HANDWRITTEN CHARACTER RECOGNITION IN TABLET-BASED APPLICATION

MYAT THIRI WAI

M.C.Tech. **FEBRUARY 2020**

HANDWRITTEN CHARACTER RECOGNITION

IN TABLET-BASED APPLICATION

By

Myat Thiri Wai

B.C.Tech.

A Dissertation Submitted in Partial Fulfillment of the Requirements for the Degree of

Master of Computer Technology

(M.C.Tech.)

University of Computer Studies, Yangon

FEBRUARY 2020

ACKNOWLEDGEMENTS

First and foremost, I would like to express my gratitude and sincere thanks to **Dr. Mie Mie Thet Thwin**, Rector, University of Computer Studies, Yangon, for allowing me to carry out this thesis.

I owe my special gratitude to my respectful supervisor, **Dr. Khin Than Mya**, Professor, Dean of Faculty of Computer Systems and Technologies, Head of Embedded System Lab, University of Computer Studies, Yangon (Myanmar), for giving me a lot of suggestion, encouragement and motivation on my thesis work to reach the goal. I am deeply grateful to her, for giving me a chance to discuss my work freedomly and also for strongly recommendation for double degree program. Without her assistance and dedicated involvement in every step throughout the process, this thesis would have never been accomplished.

I heartily thank and appreciate my supervisor and special advisor, **Dr. Thi Thi Zin**, Professor, Graduate School of Engineering, University of Miyazaki, for accepting me as her student by Master Double Degree Program. I would like to give my special thanks again to her, for giving me good suggestion, moral support and sympathetic attitude at every point. Through her supervision, I got so many motivation in my practical work and also perceived a lot of experiences.

Special thanks are due to my co-supervisor, **Professor Mitsuhiro YOKOTA**, one of my thesis committee members, Dean of Faculty of Engineering, University of Miyazaki, for his advice and comments in carrying out the research work very effectively.

I am immensely indebted to the member of thesis committee, **Professor Kunihito YAMAMORI**, Graduate School of Engineering, University of Miyazaki, for giving insightful remark, comments and suggestions to widen my research. He gave the attention and kindly directed towards the progression to work out my research in advance.

I also thank to my teacher, **Dr. Mi Mi Nge**, Managing Director, Realistic Infotech Grup Co., Ltd, an external teacher of this thesis committee, for her precious comments and advice on this work.

I am also thankful to **Dr. Thi Thi Soe Nyunt,** Professor, Dean of Faculty of Computer Science, Head of Computer Graphics and Visualization Lab, for her kind support and attention about DDP program.

I would like to respect and thank **Dr. Myint Myint Sein**, Professor, Head of Geographic Information System Lab, for her valuable advices and suggestions to this thesis.

I want to show special thanks to my teacher, **Dr. Thet Thet Khin**, Professor and Coordinator of Faculty of Computer Systems and Technologies, University of computer Studies, Yangon, for her suggestion and help in this work.

Moreover, I would like to give my special gratitude to **Daw Khin Mar Kyu**, Lecturer, Department of English, University of Computer Studies, Yangon, for editing this thesis from the language point of view.

I am also grateful to the university staffs for their unfailing support and assistance to this work.

I am extremely grateful to my parents for their love, prayers, caring and sacrifices for educating and preparing me for my future. I am very much thankful to my teachers and my relatives for their love, understanding, prayers and continuing support to complete this research work. Also I would like to give my thanks to my elder sister for her kindly support and valuable prayers. My Special thanks goes to my friends for the keen interest shown to complete this thesis successfully. This accomplishment would not have been possible without them.

Finally, my thanks goes to all the people who have supported me to complete the research work directly or indirectly.

ABSTRACT

Portable tablet PCs are very useful in relevant industry of this age because tablets are elegant in appearance and convenient to use. Important things are noted on tablet by handwriting easily in respective industry. Recognition of handwritten characters automatically on tablet like human's brain is also necessary to be more convenient. To split each character of different handwritten styles is very difficult and it is the main challenging of handwritten character recognition. The previous handwritten character segmentation approaches are still continuing in different problems because of different handwritten styles. The combination of sliding windows, Region of Interest (ROI) box and Convolutional Neural Network (CNN) are used to execute recognition based segmentation (implicit) of handwritten characters.

This system is intended to perform both segmentation and recognition of tablet based application input handwritten characters. Handwritten data are collected from 24 members of our laboratory using three tablets PC models to perform the experiments.

CONTENTS

LIST OF FIGURES

Figure **Page**

List of Tables

CHAPTER 1

INTRODUCTION

1.1 Introduction

This thesis presents recognition of tablet input handwritten characters like human's brain. The aim of this work is to promote in touch-screen tablet based application for kindergarten education. This system is intended to perform both segmentation and recognition of tablet based application input handwritten characters. The new handwritten character segmentation approach is introduced in this thesis. This chapter addresses the overall introduction of this thesis. The objectives and the organization of the thesis are also described in this chapter.

Nowadays, modern tablets are widely used in the contribution of universal access to education, equality in the exercise of teaching and learning, aiming for more efficient management and administration. The recognition of handwritten characters on these tablets becomes a necessary consequence in this technology age. Automatic handwritten character recognition can be said as the imitation of the intelligence of human's brain. Although optical character recognition has been developed, the challenging of the handwritten character recognition is still remaining to solve many problems.

Since last few decades and advancement in technology, computers interact more effectively with humans and with the natural world e.g. speech recognition, handwritten recognition, gesture recognition etc. However, humans are outperforming far better than machines in recognizing patterns. Some of the tasks are generally easy for humans, such as identifying the human voice based on frequency and pitch, recognizing and differentiating aroma of flower and a food, identifying characters etc. These kinds of perceptual problems are difficult for the computer because of voluminous data with composite and hidden information each pattern usually contains.

Character recognition, usually abbreviated to Optical Character Recognition (OCR), is the mechanical or electronic translation of images of handwritten, typewritten or printed text into machine-editable text. It is a field of research in pattern recognition, artificial intelligence and machine vision. Though academic research in the field continues, the focus on character recognition has shifted to implementation of proven techniques. OCR is a scheme, which enables a computer to learn, understand, improvise and interpret the written or printed character in their own language, but present correspondingly as specified by the user. OCR uses the image processing technique to identify any character machine printed or handwritten. A lot of work has been done in this field. However, a continuous improvisation of OCR techniques is being based on the fact that algorithm must have higher accuracy of recognition.

Handwritten character recognition (HCR) is the ability of a computer to receive and interpret handwriting from sources such as paper documents, photographs, touch-screens and other devices. The image of the written text may be sensed "off line" from a piece of paper by optical scanning (optical character recognition) or intelligent word recognition. A handwriting recognition system handles formatting, performs correct segmentation into characters, and finds the most plausible characters.

Handwritten character recognition is challenging OCR environment. OCR is a process of recognition of machine printed or handwritten characters from images. Machine printed character recognition has been developed whereas handwritten character recognition is difficult and still continuing with different problems in OCR environment. Most of handwritten recognition systems have still the problems in preprocessing and segmentation to recognize correctly. It is very difficult to segment different handwritten formats by different people. To segment and recognize tablet input handwritten characters is more difficult because there are many noisy and different handwriting styles. Researchers in [1, 2] accepted that segmentation is the most important part in recognition of handwritten character because bad segmentation errors lead to the recognition errors of the system. Segmentation of unconstraint handwritten characters of different people tend to unexpected difficult challenging. The method used to segment the handwritten character is the important role in this system.

The main challenging of handwritten character recognition is splitting each character of unrestricted handwritten scripts and there is no complete solution yet according to our knowledge. In this system, the combination of sliding windows, Region of Interest (ROI) box and Convolutional Neural Network (CNN) are used to execute implicit segmentation of handwritten characters. For performing the experiments, our own dataset is constructed by collecting handwritten data from 24 members of our laboratory using three different tablet PC models.

As portable computers become more personal and are made smaller, they reach some physical limitations for having a keyboard. For pocket size computers one cannot use an efficient keyboard. For these computers alternative ways of man-machine communication are necessary. The most efficient way of solving this problem is to use communication skills of man, which have developed for thousands of years namely, speech and handwriting. There are certain merits and drawbacks to both of these techniques, thus comprising the reason why we still communicate using both methods. One of the most obvious reasons that handwriting recognition capabilities are important for future personal systems is the fact that in crowded rooms or public places one might not wish to speak to his computer due to the confidentiality or personal nature of the data. Another reason is that it might be annoying to us if someone sitting next to us in the train or airplane keeps speaking to his machine. Another reason for the practicality of a system, which would accept hand-input, is that with today's technology it is possible to have handwriting recognition in very small hand-held computers.

In this thesis, segmentation of the tablet based application input handwritten characters is proposed with the purpose of using in kindergarten education. One of the main objectives of this system is to introduce new handwritten characters segmentation. In this system, segmentation of handwritten words and two digits number into each character and recognition of segmented character are implemented to test what characters are write on tablet. This system is proposed to use in touch-screen tablet based application for kindergarten education as the application in [3].

The handwritten data were collected from the laboratory members using tablet. The three tablet PC models were used in construction the own dataset. The dataset can be categorized into seven categories: animals, body parts, education, fruits, drinks, numbers, and others. There are 251 data with 8 types of animals, 162 data with 5 types of body parts, 185 data with 6 types of education, 359 data with 12 types of fruits, 172 data with 5 types of drinks, 139 data of two digits numbers, and 335 data with 13 types of others. The total data of own dataset are 1603 images. For training the CNN, the training and testing dataset are prepared using Extended Modified National Institute of Standards and Technology (EMNIST) dataset [4]. There are 62 classes: 0 to 61 in training and testing datasets including numbers ‘0’ to ‘9’, capital alphabets ‘A’ to ‘Z’, and small alphabets ‘a’ to ‘z’. Training dataset included 697,932 data images and testing dataset included 116,324 images. The architecture layers of the CNN are implemented in MATLAB 9.5 version of R2018b, to construct the MatConvNet [5] and that implemented CNN is trained with the prepared training dataset.

1.2 Objectives of Thesis

The main objective of this thesis is to implement handwritten character recognition of touch-screen tablet based application input handwritten characters to use in kindergarten English education. Some sub-objectives are in the following:

- To introduce new handwritten characters segmentation process
- To recognize the segmented characters correctly to determine what characters are written on tablet
- To promote the handwritten character recognition application for kindergarten education
- To identify handwritten characters with the use of convolutional neural network
- To construct suitable convolutional neural network and train it.

