inputs of the neurons CR_i, i=1, 2,…, 12, are the feature extract in chapter 3.5.1 The outputs of the neurons e_i, i=1, 2,…, 4, are the results corresponding each character. The learning process of CRNN are shown in Fig. 3.6(b), from left to right and downwards for the entire image.

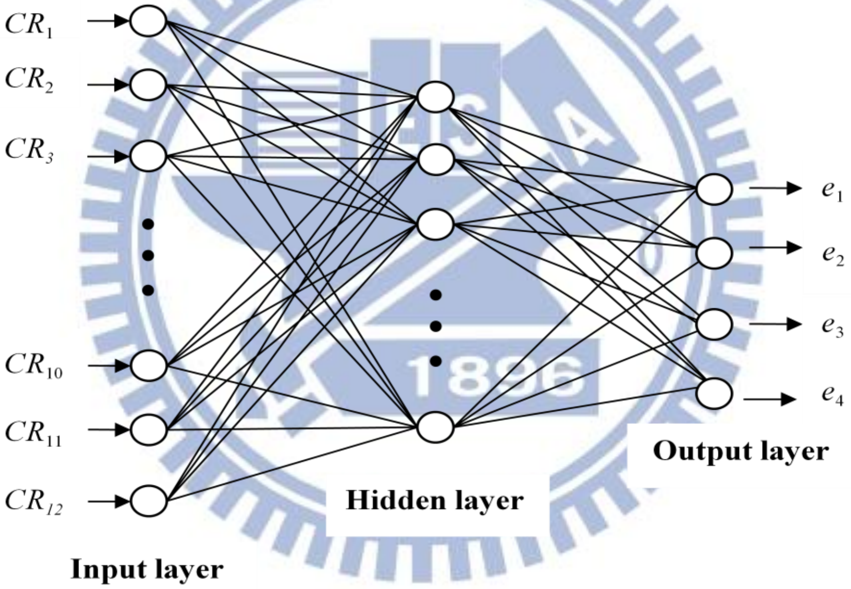


Fig. 3.16 CRNN structure.

For ENN based on the structure in Fig. 3.16, the gray level values, represented by $CR(p)$, p=1,2,…,12, are sent into the 12 neurons of the input layer, correspondingly. The p-th input neuron is connected to the q-th neuron of the hidden layer with weighting $W_{CR1}(p,q)$. Hence, there exists a weighting array $W_{CR1}(p,q)$ of dimension 12x17, p=1,2,…,12 and q=1,2,…,17. Besides, the q-th neuron of the hidden layer is also with an extra bias $b_{CR1}(q)$, q=1,2,…,17. Finally, the q-th neuron of the hidden layer is connected to the output neuron with weighting $W_{CR2}(q,r)$, q=1,2,…,17 and r=1,2,…,4. The r-th neuron of the output layer is also with an extra bias $b_{CR2}(r)$, r=1,2,…,4.

Choose the output of the q-th neuron in the hidden layer as the following hyperbolic tangent sigmoid transfer function

$$O_{CR1}\left(q\right)=tansig(n_1\left(q\right))=\frac{2}{1+exp\left(-2n_1\left(q\right)\right)}-1, \quad q=1,2,...,17. \qquad (3.19)$$

where $n_1(q)$ is the input of the q-th neuron obtained as:

$$n_1\left(q\right)=\sum_{p=1}^{12}W_{CR1}(p,q)CR(p)+b_{CR1}(q) \qquad (3.20)$$

For the output layer, its r-th neuron is selected to be the following log-sigmoid transfer function

$$O_{CR2}\left(r\right)=logsig(n_2\left(r\right))=\frac{1}{1+exp\left(-n_2\left(r\right)\right)}, \quad r=1,2,...,4. \qquad (3.21)$$

where the input $n_2(r)$ of the r-th neuron is attained as:

$$n_2\left(r\right)=\sum_{q=1}^{17}W_{CR2}(q,r)O_{CR1}(q)+b_{CR2}(r) \qquad (3.22)$$

The above operations are shown in Fig. 3.17.

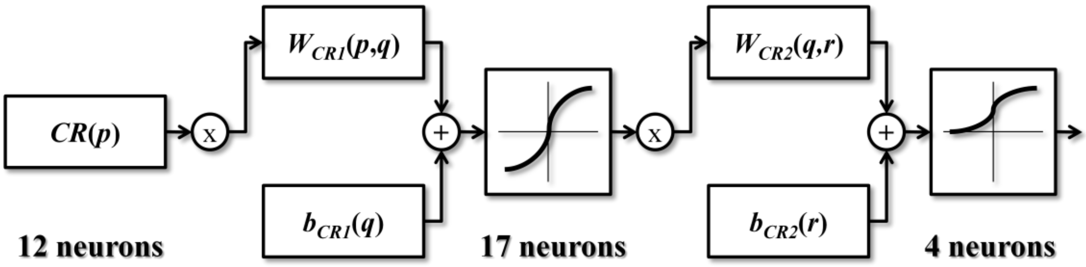


Fig. 3.17 CRNN.

Chapter 4 Experiments

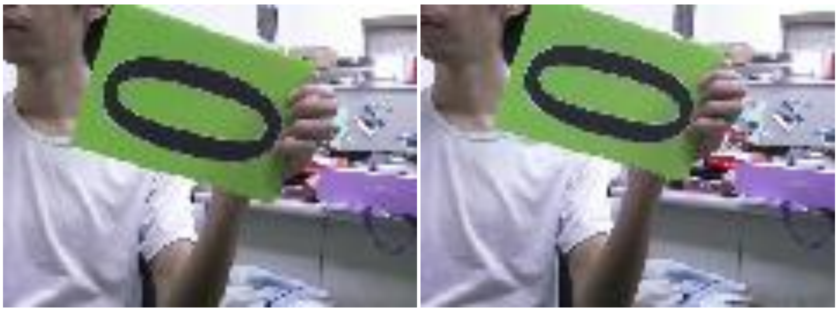
In this chapter, the experiment results of the system are present in the first part and the environment is set in our lab. In the chapter, some additional finding will be described in the second part and the execution time will be discussed at the third part. The proposed algorithm will be obtained by MATLAB R2010b.

4.1 Part I: Result of Each Steps

In the previous chapters, three main steps of the proposed system are introduced. In this part, the experiment results of each step will be expressed and character recognition results will be separated into feature extraction and classification.

4.1.1 Result of Potential Object Localization

For this system, there are three steps in potential object localization. Detection of moving object in special color, morphology operation and connected components labeling are used in to classify the potential region. First, Detection of moving object in special color extract target pixels from two input color image with time difference, which with the size 320x240 pixels. Two input color images are shown in Fig. 4.1(a) and Fig. 4.1 (b), the result image is shown in Fig. 4.2. After detection of moving object in special color extraction, there are still many noise in the scene, to reduce the noise, morphology operation is applied to eliminate small area and to clear the connect region as shown in Fig. 4.3(a),(b). To find out the potential regions, the CCL is applied as shown in Fig. 4.4 and more example for moving object extraction are shown in Fig. 4.5.



(a) (b)

Fig. 4.1(a)Input image (T=t-1). (b)Input image (T=t)

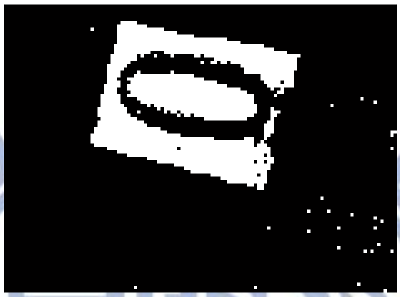
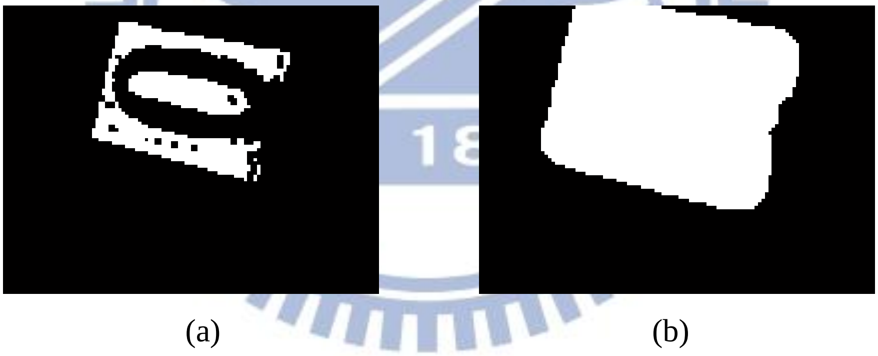


Fig. 4.2 Detection of moving object in special color.



(a) (b)

Fig. 4.3 (a) Erosion. (b) Dilation.

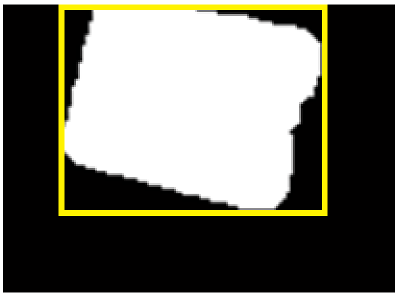


Fig. 4.4 CCL

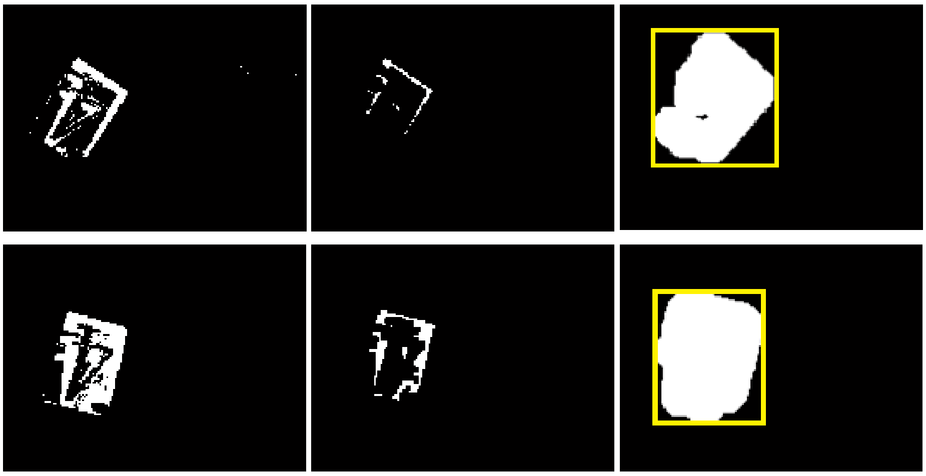


Fig. 4.5 More example for moving object extraction. (a) Detection of moving object in special color. (b) Result after morphology operation. (c) Result after CCL

4.1.2 Result of Character extraction

After find out the potential regions, the next step is to extract the regions from the original frame with the same potential regions, shown in Fig. 4.6(a). In character extraction, the result will show how to find the character in potential regions. The first step in character extraction used color as the feature to find the plate as shown in Fig. 4.6(b). After color extraction, noise reduce is shown in Fig. 4.6(c) and image filled is shown in Fig. 4.6(d).We subtract the plate with the image filled plate and using CCL classify the largest object to character. The result is shown in Fig. 4.6(e) and more examples for character extraction are shown in Fig. 4.7.

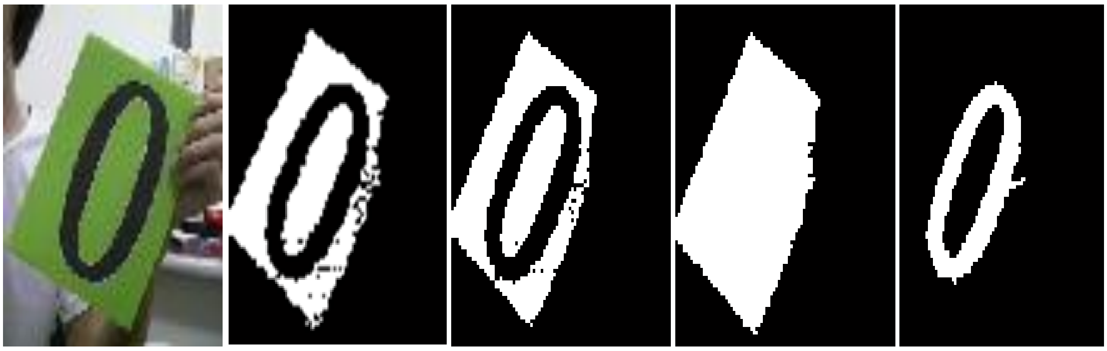


Fig. 4.6 (a) Potential region(b)color extraction.