1.3 Organization of Thesis

This thesis is organized as five chapters. Chapter 1 presents about the overall contents of the thesis. In chapter 2, some previous works related with this thesis are described. In chapter 3, the detailed step by step process of the proposed system is explained to implement in the work. The experimental results of the implemental work and the performance analysis of experiments are presented in chapter 4. Finally, the discussions and overall conclusions are summarized in chapter 5.

CHAPTER 2

LITERATURE REVIEWS

2.1 Handwritten Character Segmentation

Various methods of segmentation in the literature can be categorized into three: explicit, implicit, and holistic. Pure segmentation [6-7] is explicit, segmentation free recognition [8] is holistic, and recognition based segmentation [9] is implicit. Our approach is recognition based segmentation by combination of convolutional neural network, sliding window and finding region of interest.

In the region-based segmentation [6], the segmentation path is determined to split the only two touched characters into two sub images by defining the three zones: left, right, and middle. That paper presents an intelligent technique for segmentation of off-line cursive handwritten words particularly on touching characters problem. Self Organization Feature Maps (SOM) was implemented to identify the touching portion of the cursive words. The authors introduced an improved segmentation-based approach to separate touching handwritten character pairs.

Locating the segmentation points based on the analysis of the character's geometric features [7] can be done after thinning the word image to get the stroke width of a single pixel. The paper proposed only a new vertical segmentation algorithm in which the segmentation points are located after thinning the word image to get the stroke width of a single pixel. That system did not include recognition. The knowledge of shape and geometry of English characters is used in the segmentation process to detect ligatures. The segmentation approach of that paper is tested on a local benchmark database and the segmentation accuracy of 83.5% was obtained.

An offline handwritten word recognition system [8] combines Neural Neetworks (NN) and Hidden Markov Model (HMM) by employing a simple vertical slicing method to divide the handwritten word images into a left-to-right sequence of vertical slices. Using a fast left-right slicing method, they generate a segmentation graph that describes all possible ways to segment a word into letters. That paper used the IRONOFF database. This paper recognizes handwritten cursive words using recognition based segmentation method. This paper gives the comparison between two methods. The first recognition system uses combination of Neural Network and HMM for recognition. In second method, discrete HMM is used. In first method, pre-segmentation of word is performed using segmentation graph. Neural network calculates the probability for each letter hypothesis in graph and then HMM computes likelihood for each word in lexicon by adding the probability along each possible path in graph. In second method, 140 geometric features were extracted from each segment which is separated by pre-segmentation. This features were converted to single symbol by vector quantization (VQ), and finally, each word in lexicon word is recognized by calculating the likelihood.

Analyzing character's geometric features and ligatures [10] were strong points for segmentation in cursive handwritten words. The columns would be saved as candidate segmentation column (CSC) if the sum of foreground pixels was 0 or 1. There was the over-segmentation problem by putting many CSC for ligatures-between-characters (connection between characters) and ligatures-within-characters (connections within characters) such as 'm', 'u', 'w', etc. A simple criterion is adopted to come out with fine segmentation points based on character shape analysis. Finally, the fine segmentation points are fed to train neural network for validating segment points to enhance segmentation accuracy.

A rule based segmentation approach and artificial neural network for the character segmentation of unconstrained handwritten words [11] were integrated for segmentation performance. To enhance segmentation accuracy, an artificial neural network (multilayer perceptron MLP) was employed to overcome the over-segmentation. The proposed character segmentation approach analyses the characters geometric features to identify ligatures and characters. There were three main phases: segmentation of closed characters, identification of boundaries of open characters, and applying MLP. MATLAB was used for all experiments performed.

For cursive handwriting, some of the successful results had been obtained with the use of techniques that possess tightly coupled segmentation and recognition components [12]. The approach was based on the use of generating multiple possible segmentations of a word image into characters and matching these segmentations to a lexicon of candidate strings. The segmentation process used a combination of connected component analysis and distance transform-based connected character splitting. The approach wa broken into two parts: a distance transform-based computation of multiple segmentations of the word image and neural network based computation of match scores.

2.2 Handwritten Character Recognition

Recognition accuracy of the image depends on the sensitivity of the selected features and type of classifier used.

A Convolutional network has a benefit over other Artificial Neural networks in extracting and utilizing the features data, enhancing the knowledge of 2D shapes with higher degree of accuracy and unvarying to translation, scaling and other distortions [13]. A class of multilayer sustain forward system called Convolutional network was taken into consideration in this paper. This paper was going to implement the network using keras deep learning inbuilt python library. They were going to take the MNIST dataset for training and recognition.

The comparison between conventional and directional feature extraction method was done [14]. Twelve directional features were used for recognition of alphabets and numerals. In order to extract directional feature, gradient features of each pixel were extracted the gradient values were mapped onto 12 direction values to the angle span of 30 degree between any two adjacent direction values. Feature vector of each class was obtained by taking mean of feature matrix of each class. The similarity between testing feature vector and feature vector of all the classes was calculated, thus, testing image belongs to the class which has the highest similarity.

Handwritten character recognition of lowercase English alphabets was performed by using binarized pixels of the image as features and multilayer back-propagation neural network as classifier [15]. The character image was binarized, filtered and resized to 15X12, thus feature vector of size 180 was created of each character which was given to neural network for its training. MSE (mean square error) was used as cost function. The use of binarization features with back-propagation neural network classifier gave classification accuracy of 85.62%. It had simplicity of features as direct pixel values were taken.

The researchers of [16] has designed Neural Network based recognition system. They used different neural network topologies- back propagation neural network, nearest neighbor network and radial basis function network for same training dataset. They compared the performance of each network and optimized the number of neurons in hidden layer which is not dependent on initial value and concluded that combination of standard feature extraction technique with feed forward back propagation.

The recognition separated handwritten cursive characters [17] is one of the recognition challenges. Here different features were extracted among them two features modified edge map and multiple zoning were proposed by authors. Total nine features were extracted, and drawback of each feature was overcome by other. Each feature was individually given as input to nine multilayer perceptron network. Output of all this classifier were combined with each other by different rule like sum rule, product rule, max rule, mean rule etc. Among them, trained MLP combiner gave maximum result. Among proposed features, modified edge map feature gave highest result.

The recurrent neural network (RNN) had been shown to perform better than HMM for several sequence-decoding problems, in particular handwriting recognition [18]. One of the possible reasons was that RNNs are discriminative models, while standard HMMs are generative. This paper proposed an alternative approach based on a novel type of recurrent neural network, specifically designed for sequence labelling tasks where the data were hard to segment and contains long range, bidirectional interdependencies. This paper proposed an alternative approach, in which a single RNN was trained directly for sequence labelling.

2.3 Conclusion

As part of our work, the segmentation and recognition of the handwritten characters taken on the tablet are designated. In contrast to other previous work, the tablet based application input handwritten character recognition system is also implemented for using in kindergarten education. This system is not only to overcome the challenging of unconstraint handwritten character segmentation but also to recognize each segmented character correctly.

CHAPTER 3

PROPOSED SYSTEM

3.1 Introduction

Segmentation is the main important part in this system for improving the recognition rate. To implement the system of handwritten character recognition, the image data were firstly collected by using the tablets. After that, finding ROI boxes, sliding window segmentation and CNN were performed on our self-collected image data to segment into each character. And then, the system gives the segmented classifiable characters. Finally, those pure classifiable characters are recognized by the classifier trained by using CNN.

An overview of this proposed system is shown in **Figure 3.1**. Handwritten input image taken using touch-screen tablet is loaded as input image to this system. Firstly, the input image is pre-processed to get the binary image. Binarization inverts the original input image into a binary image. Then, segmentation process splits each character clearly from pre-processed image in recognition based character segmentation stage. Two main segmentation steps - outer ROI segmentation and sliding window segmentation - are performed in this stage and slant correction is processed after outer ROI segmentation to correct the slanting of the each outer ROI. In outer ROI segmentation step, firstly, ROI rectangle boxes are defined for each connected component and each outer ROI box is classified and determined whether it is a valid qualified character ROI or not. Qualified outer ROI is kept as a valid character and unqualified outer ROI is fed into sliding window segmentation step. Slanted outer ROI segmented characters are corrected into normal position in slant correction process. Sliding window slides the outer ROI vertically with various sizes. Each slide is classified with threshold parameter and the most suitable windows so that valid characters are determined. Finally, this system classify each segmented character correctly and print out the recognized characters.

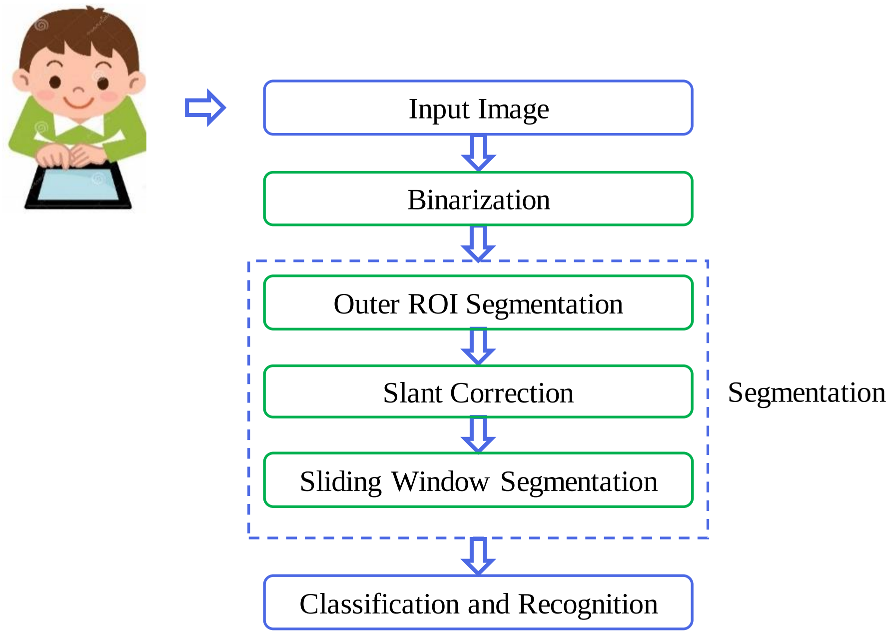


Figure 3.1 Overall System Architecture

3.2 Experimental Environment

Handwritten data are collected from 24 members of our laboratory to perform the experiments. Our own dataset is constructed by using three tablet PC models: ASUS ZENPAD 10 SPECS, HUAWEI MediaPad T310, and Gecoo Tablet A1.