(c)Reduce the noise. (d) Filled image. (e) Character extraction.

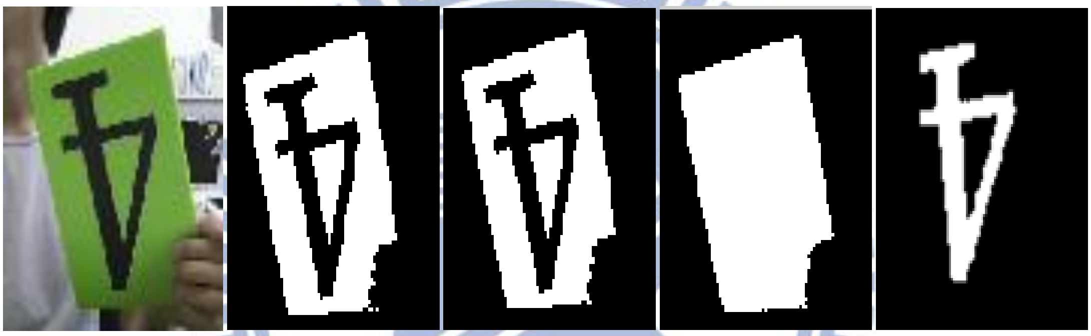


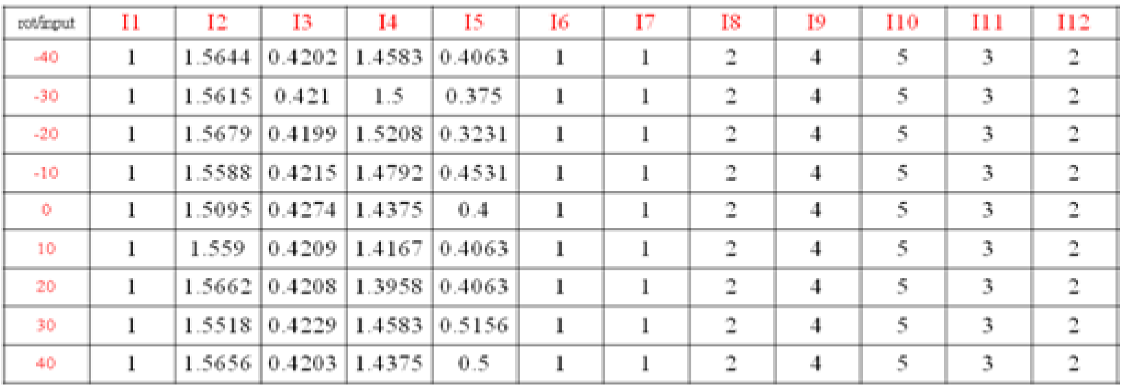
Fig. 4.7 More example for character extraction. (a) Potential region.

(b) Color extraction. (c)Reduce the noise. (d) Filled image. (e) Character extraction.

4.1.3 Result of Feature extraction

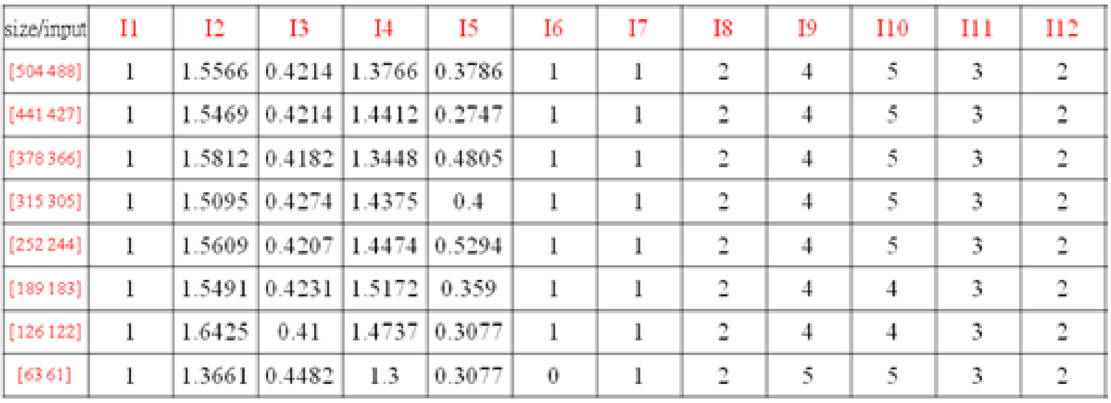
In this section, there are 4 cases which would be discussed for the feature extraction and the character “4” will used as an example. In case 1, consider the feature with 9 different angles from angle -40 to angle 40 with the same image size which is 315×300, the results in case 1 are shown in Table 4.1. In Case 2, consider the feature with different scale from 540×488 to 63×61 with the same aspect ratio, the results in case 2 are shown in Table 4.2. In Case 3, consider the feature with different aspect ratio from shorten the axis and the results in case 3 are shown in Table 4.3. As Table 4.1 shown, the parameters I_1~I_{12} are rotation-invariant. Table 4.2 shows, when the pattern is large enough, parameter’s values will be very similar. Therefore, the parameters can achieve invariance under translation, rotation, and scaling. As shown in Table 4.3, although the value of will greatly change with the different aspect ratio. With the different proportion as the value of I_2 and I_3, I_8~I_{12} may change as well . Therefore, when the character is different, even if different angle of rotation, different sizes or aspect ratio, the proposed system still can excellent to recognize which character it is.

Table. 4.1 Same image size with different angle.



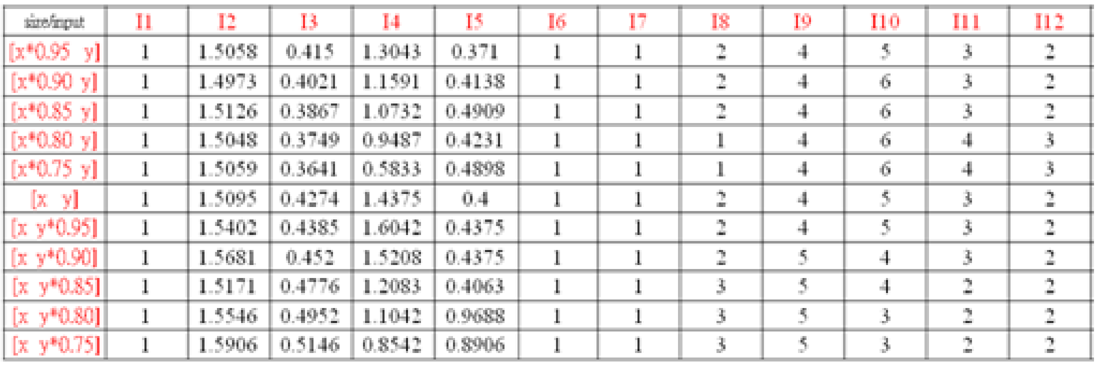
rot/input	I1	I2	I3	I4	I5	I6	I7	I8	I9	I10	I11	I12
-40	1	1.5644	0.4202	1.4583	0.4063	1	1	2	4	5	3	2
-30	1	1.5615	0.421	1.5	0.375	1	1	2	4	5	3	2
-20	1	1.5679	0.4199	1.5208	0.3231	1	1	2	4	5	3	2
-10	1	1.5588	0.4215	1.4792	0.4531	1	1	2	4	5	3	2
0	1	1.5095	0.4274	1.4375	0.4	1	1	2	4	5	3	2
10	1	1.559	0.4209	1.4167	0.4063	1	1	2	4	5	3	2
20	1	1.5662	0.4208	1.3958	0.4063	1	1	2	4	5	3	2
30	1	1.5518	0.4229	1.4583	0.5156	1	1	2	4	5	3	2
40	1	1.5656	0.4203	1.4375	0.5	1	1	2	4	5	3	2

Table. 4.2 Same aspect ratio with different scale.



size/input	I1	I2	I3	I4	I5	I6	I7	I8	I9	I10	I11	I12
[504 488]	1	1.5566	0.4214	1.3766	0.3786	1	1	2	4	5	3	2
[441 427]	1	1.5469	0.4214	1.4412	0.2747	1	1	2	4	5	3	2
[378 366]	1	1.5812	0.4182	1.3448	0.4805	1	1	2	4	5	3	2
[315 305]	1	1.5095	0.4274	1.4375	0.4	1	1	2	4	5	3	2
[252 244]	1	1.5609	0.4207	1.4474	0.5294	1	1	2	4	5	3	2
[189 183]	1	1.5491	0.4231	1.5172	0.359	1	1	2	4	4	3	2
[126 122]	1	1.6425	0.41	1.4737	0.3077	1	1	2	4	4	3	2
[63 61]	1	1.3661	0.4482	1.3	0.3077	0	1	2	5	5	3	2

Table. 4.3 Different aspect ratio with the original size is 315×300



size/input	I1	I2	I3	I4	I5	I6	I7	I8	I9	I10	I11	I12
[x*0.95 y]	1	1.5058	0.415	1.3043	0.371	1	1	2	4	5	3	2
[x*0.90 y]	1	1.4973	0.4021	1.1591	0.4138	1	1	2	4	6	3	2
[x*0.85 y]	1	1.5126	0.3867	1.0732	0.4909	1	1	2	4	6	3	2
[x*0.80 y]	1	1.5048	0.3749	0.9487	0.4231	1	1	1	4	6	4	3
[x*0.75 y]	1	1.5059	0.3641	0.5833	0.4898	1	1	1	4	6	4	3
[x y]	1	1.5095	0.4274	1.4375	0.4	1	1	2	4	5	3	2
[x y*0.95]	1	1.5402	0.4385	1.6042	0.4375	1	1	2	4	5	3	2
[x y*0.90]	1	1.5681	0.452	1.5208	0.4375	1	1	2	5	4	3	2
[x y*0.85]	1	1.5171	0.4776	1.2083	0.4063	1	1	3	5	4	2	2
[x y*0.80]	1	1.5546	0.4952	1.1042	0.9688	1	1	3	5	3	2	2
[x y*0.75]	1	1.5906	0.5146	0.8542	0.8906	1	1	3	5	3	2	2

4.1.4 Result of Classification

The experiment results as shown in Fig. 4.8, the binary pictures are the characters after extraction and the classification results are at the left side. As shown in pictures, even if different angle of rotation, different sizes, noises or aspect ratio, the proposed system can excellent to recognize which character it is.

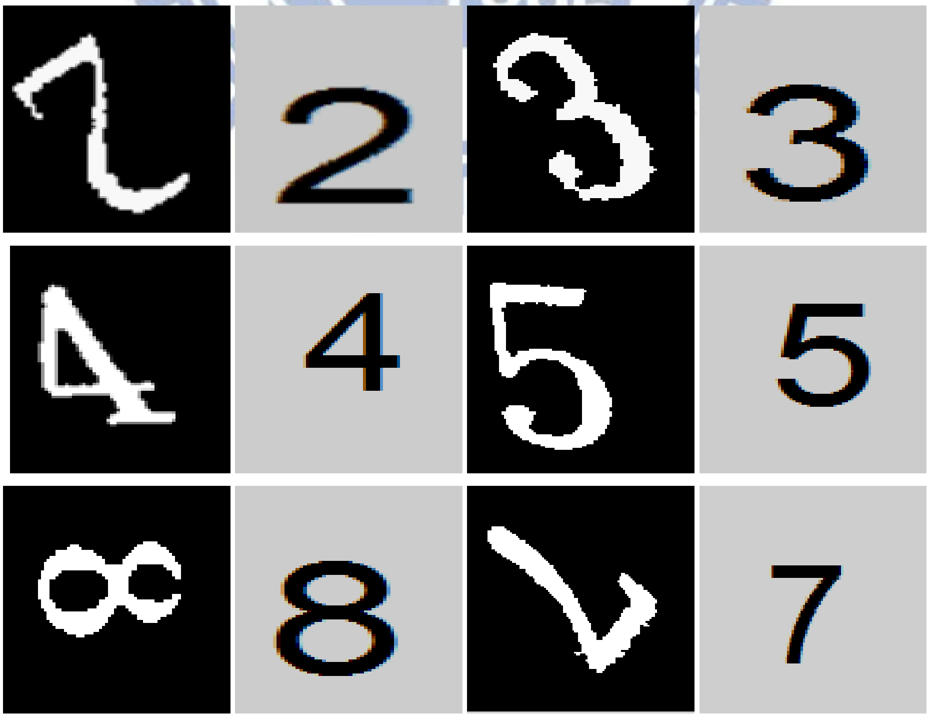


Fig. 4.8 Example for classification.