3.3 System Initialization

The system is initialized after inserting the handwritten image data. After that, pre-processing, segmentation and recognition steps are following by the next steps. In the next sections, pre-processing, the two segmentation steps and classification are described in detail.

3.4 Binarization

Before segmenting the characters of handwritten image, pre-processing is performed on the input handwritten image. There are two processes in pre-processing stage: binarization, and slant correction. Converting the original image into binary image is performed by applying the Otsu's thresholding that utilizes discriminant analysis to find the maximum separability of classes. The optimal threshold is calculated based on the Otsu's method from (3.1), where *'t'* means the global thresholding. According to the Otsu's method, σ_w^2, the within-class variance is calculated using the background weight, W_b, the background variance, σ_b^2, the foreground weight, W_f, and the foreground variance, σ_f^2, respectively based on the threshold '*t*'. The final threshold is selected where also has the lowest sum of weighted variances. All pixels with a level less than the threshold are background and all pixels with a level equal to or greater than the threshold are foreground.

$$\sigma_w^2(t) = W_b(t)\sigma_b^2(t) + W_f(t)\sigma_f^2(t) \qquad 3.1$$

3.5 Segmentation

The combination of sliding windows, ROI box and CNN are used in segmentation of handwritten characters. There are two main steps in segmentation: outer ROI segmentation and sliding window segmentation. Pre-processed image is segmented by applying ROI boxes and each outer ROI is determined a valid character or not by using trained classifier. The valid character outer ROI's coordinates and dimensions are kept as a qualified Outer ROI. Unqualified outer ROI is inserted to sliding window segmentation process. Firstly, sliding window splits the unqualified outer ROI into vertical slides of various widths [19]. And then, inner ROI is defined on each window slides and trained CNN classifier determines the valid character window slides. After extracting the valid sliding windows, the coordinates and dimensions of these qualified sliding window is kept as a valid character.

(a) Outer ROI Segmentation

Outer ROI segmentation process is shown in **Figure 3.2**. After pre-processing the input image, outer ROIs are defined by labelling process. After that, region combining for small characters, 'i' and 'j', and small dot noises removing are operated to detect the region exactly. Normalization of outer ROI is performed to insert to the trained classifier. The classifier determine that outer ROI is a qualified or unqualified character ROI using threshold parameter. If the outer ROI is qualified, it is determined as a valid character and the coordinates and dimensions of that outer

ROI are stored. Otherwise, sliding window segmentation is applied on that outer ROI. In outer ROI segmentation, there are three main processes:

- Finding Outer ROI
- Pre-processing of Outer ROI
- Classification of Outer ROI

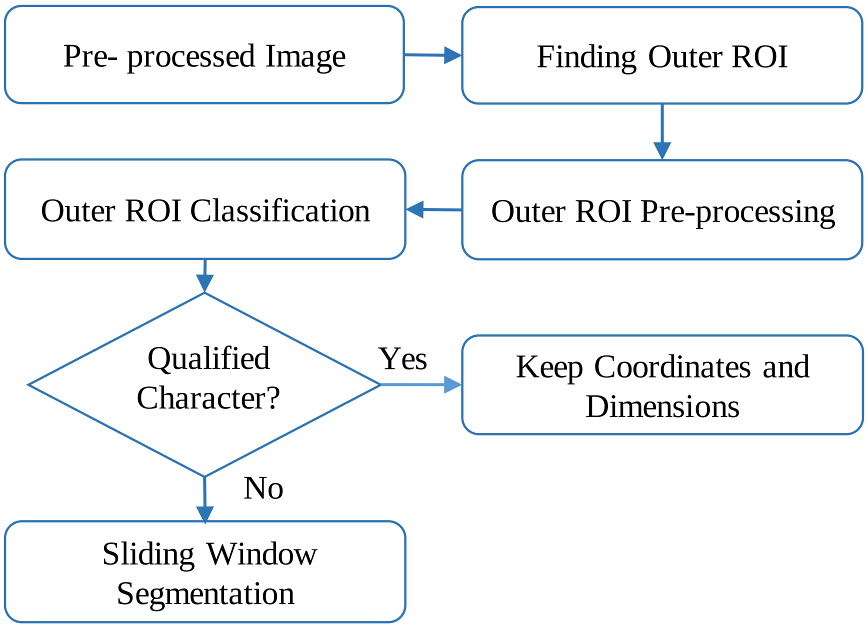


Figure 3.2 Overview of Outer ROI Segmentation

(b) Sliding Window Segmentation

The overview of the sliding window segmentation is shown in **Figure 3.3**. If there is unqualified outer ROI, that ROI is segmented into vertical slides by applying the sliding window processing. After that, inner ROI is specified to extract the character region in sliding window, and that inner ROI is normalized into classifiable size. The normalized and resized image is fed to the classifier to find the valid character. If the most possible valid character is found, the coordinates and dimensions of that sliding window are stored and it is determined as a valid character. There are four main processes in sliding window segmentation:

- Sliding Window Processing
- Finding Inner ROI
- Inner ROI Pre-processing
- Classification of Inner ROI.

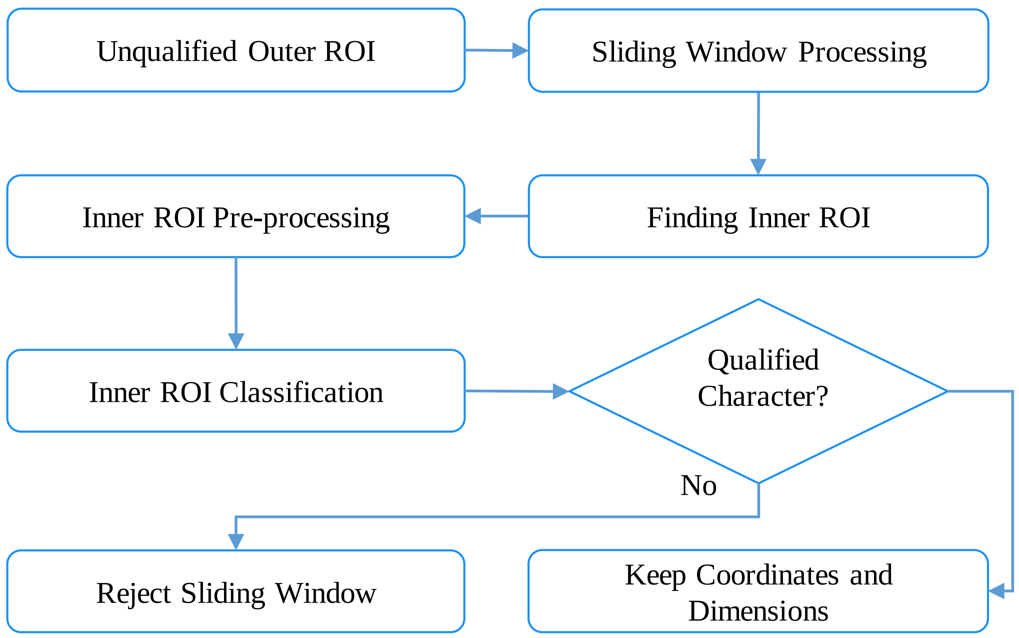


Figure 3.3 Overview of Sliding Window Segmentation

3.5.1 Finding Outer ROI

The process of finding outer ROI includes the following two main processes:

(a) Region Labelling

(b) Region Combining and Small Dot ROI Removing.

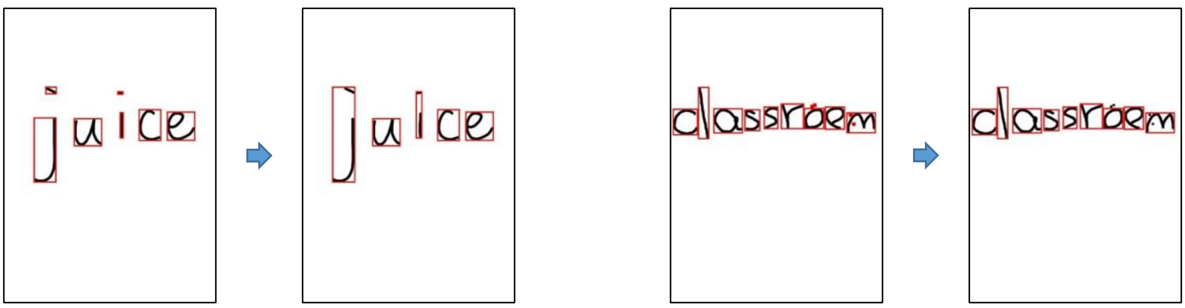
Firstly, all connected elements are labeled and bounded with rectangle boxes. Secondly, combining the two ROIs for bounding the whole character ‘i’ or ‘j’ is executed and the unnecessary small dot ROI noise are removed.

(a) Region Labelling

Outer ROI is defined by applying the bounding rectangle of all connected black pixels by labelling based on the binary image. Labeling to the all connected elements is the process of identifying connected elements in an image and assigning a unique label to each connected element. A connected element in a binary image is a collection of pixels that make up a connected group.

(b) Region Combining and Small Dot ROI Removing

In labelling and defining the ROI, characters, ‘i’ and ‘j’, are split into two ROIs in small dot and main body part of character. To detect the whole character ‘i’ or ‘j’ with only one bounding box, it is needed to combine the two labeled ROI boxes into only one labeled ROI bounding box as shown in **Figure 3.4 (a)**. If there is a very small ROI, a narrow width ROI is searched from its neighboring ROIs. If one of the neighboring ROIs’ features are with long height and narrow width, and the small dot ROI is in the upper of that neighboring ROI, those two ROIs are combined to bound the whole character: ‘i’ or ‘j’. If the small dot ROI is not lower of all neighboring ROIs or the features of all neighboring ROIs are not similar with characters ‘i’ or ‘j’, that small dot ROI noise is removed by filling the foreground pixels with background pixel value. Removing the outer ROI small noises is shown in **Figure 3.4 (b)**. Finally, the accurate and clear outer ROI specification is got.