To compare the performance of the classification and then discusses 2 cases concerning the classification. In Case 1, consider the accuracy rates with 3 kinds of different training data. In Case 2, consider the accuracy rates with 2 kinds of different input neurons.

For training the Character-recognition neural network or CRNN in short, the thesis uses 137700 training data as shown in Fig. 4.9, with different size, rotation, translation, tilt and different aspect ratio and 6000 testing data. The accuracy rate is shown in Table 4.4. CRNN2 and CRNN3 are using the same structure with different training data. Clearly, the more training data will get the higher accuracy rate.

In Case 2, two kinds of CRNN are discussed, the CRNN4 is using the features without neglected the number of arcs of 8-th circle outside the character and CRNN5 is using the features with neglected the number of arcs of 7-th and 8-th circle outside the character. Clearly, although the error in CRNN4 is fewer, there are almost no different in accuracy rate between CRNN and CRNN4. Consider to neglected another feature, neglected the I_{12} can get the best accuracy rate is 99.0%.

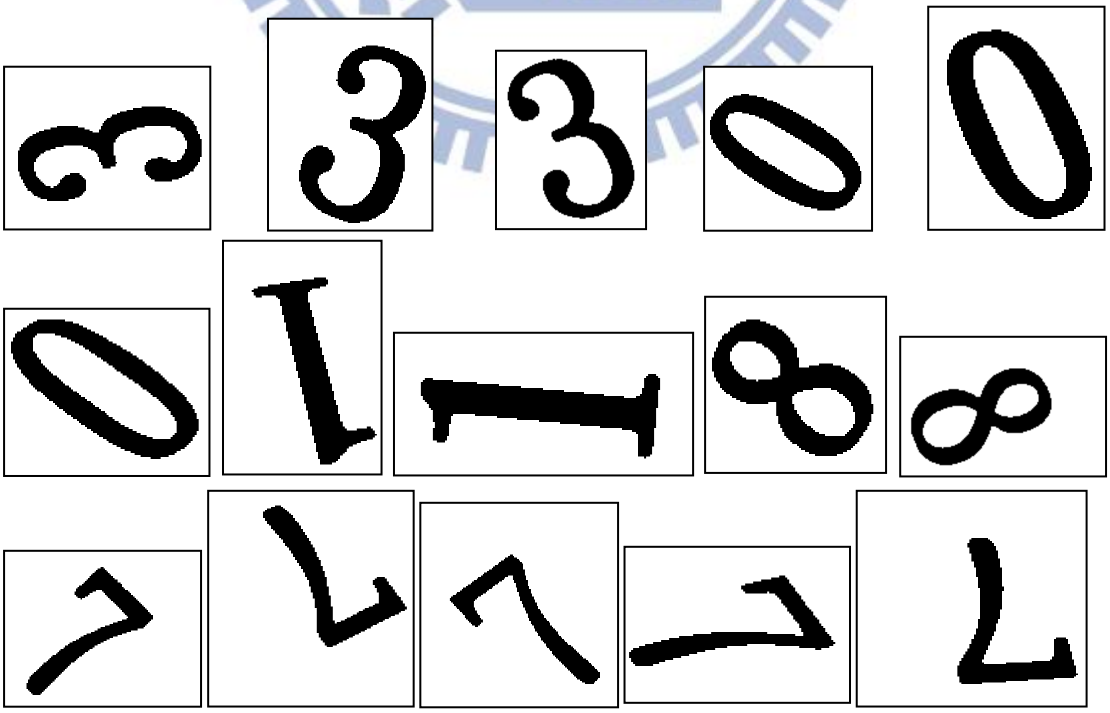


Fig. 4.9 Examples of training data for character recognition.

Table. 4.4 Different training data with CRNN

	training data	testing data	error	Accuracy rate
CRNN	137700	6000	232	99.6%
CRNN2	85262	6000	687	88.5%
CRNN3	23000	6000	1344	77.6%

Table. 4.5 Different condition with CRNN

	Condition	training data	testing data	error	Accuracy rate
CRNN	X	137700	6000	232	99.6%
CRNN4	With I_{13}	137700	6000	193	99.7%
CRNN5	Without I_{12}	137700	6000	583	99.0%

4.2 Part II: Intelligent system

Three intelligent neural networks are respectively proposed to detect moving word card in special color, to extract color and to recognize characters. In the research, we find out that supervised learning neural networks can replace the algorithm like morphology operation and detect moving object in two or more special colors. Fig.4.10 shows the system flowchart. Blue colors are the parts we use with intelligent neural networks and the yellow parts can be used either. The experiment result about morphology operation with intelligent neural networks will be expressed, considering the executing time, we don’t use all of the intelligent neural networks.