(a) The two ROIs combination **(b) Removing the ROI noises**

Figure 3.4 Region Combining and Small Dot ROI Removing

3.5.2 Slant Correction

The slant correction is the transforming the image into the normal character position. The slant correction is performed by employing the horizontal shear transformation of (3.2) on the whole word image. A horizontal shear is a function that takes a generic point with coordinates (x, y) to the point ($x + sy$, y) ; where, 's' is a fixed parameter, denotes shearing factor which is specified by computing the slant of each character. Orientation, the angle between the main vertical axis of the character and horizontal line is computed to predict the slanting of the words. If the orientation of all characters are positive, that word is assumed slanting to the right and if the orientation are negative value, that word slants to the left. If the average orientation of slanting character to the right is less than the 75, the image is transformed to the left by s. The image is not corrected if the average orientation is greater than 75. On the other hand, the average orientation of the image that is slanted to the left is greater than -75, the image is transformed to the right with s. 'x' and 'y' are original pixel coordinates of slanted image and, 'x'' and 'y'' are the new coordinates of slant corrected images.

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} x + sy \\ y \end{pmatrix} = \begin{pmatrix} 1 & s \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \qquad 3.2$$

The effect of this mapping is to displace every point horizontally by an amount proportionally to its y coordinate. Any point above the x-axis is displaced to the right (increasing x) if $s>0$, and to the left (decreasing x) if $s<0$. Points below the x-axis move in the opposite direction, while points on the axis stay fixed.

3.5.3 Pre-processing of Outer ROI

Before classification, the outer ROI is pre-processed into classifiable image. Noise removal, thickening and normalization processes are included in pre-processing for classification. The detail of these three processes of pre-processing are explained in next section.

The trained CNN classifier is used to classify each outer ROI. The outer ROI is needed to pre-process before inserting to the classifier to improve the performance of the classifier. Firstly, each region is processed according to the outer ROI specification. After that, the outer ROI is pre-processed to improve the recognition performance. After performing the pre-processing of the outer ROI, the image is ready to insert to the classifier with clear features and suitable input image size of 28*28 as shown in **Figure 3.5**.

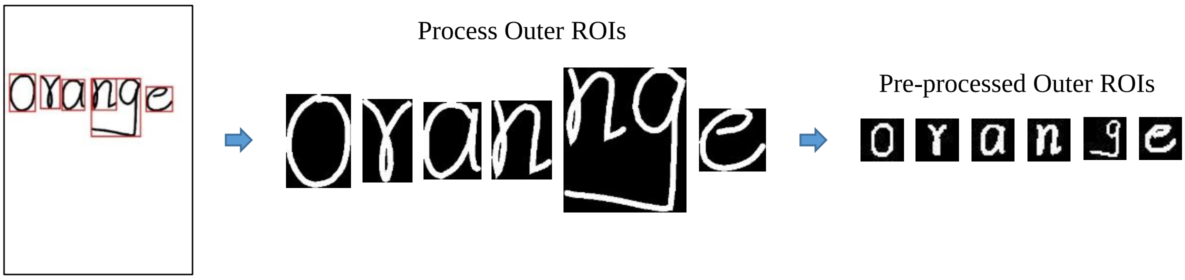


Figure 3.5 Outer ROI Pre-processing

3.5.4 Three Pre-processing Steps

There are three processes to execute as a pre-processing before classification of the segmentation results of handwritten input image.

- Noise Removal
- Thickening
- Normalization

(a) Noise Removal

Noise Removing is performed in both outer ROI segmentation and final classification of segmented characters, but pre-processing of inner ROI in sliding window segmentation does not include noise removal. There are some noise pieces in the region of the character to classify. Those unnecessary noises are small pieces of neighboring characters. Those noises interrupt the extraction of the features of the character and make the wrong recognition. To make the classifiable character for the best recognition, those unnecessary noises are needed to remove. To remove the small pieces of another character, all connected components of the image are labeled and bounded the region. After that, each connected component is classified to determine it is a part of the character for classification or noise based on the position and area of the labeled region. The biggest labeled region is fit with the whole image. That biggest region is the character region to classify and another region pieces are assumed noises. If the noise is enough small to be the dot of the character, ‘i’ or ‘j’, and upper of the main biggest region, the features of the main region is extracted. When the main region is narrow width and long enough to be main body of the ‘i’ or ‘j’, that noise is not removed. If not, the noise is removed to get the clear recognition. To remove the noises, all foreground pixels in the removable regions are filled with background pixel value to get the clear and classifiable character as shown in **Figure 3.11**.

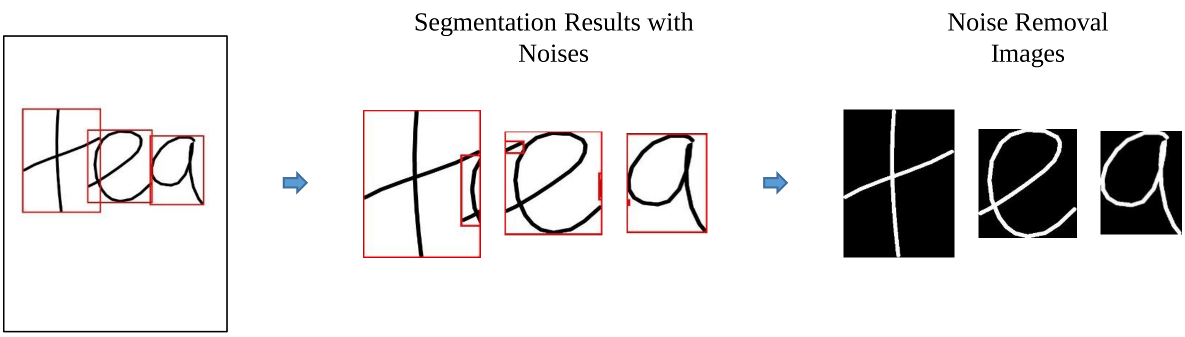


Figure 3.6 Noise Removal

(b) Thickening

In normalizing the character image, the image is needed to resize into the input size 28*28 of the CNN trained classifier. When resizing the image, the objects are very thin and sometime the very small objects are lost. The dilation of the segmented character image is implemented before normalization of the image to prevent the original features of the objects. The morphological thickening operation is used to perform this step. A comparison of recognition without thickening and recognition with thickening is shown in **Figure 3.12** with images. Normalization with thickening can be recognized the character, ‘f’, correctly whereas normalization without thickening produces the wrong recognition.

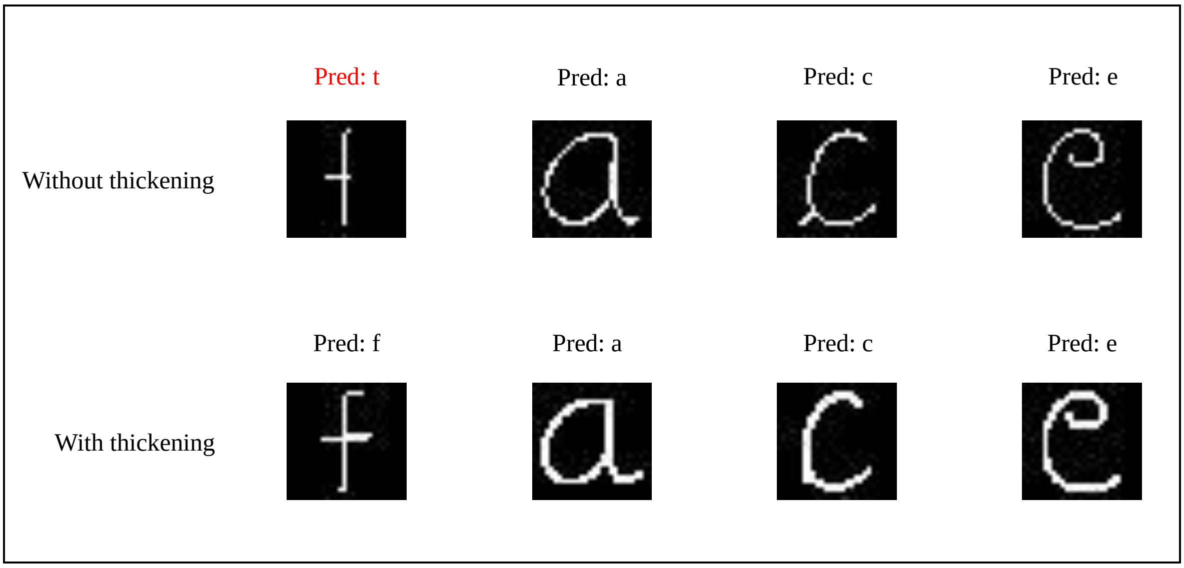


Figure 3.7 Comparison of without Thickening and with Thickening

(c) Normalization

In resizing the image, the shape of the character is destroyed because the resized image size is the square size of 28*28. To prevent this, initially, the thicken classifiable image is padded on the base of the row and column of that image to get the square image. If the width of the image is greater than the height, the image is padded in the top and bottom with half of the exceeded pixel number of the width over the height. If the height is exceeded than the width, the left and right of the image are padded with the half of the exceeded number of pixels. Then, the padded image is resized into the input image size of 28x28.

3.5.5 Classification of Outer ROI

The image obtaining after the pre-processing step is fixed to the trained convolutional neural network (CNN) classifier. The classifier determines the ROI whether it is a valid character or not based on the confidence threshold parameter and probability. The CNN calculates the confidence scores of each label for outer ROI. If the highest output confidence is greater than the confidence threshold parameter, the coordinates and dimensions of that valid outer ROI candidate are kept. If the highest confidence is less than the threshold parameter, the sliding window segmentation is applied to that outer ROI. Top five output probabilities of sample qualified outer ROI are shown in **Figure 3.6** and **Figure 3.7** shows the sample qualified and unqualified outer ROIs.

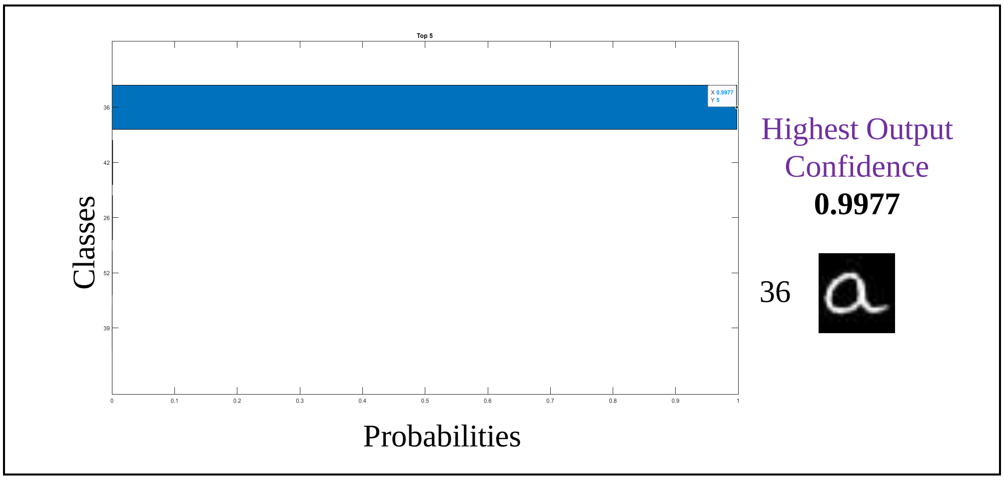


Figure 3.8 Top Five Output Probabilities

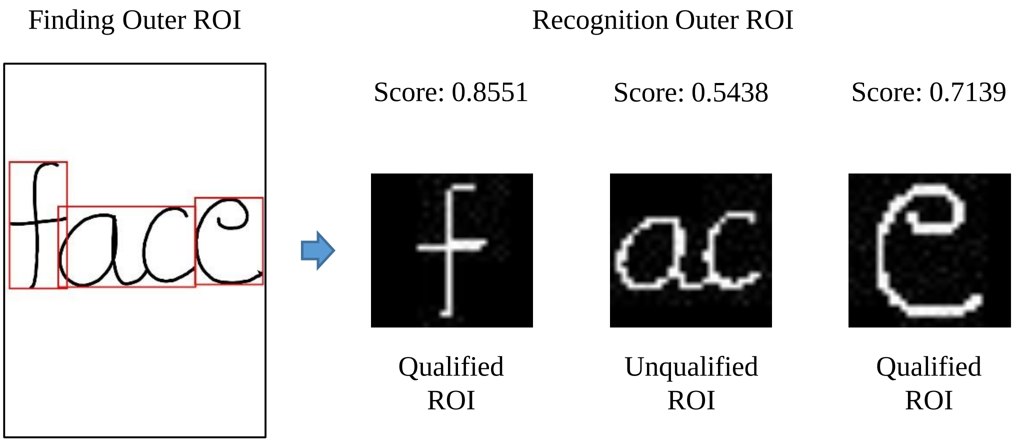


Figure 3.9 Extracting Unqualified Outer ROI

3.5.6 Sliding Window Processing

Unqualified outer ROI is split into multiple vertical slides with various widths by applying the sliding window processing. Initially, the window size is set to the size of the height of outer ROI and 20% (1/5) of the width of outer ROI to get the smallest window slide. That initial width is enough to cover the smallest width of the character, 'l'. Then, the sliding window is moved until reaching to the end of the outer ROI.

Moving distance of sliding window is set 20% (1/5) of the window width to get the overlap ratio of 4/5 with the previous slide. Before exceeding the bounding of outer ROI, the sliding window moving is stopped and repeat the process with the wider window width. To get the wider width of the window, 50% (1/2) of the initial width is added to the previous width to repeat the process until the window width is reached to the ROI width. After processing the sliding window, the various slides detecting everywhere of the outer ROI are got with various widths. When the window width is same as the outer ROI width, the sliding window processing is stopped and the qualified sliding windows are extracted. The processing of sliding window segmentation is shown in **Figure 3.8**.

3.5.7 Finding Inner ROI

Finding inner ROI is applied to fix all foreground objects in the sliding window. To define the inner ROI, uppermost, lowermost, leftmost and rightmost of the foreground pixels are specified in the window. After that, inner ROI is defined by applying the bounding rectangle on uppermost, lowermost, leftmost and rightmost of all foreground pixels found within the sliding window as shown in **Figure 3.9**. Finding inner ROI is intended to fit in normalization of inner ROI in the next step.

3.5.8 Inner ROI Pre-processing

The inner ROI can be classified by inserting into the trained CNN classifier. Before inserting to the classifier, the inner ROI is needed to pre-process into the input image size of the CNN classifier as shown in **Figure 3.10**. Thickening and normalization are performed in pre-processing of inner ROI. These two pre-processing steps are explained in section 3.6. Processing the inner ROI is performed before pre-processing the inner ROI into the classifiable image. After that, the inner ROI is thicken to and normalized into the input image size of 28x28. Finally, the normalized classifiable inner ROI image is inserted to the trained CNN classifier.

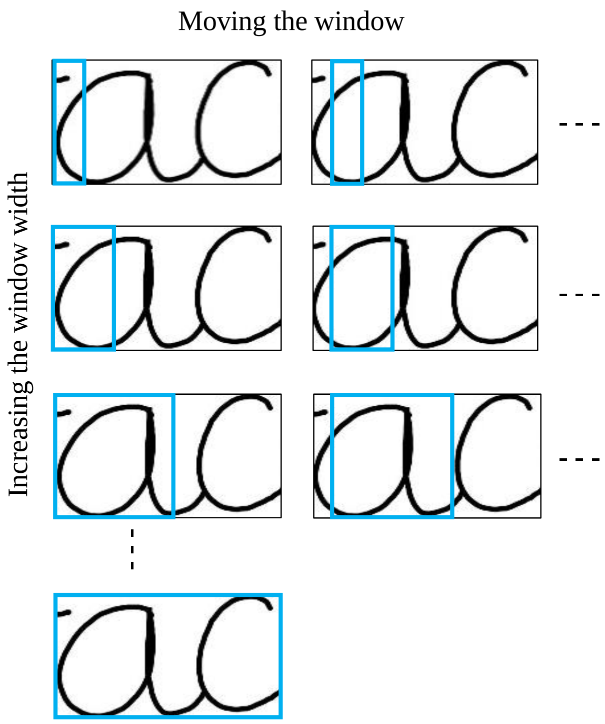


Figure 3.10 Sliding Window Processing

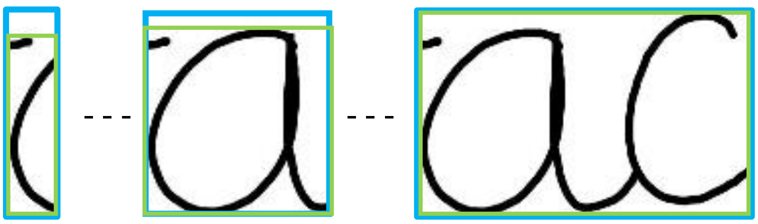


Figure 3.11 Finding Inner ROI

3.5.9 Classification of Inner ROI

The normalized inner ROI is classified by using trained CNN classifier. Based on the top performance accuracy of the classifier and confidence threshold parameter, the system determines whether the window is a valid character or not. If the highest output confidence is greater than the confidence threshold parameter, the overlap of the window over the previous valid character window is detected. If the overlap ratio of the one candidate over the other is greater than or equal to the overlap threshold parameter, the most possible window from these overlap candidates is qualified and the coordinates and dimensions of that inner ROI are kept as a candidate for representing a valid character. If overlap of one candidate over the other is smaller than overlap threshold parameter, both candidates are qualified. The confidence threshold parameter for outer ROI was determined 0.5 because the highest output confidence of the character, ‘m’, is between 0.5 and 0.6. The confidence of that character is the lowest in 62 characters – 10 digits, 26 lowercase letters and 26 uppercase letters. The confidence threshold for sliding window segmentation is 0.7 because the maximum likelihood slides are needed to extract from the various sizes of windows.

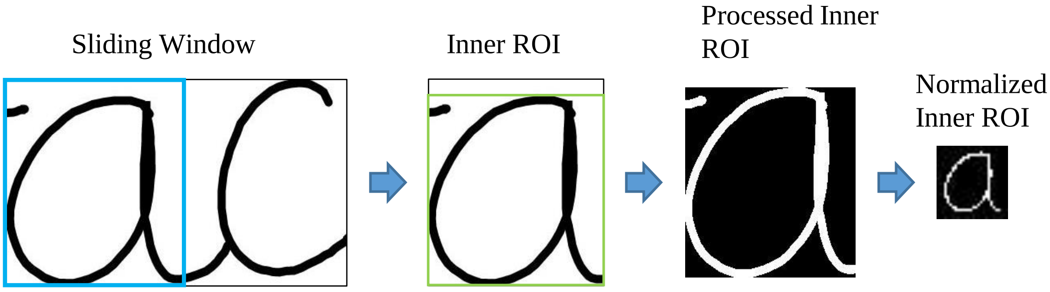


Figure 3.12 Pre-processing of Inner ROI

3.6 Classification

To recognize the segmented classifiable character, this system uses convolutional neural network (CNN) that is widely accepted classifier in image classification. Firstly, training and testing data are prepared and labeled using EMNIST (Extended Modified National Institute of Standards and Technology) database [4]. There are 62 categories in the dataset including capital letters: ‘A’ to ‘Z’, small letters, ‘a’ to ‘z’ and 10 numbers, ‘0’ to ‘9’. After that, the network is created and trained using prepared database. The trained classifier determines that what characters are written on the tablet with label. According to the recognized label, the system prints out the recognized machine printed character.

After pre-processing the segmented image, the classification of the segmented classifiable character image is executed to recognize the handwritten character of the input image. Some classified images and recognition are shown in **Figure 3.13**.

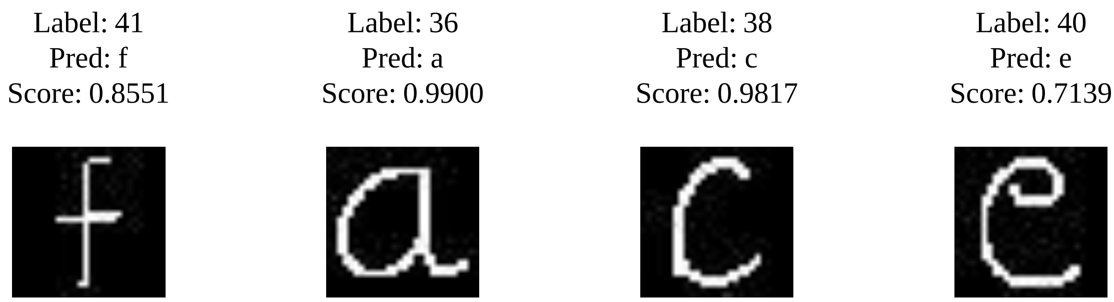


Figure 3.13 Sample Image Recognition

3.7 Convolutional Neural Network (CNN)

This section explains the architecture and model of the Convolutional Neural Network (CNN). Convolutional neural networks are deep artificial neural networks. It can be used to classify images, cluster them by similarity and perform object recognition within scenes. It can be used to identify faces, individuals, street signs, tumors, platypuses and many other aspects of visual data. The convolutional layer is the core building block of a CNN. The layer's parameters consist of a set of learnable filters (or kernels) which have a small receptive field but extend through the full depth of the input volume. During the forward pass, each filter is convolved across the width and height of the input volume, computing the dot product, and producing a 2-dimensional activation map of that filter. As a result, the network learns when they see some specific type of feature at some spatial position in the input. Then the activation maps are fed into a down sampling layer, and like convolutions, this method is applied one patch at a time. CNN has also fully connected layer that classifies output with one label per node.