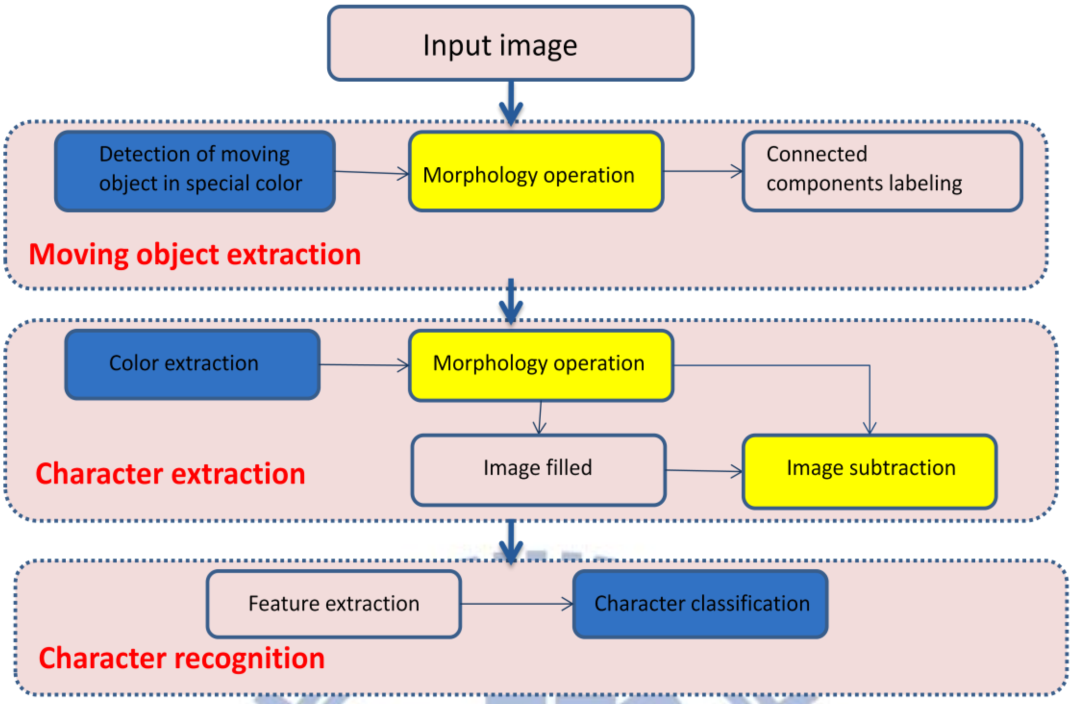


Fig.4.10 System flowchart.

4.2.1 Morphology operation with Artificial Neural Network

This chapter presents the morphology operation based on the artificial neural network, which structure is similar to CRNN as shown in Fig. 4.11, called morphology-operation neural network or MONN in short. MONN contains 25 neurons from m_1 to m_{25} for input layer, 10 neurons for hidden layer with the log-sigmoid function, and 9 neurons with from e_1 to e_9 for output layer log-sigmoid transfer function. The inputs of the neurons m_i, i=1, 2,…, 25, are logical value of a 5x5 range retrieved from the original image as shown in Fig. 4.12(a). The outputs of the neurons e_i, i=1, 2,…, 9, are the morphology operation results corresponding the central 9 pixels of the 5x5 range. The learning process of MONN based on the 5x5 range for the morphology operation as shown in Fig. 4.12(b), from left to right and downwards for the entire image.

Two kinds of morphology operation output pair are used, erosion (3.9) structuring element with radius 3 called MONNE and dilation (3.10) structuring element with radius 4 called MONND. Two kinds of morphology-operation neural network used the same artificial neural network structure which shown in Fig. 4.11. Fig. 4.13 shows an example of erosion and dilation using the MONNE and MONND.

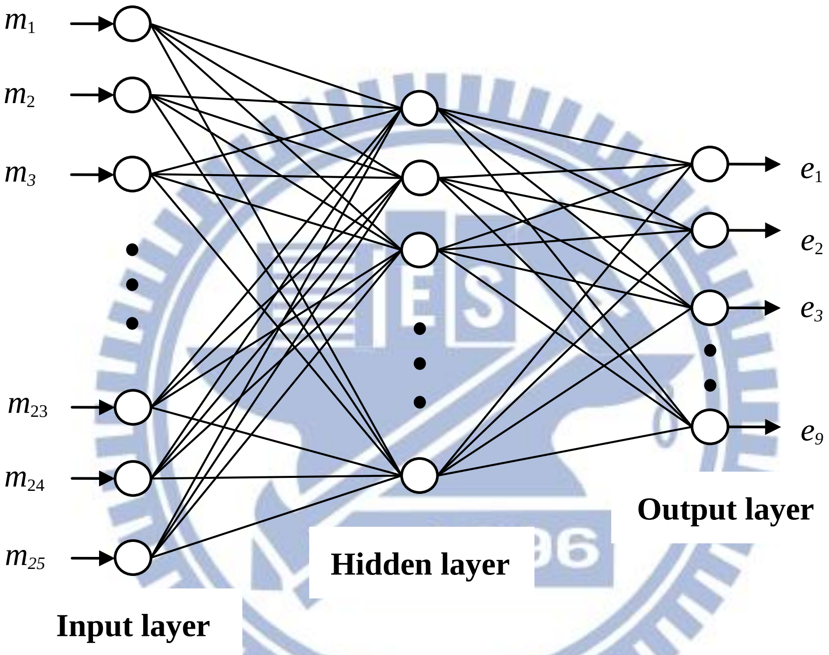
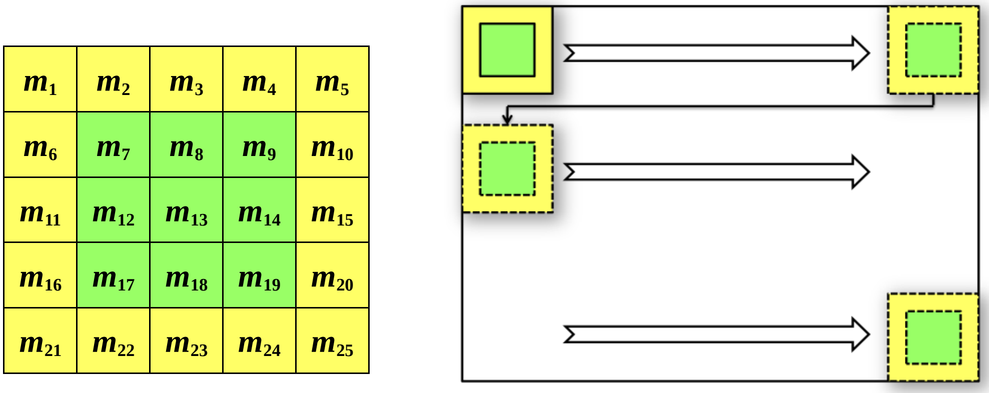


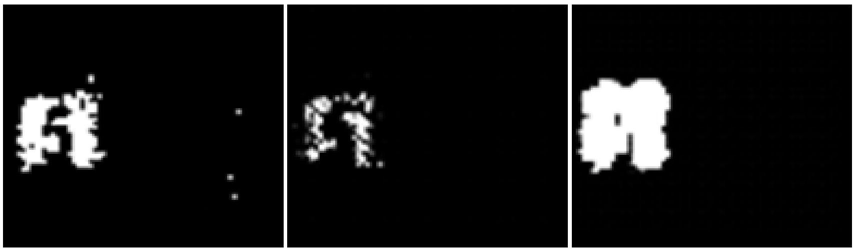
Fig.4.11 Neural network structure for morphology operation.



m_1	m_2	m_3	m_4	m_5
m_6	m_7	m_8	m_9	m_{10}
m_{11}	m_{12}	m_{13}	m_{14}	m_{15}
m_{16}	m_{17}	m_{18}	m_{19}	m_{20}
m_{21}	m_{22}	m_{23}	m_{24}	m_{25}

(a) (b)

Fig. 4.12 (a) Square block (b) Scanning the image.