Among all the operations of CNN, convolutional layers, pooling layers, and fully connected layers are the most important ones. Therefore, these layers are introduced before presenting our proposed model.

The convolutional layer is the first layer, which can extract features from the images. Because pixels are only related to the adjacent and close pixels, convolution allows us to preserve the relationship between different parts of an image. Convolution is filtering the image with a smaller pixel filter to decrease the size of the image without losing the relationship between pixels.

When constructing CNN, it is common to insert pooling layers after each convolution layer in order to reduce the spatial size of the features maps. Pooling layers also help with the overfitting problem. A pooling size is selected to reduce the amount of the parameters by selecting the maximum, average, or sum values inside these pixels.

A fully connected network is in any architecture where each parameter is linked to one another to determine the relation and effect of each parameter on the labels. Since convolution and pooling layers reduce time-space complexity, a fully connected network can be constructed in the end to classify the images.

3.7.1 CNN Architecture of Proposed Model

Now, it is time to explain about an overview look of the proposed convolutional neural network. It has similarity with other handwritten recognition architectures but has changed in a number of filters, neurons and activation functions for better performance [13, 20, 21]. It has seven layers.

Layer-1 consists of a convolutional layer with ReLU (Rectified Linear Unit) activation function. It is the first convolutional layer of our CNN architecture. This layer gets the pre-processed image as the input of size n*n=28*28. The convolution filter size (f*f) is 5*5; padding (p) is 0 (around all the sides of the image), stride (s) is 1 and the number of filters is 32.

Layer-2 is the max pooling layer. This layer gets the input of size 32@24*24 from the previous layer. The pooling size is 2*2; padding is 0 and stride is 2.

Layer-3 is the second convolutional layer with ReLU activation function. This layer gets the input of size 32@12*12 from the previous layer. The filter size is 5*5; padding is 0, the stride is 1, and the number of filters is 32.

Layer-4 is the second max pooling layer. This layer gets the input of size 32@8*8 from the previous layer. The pooling size is 2*2; padding is 0, and stride is 2.

Layer-5 is the third convolutional layer without ReLU activation function. This layer gets the input of size 32@4*4 from the previous layer. The filter size is 4*4; padding is 0, the stride is 1 and the number of filters is 64.

Layer-6 is the fully connected layer. This layer takes an input one-dimensional vector of size 64 and outputs a one-dimensional vector of size 256. It has ReLU activation function.

Layer-7 is the last layer of the network. It is also fully connected layer. This layer computes the class scores, resulting in a vector of size 62, where each of the 62 character corresponds to a class score, such as among the 62 categories of EMNIST dataset. It has softmax activation function for final outputs.

The layers of the CNN implemented in MATLAB, MatConvNet [5] and the architecture of the CNN is shown in **Figure 3.14**.

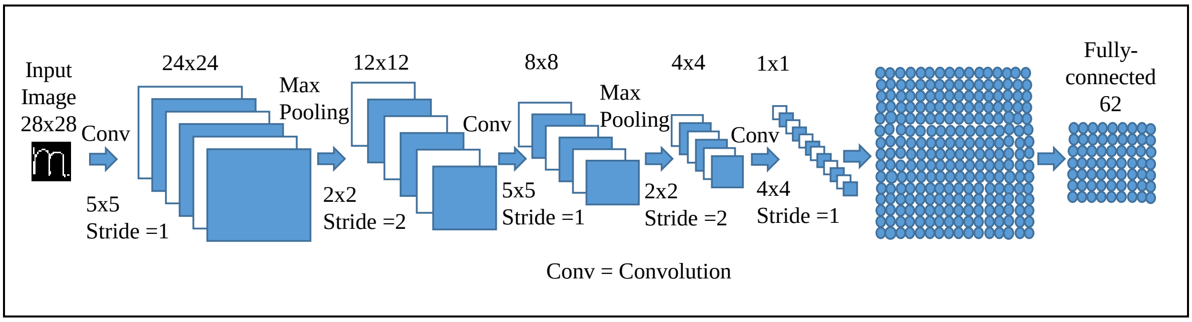


Figure 3.14 Architecture of Convolutional Neural Network (CNN)

3.8 Conclusion

A method for automatic recognition of the tablet based application input handwritten character is proposed in this chapter. The new recognition based handwritten character segmentation is implemented in this research. In future, a real-time handwritten character segmentation and recognition system of tablet based application will be developed.

CHAPTER 4

EXPERIMENTAL RESULTS

4.1. Experimental Results of Handwritten Character Segmentation System

In this system, handwriting of different people are collected on tablet to build our own dataset. There may be unexpected ligatures problems and noises in character images written on tablet PC. This system is not only to overcome the challenging of unconstraint handwritten character segmentation but also to recognize each segmented character correctly. This approach is to implement recognition based segmentation by combination of convolutional neural network (CNN), sliding window segmentation and finding ROI.

The own dataset is constructed by using the four tablet models: ASUS ZENPAD 10 SPECS, HUAWEI MediaPad T310, Gecoo Tablet A1, Galaxy Tab SM-T231. Four Tablets using are shown in **Figure 4.1**. This system aims for educational developments of primary school students who live in developing countries to help for their capabilities and at their independence. Even though every tablet type can be used for this system, the cheapest tablet models are used to collect the data to implement this.

Most of handwritten data images of the own dataset were collected from the laboratory members and some images were collected from Myanmar primary school students. The data are collected from primary school students of middle school, Hmaw Gan in Sinmezwe, Thaegone Township, Bago Region, Myanmar.

Some memorable photos collecting handwritten data of primary school students in Myanmar are shown in **Figure 4.2**. **Table 4.1** describes the detail specifications including model, OS, storage, memory, display, resolution, image type of tablets used for this system.

(a) ASUS ZENPAD 10 SPECS

(b) HUAWEI MediaPad T310

(c) Gecoo Tablet A1

(d) Galaxy Tab SM-t231

Figure 4.1 Four Tablet Models

Figure 4.2 Data Collecting Photos

Table 4.1 Four Tablet Models

Model	ASUS ZENPAD 10 SPECS	HUAWEI MediaPad T310	Gecoo Tablet A1	Galaxy Tab SM-T231
OS	Android 7.0	Android 7.0	Android 5.1.1	Android 4.4.2
Storage	16GB	16GB	8 GB	16 GB
Memory	2GB	2GB	2GB	1.5 GB
Display	10.1 inches	9.6 inches	8.0 inches	7.0 inches
Resolution	1280x800	1280x800	1920x1200	1280x720
Image Type	<.jpg>	<.jpg>	<.jpg>	<.jpg>

The dataset consists of 7 categories: Animals, Body Parts, Education, Fruits, Drinks, Numbers and Others as described in **Table 4.2**. There are total 1684 images comprising of 7 categories images. The categories were prepared based on the international curriculum for primary school students. **Figure 4.3** shows the some sample collected handwritten data images of our own dataset that include the collecting data in both Japan and Myanmar.

Binarization is performed as a pre-processing process. Before segmenting the characters of handwritten image, pre-processing is performed on the input handwritten image. Converting the initial image into binary image is performed in binarization process by calculating the global threshold based on the Otsu’s method. Some binary images are shown in **Figure 4.4**.

Table 4.2 Categorization of Own Dataset

Categories	Types	Count
Animals	ant	20
	bird	33
	cat	38
	cow	33
	dog	44
	fish	33
	fox	36
	zebra	30
Body Parts	ear	43
	eye	35
	face	34
	head	36
	mouth	29
Education	book	21
	classroom	32
	homework	33
	school	33
	student	33
	teacher	33
Fruits	apple	47
	avocado	35
	banana	44
	blueberries	21
	cherries	19
	coconut	25
	durian	27
	grapes	21
	guava	28
	lemon	39
	mango	35
	orange	33
Drinks	coffee	38
	juice	33
	milk	33
	tea	36
	water	32
Numbers	18	32
	24	32
	35	38
	60	31
	97	31
Others	bell	21
	birthday	21
	blue	20
	box	30
	candy	22
	girl	24
	good	23
	hill	21
	quarter	30
	queen	38
	vegetable	30
	yard	30
	zoo	35
Total	**54**	**1684**

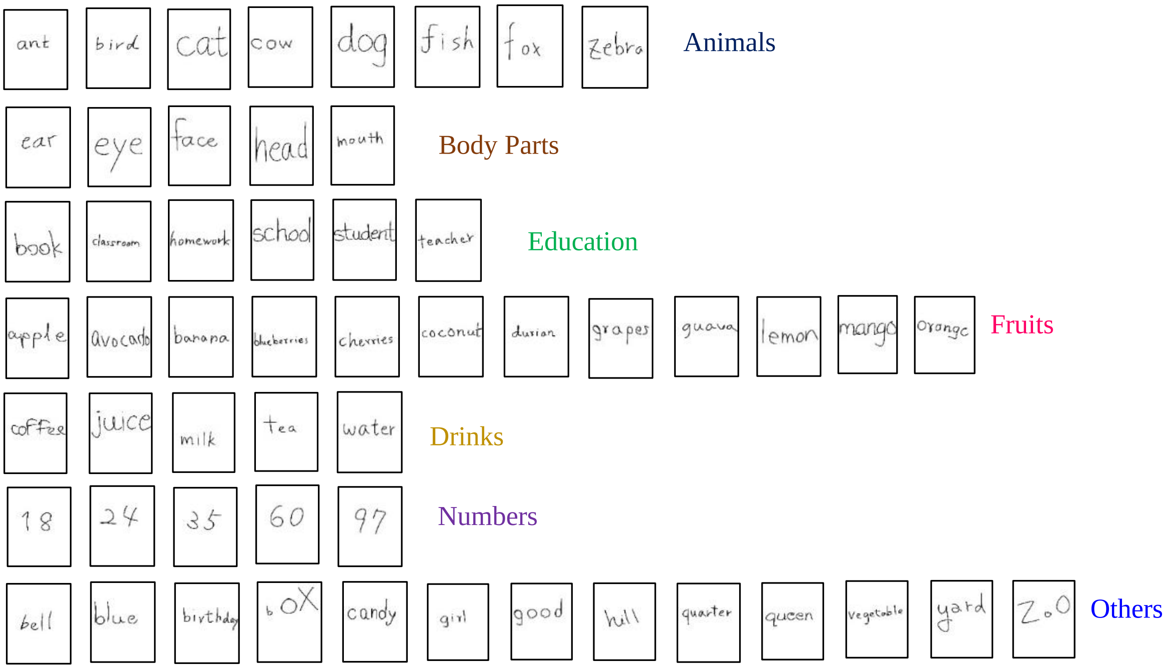


Figure 4.3 Sample Images of Each Categories in Our Dataset

Figure 4.4 Some Binary Images

4.1.1 Outer ROI Segmentation Result

Firstly, the ROI box is applied to detect the region of the all connected components by labelling based on the binary image. The images with ROI boxes of labelling are shown in **Figure 4.5**.