(a) (b) (c)

Fig. 4.13 (a) Original image.
(b) The result with using MONNE. (c) The result with using MONND.

4.2.1 Detect Moving Object in Several Colors with Artificial Neural Network

In this experiment, we implement supervised learning artificial neural network that structure is same as MCNN as shown in Fig. 3.3, called several color neural network or SCNN in short. SCNN contains 6 neurons for input layer, 7 neurons for hidden layer with the log-sigmoid function, 1 neuron for output layer with the log-sigmoid function.

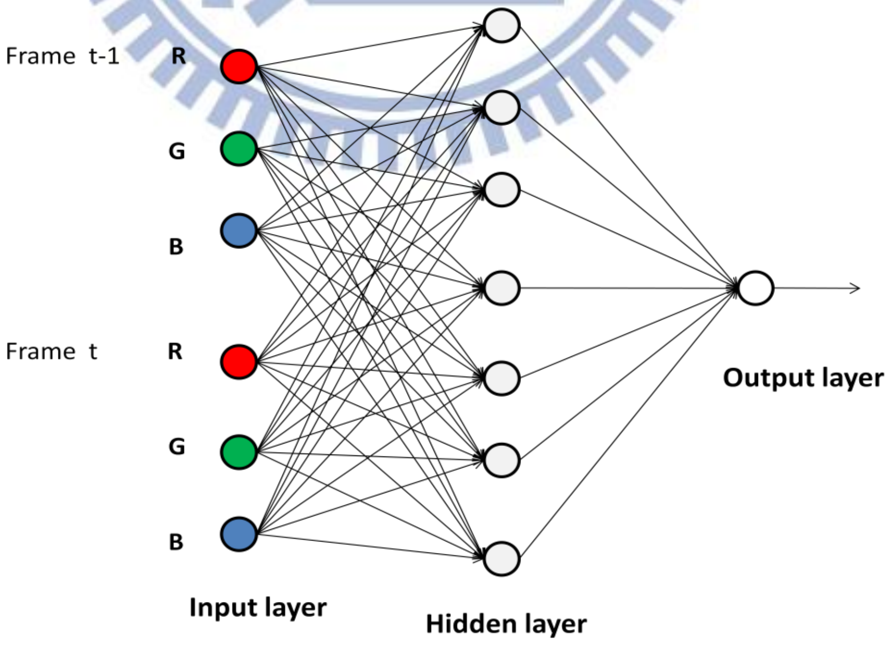
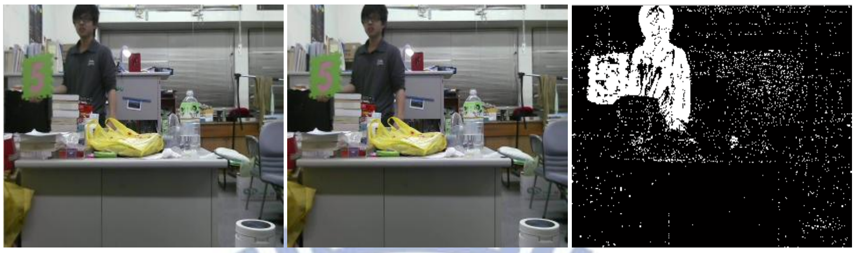


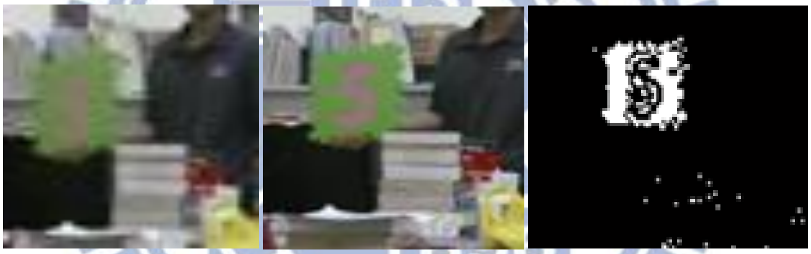
Fig.4.14 Neural network structure for SCNN1 and SCNN2.

Two kinds of situation of output pair are used, the one is moving object detection (SCNN1) and the other is moving object detection with green and pink color (SCNN2). Two kinds of SCNN neural network used the same artificial neural network structure which is same as MCNN as shown in Fig. 4.15. Fig. 4.16 shows an example of using the SCNN1 and SCNN2.



(a) (b) (c)

Fig. 4.15 (a) Input image (T=t-1). (b) (T=t). (c) The result with using SCNN1.



(a) (b) (c)

Fig. 4.16 (a) Input image (T=t-1).
(b) Input image (T=t). (c) The result with using SCNN2.

4.3 Part III: Execution time

In the previous chapter, we find out that supervised learning neural network can replace the algorithm like morphology operation and detect moving object in two or more special colors. The experiment in the section will show each execution time. Table 4.6 shows the MONNE's execution time in different block sizes, Table 4.7 shows the SCNN1's execution time in different block sizes and Table 4.8 will show the total execution time with different condition.

We find out that fewer outputs is a absolute way and input size must be proportional to the operation radius if we want to get the better performance. Although MONNE and MONND have different operations output design, with the same neural network structure, the execution time will be almost the same. As Table 4.6 and Table 4.7 shown, MONNE's execution time is between 0.032sec~0.062sec and SCNN1's execution time is between 0.036sec and 0.038sec. The executing time of morphology operation and image subtraction is quiet shorter in MATLAB R2010b, only 0.0025sec and 0.0001sec, respectively. As shown in Table 4.8, when system using MONNE, MONND and SCNN1 will let the execution time twice longer and the screen will be discontinuous. Considering the executing time, we don't use all of the intelligent neural networks in this thesis.

Table 4.6 MONNE's execution time in different block sizes.

Block size	Input neuron	Hidden neuron	Output neuron	Extraction time
m=4, n=2	16	10	2	0.046sec
m=5, n=3	25	10	9	0.037sec
m=7, n=5	49	10	9	0.032sec
m=9, n=5	81	10	25	0.062sec
m=9, n=7	81	10	25	0.035sec

Table 4.7 SCNN1's execution time in different block sizes.

Block size	The time to generate input	calculate time	The time to generate output	total time
m=1, n=1	0.004	0.03	0.004	0.038sec
m=3, n=3	0.002	0.032	0.002	0.036sec

Table 4.8 Compare average execution time in different condition.

	Average time	Average frame rate
Without Character Recognition	0.04sec	25
With Character Recognition	0.14sec	6.6
System with MONNE and MONND	0.22~0.27sec	3.7~4.5
System with MONNE,MONND and SCNN1	0.026~0.32sec	3~3.8

Chapter 5

Conclusions and Future Works

The purpose of this thesis is to build up a system for children to learn words in an interactive way. In this thesis, the developed intelligent system can recognize the character correctly in a moving word card from a sequence of images.

The system is designed in three steps, including potential object localization, character extraction and character recognition. In the first step, it is required to detect the moving object in special color, or word card, and then determine the location of the word card in the image. A supervised learning neural network (MCNN) is used to extract the color and detect the moving object simultaneously. After applying the MCNN, the region of the word card in green color is extracted from a sequence of images; unfortunately, some noise exists therein. Using morphology operation and connected components labeling (CCL), the noise is removed and the region of the word card could be located correctly.

In the second step, use another supervised learning neural network (GNN) to, and then apply the morphology operation to reduce noise. The word card is thus achieved as a binary image with the shape of the character on it. By generating a plain binary card, the character on the word card can be extracted by subtracting the plain binary card. Besides, the total number of character's pixels is used to determine whether the result is a character or not. In the third step, a scheme based on a set of concentric circles is adopted to extract the character features, and then feed the features into the third supervised learning neural network (CRNN) to recognize which word it is, the designed neural networks CRNN can robustly identify characters in different translation, size, tilt and angle of rotation. The overall system processing time is about 0.15s.

Three intelligent neural networks are respectively proposed to detect moving word card in special color, to detect color for character extraction and to recognize characters. Besides the three neural networks we proposed, we also find out that supervised learning neural networks can be used to execute the image subtraction algorithm, morphology operation, and the moving object detection in two or more special colors. Related experiments have been shown in Chapter 4. However, due to the requirement of real-time operation, the proposed system does not implement them by the neural networks.

The proposed intelligent system has been demonstrated to be successful in intelligent word card image recognition system, which is an important field in robot research. For the future, some related research should be further investigated such as noise reduction, word card in multiple colors, more characters or even images.

Reference

[1] S. Haykin, *Neural Networks: A Comprehensive Foundation (2nd Edition)*, Prentice Hall, 1998

[2] C. G. Looney, *Pattern Recognition using Neural Networks: Theory and Algorithm for Engineers and Scientists*, NY, USA: Oxford University Press, 1997.

[3] K. Omar and A. Al-Shatnawi, “*A comparative studybetween methods of Arabic baseline detection*,” International Conference on Electrical Engineering and Informatics: ICEEI, 2009.

[4] A. E. Bryson and Y. C. Ho, *Applied Optimal Control* **:** *Optimization, Estimation, and Control*, New York, Taylor & Francis, 1975.

[5] P. J. Werbos, *Doctoral dissertation: Beyond regression: New tools for prediction and analysis in the behavioral sciences*, PhD thesis, Harvard University, 1974.

[6] Y. LeCun, “*A Learning procedure for asymmetric network*,” In Proceedings of Cognitiva, Paris, pp. 599-604, 1985.

[7] J. Heikkilä and O. Silvén, , “*A real-time system for monitoring of cyclists and pedestrians,”* IEEE Proc. on Visual Surveillance, pp. 74-81, 1999.

[8] C. Cédras and M. Shah, “*Motion-based recognition: a survey,*” Image and Vision Computing .vol.13, No.2, pp.129-155, March, 1995.

[9] A. Neri, S. Colonnese, G. Russo and P. Talone, “A*utomatic moving object and background separation,*” Signal Process. pp.219-232,1998.

[10] C. Rafael, E. Woods and L. Steven, *Image Processing: Digital Image Processing Using Matlab(2nd Edition),* McGraw-Hill Education, LLC, USA, 2011.

[11] N. Rahman, K. Wei and J. See,*:* “*RGB-H-CbCr Skin Color Model for Human Face Dection,*” International Symposium on Information & Communications Technologies 2006.

[12] L. Bretzner, I. Laptev and T. Lindeberg, "*Hand Gesture Recognition using Multi-Scale Colour Features, Hierarchical Models and Particle Filtering,*" IEEE, Automatic Face and Gesture Recognition (AFGR 02), 2002.

[13] Q. Peng and X. Zhang, *Sensitive Image Recognition Technology Based on Eigenvectors,* Academic Journal:Southwest Jiaotong University, 2007.

[14] V. Vezhnevets, V. Sazonov and A. Andreeva, *A Survey on Pixel- Based Skin Color Detection Techniques,* Graphicon: 2003.

[15] A. S. Abdallah, A. L. Abbott and M. A. El-Nasr, "*A New Face Detection Technique using 2D DCT and Self Organizing Feature Map,*" World Academy of Science, Engineering and Technology, vol. 24, pp. 15-19, 2007.

[16] L. A. Torres-Méndez, J. C. Ruiz-Suárez and G. Gómez, "*Translation, Rotation, and Scale-Invariant Object Recognition,*" IEEE Transactions on Systems Man and Cybernetics Part C Applications and Reviews, vol. 30, no. 1, 2000.

[17] L. A. Torres-Mendez, "*Invariant 2-D object recognition,*" Graduate Program in Informatics, ITESM-Campus Morelos, México, 1995.

[18] Y. N. Hsu and H. H. Arsenault, "*Optical pattern recognition using the circular harmonic expansion,*" Appl. Opt., vol. 21, no. 22, pp. 4016–4019,1982.