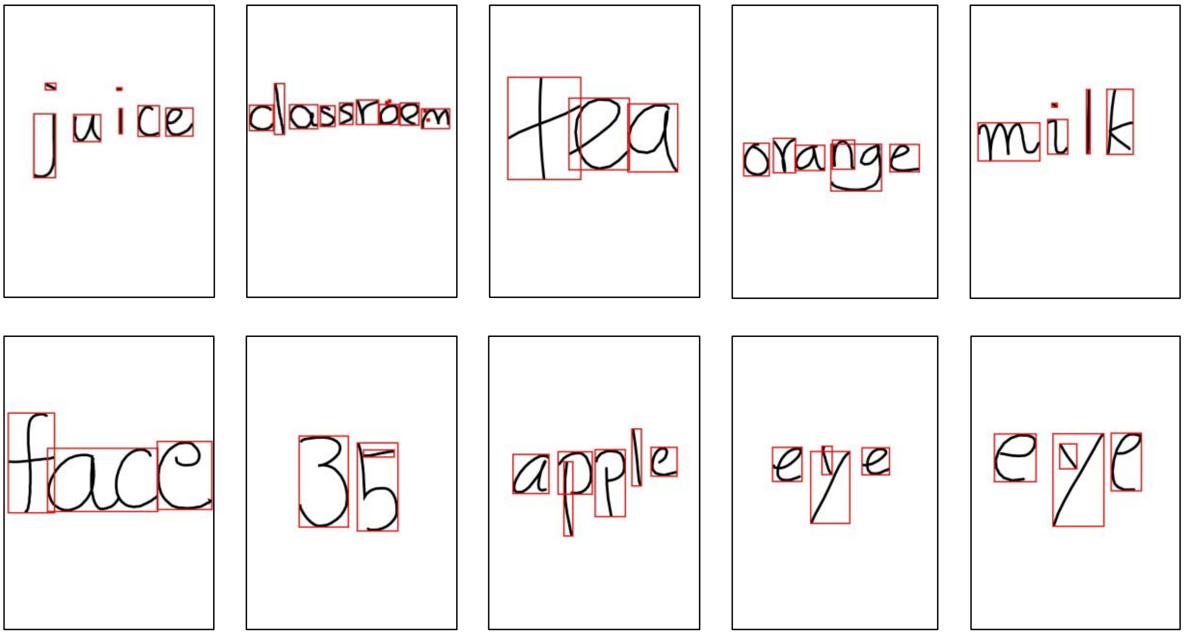


Figure 4.5 ROI Labelling and Bounding Boxes

Label combination is needed to combine two regions of characters, ‘i’ and ‘j’. The very small dot bounding boxes are extracted and the suitable ROI box is searched near it to combine for structuring the character ‘i’ or ‘j’. Some sample images including region combining are shown in **Figure 4.6.**

In some cases, some small dots are included in the image as noises. These noises are needed to remove to get a clear outer ROI. If there is no suitable ROI for forming the character ‘i’ or ‘j’, that small dot ROI box is removed as a noise. Example image with small ROI dot noises and image after removing those noises are shown in **Figure 4.7**.

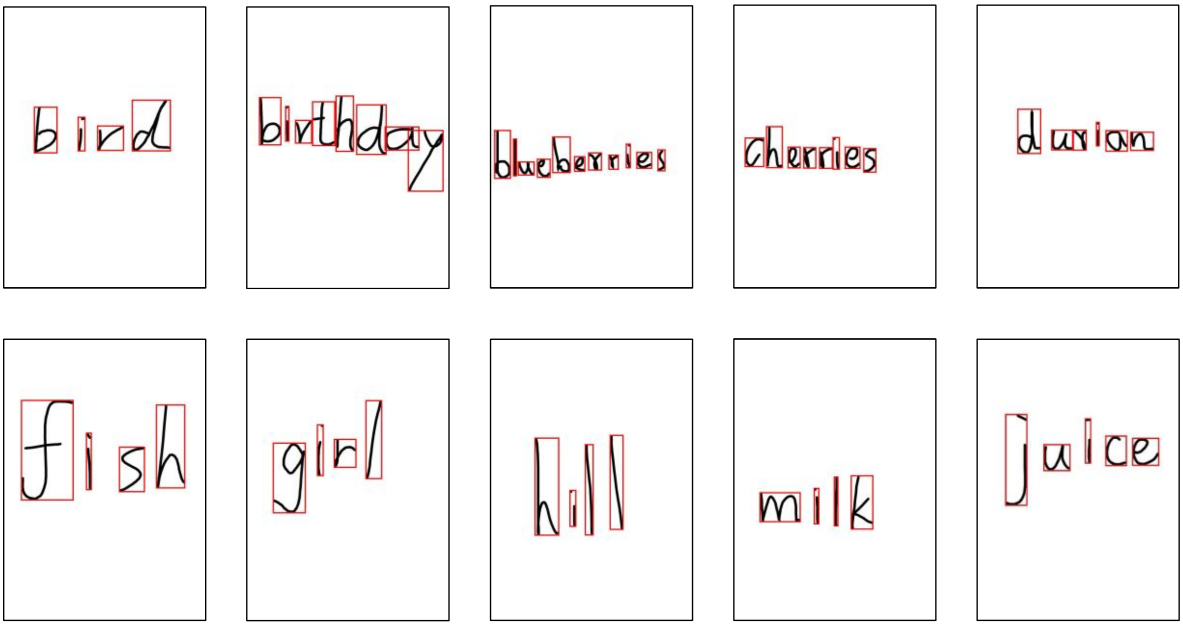


Figure 4.6 Some Region Combining Result

Finally, the clear outer ROI bounding boxes are got and those boxes are processed to extract into each outer ROI images to classify. Extracted outer ROI images are shown in **Figure 4.8**. Before classifying the outer ROI, three processes: noise removal, thickening and normalization are applied on that outer ROI as the pre-processing of classification. Pre-processing results of outer ROI images are shown in **Figure 4.9**.

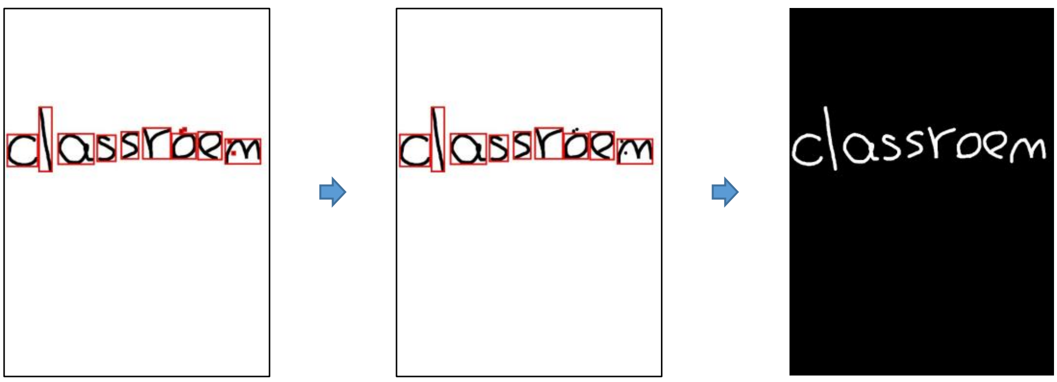


Figure 4.7 Small Dot ROI Noises Removal

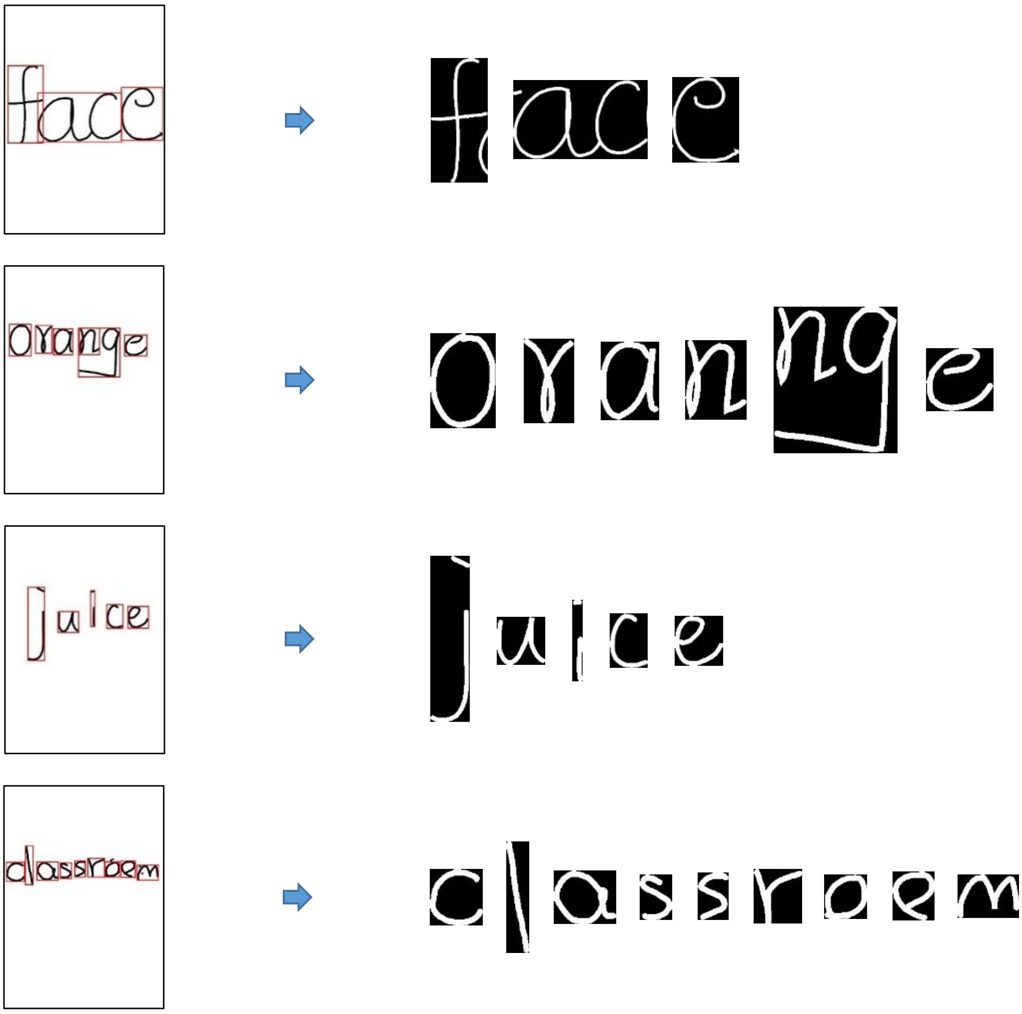


Figure 4.8 Outer ROI Processing

After pre-processing the outer ROI, trained CNN classifier classifies to extract the qualified character ROI based on the confidence threshold parameter. Outer ROI that the confidence of it is greater than the threshold parameter is qualified character ROI and coordinates and dimensions of it is noted and kept. Outer ROI that is less than the confidence threshold is segmented into vertical slides by applying the sliding window processing. Sliding window processing slides the unqualified outer ROI into various vertical portions with various widths. Qualified outer ROI classification and unqualified outer ROI classification are shown in **Figure 4.10**.

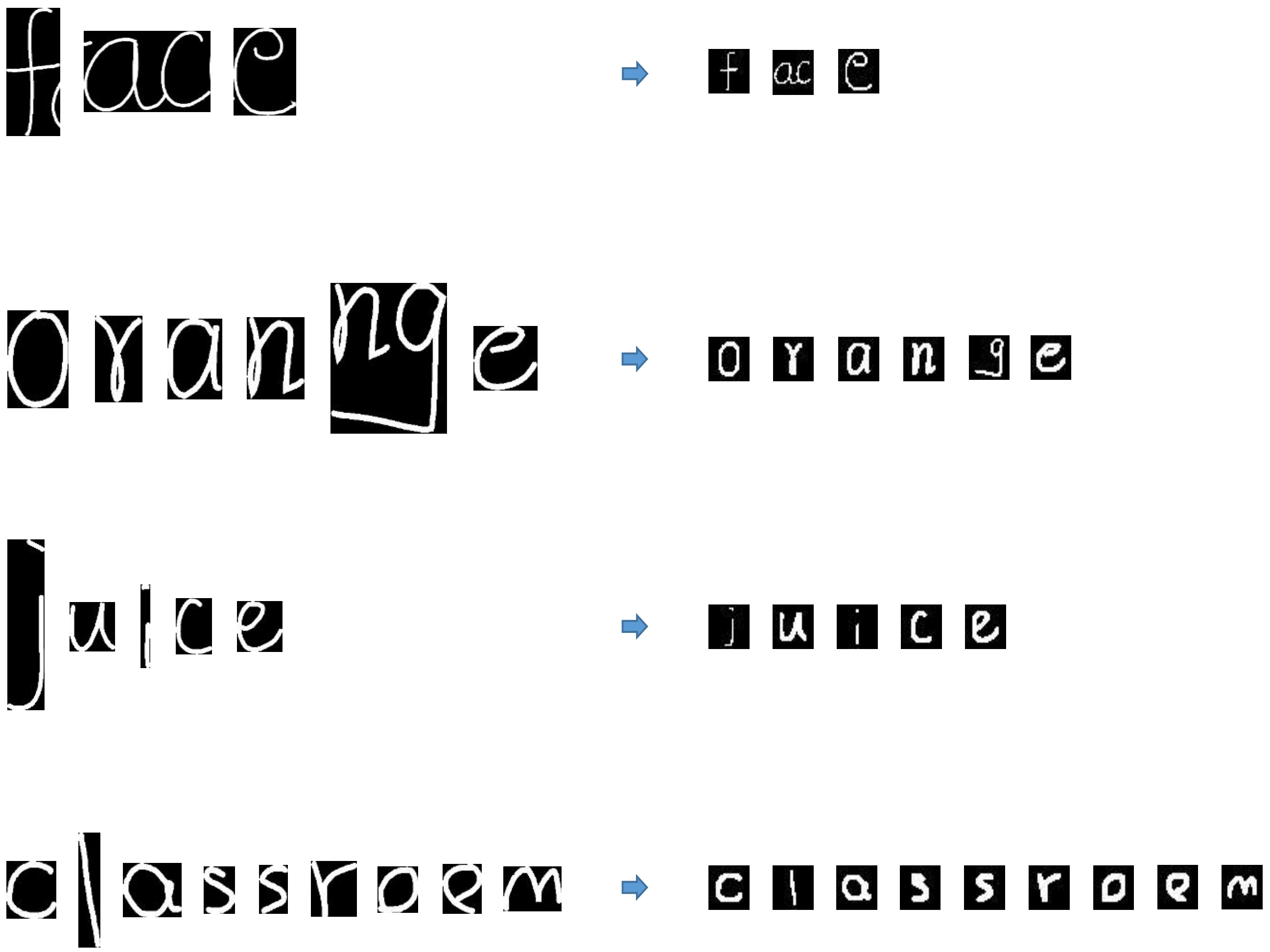


Figure 4.9 Pre-processing Result of Outer ROIs

In this system, slant correction is performed after outer ROI processing of segmentation section. This slant correction corrects individual slanted outer ROI into classifiable character position. When the outer ROI image is slanted to the right or left, it is needed to correct the slanting into the normal position. Correction of slanting can reduce the bad segmentation and segmentation noises. In slant correction, the horizontal shear transformation is applied by defining the shear factor based on the slanted angle of vertical lines, called shear angle. Slanted Image and slant corrected images are shown in **Figure 4.11**.

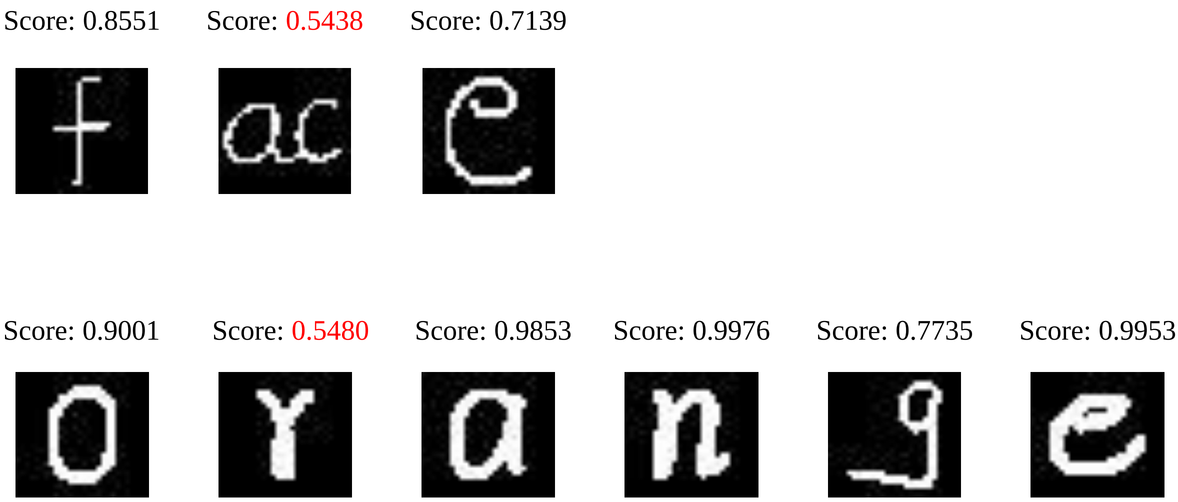


Figure 4.10 Classification of Outer ROI

When correcting the slant after outer ROI segmentation step in segmentation stage, individual slanted character can be corrected definitely.

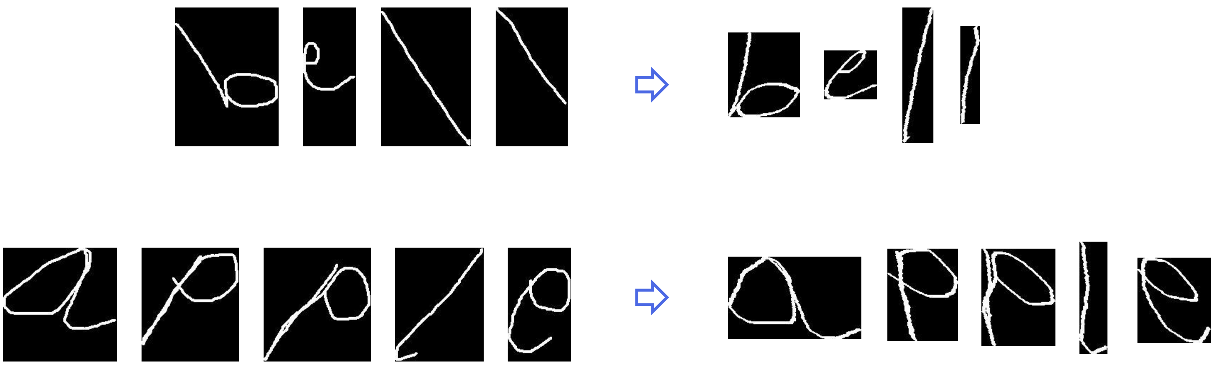


Figure 4.11 Slanted ROIs and Slant Corrected ROIs

If slant correction step is processed in pre-processing stage after binaraization, it is needed to correct the whole word as shown in **Figure 4.12**. In the first implementation of slant correction after binarization in pre-processing, it is needed to be slanted all characters of the whole word because it will slant the whole word before segmenting.

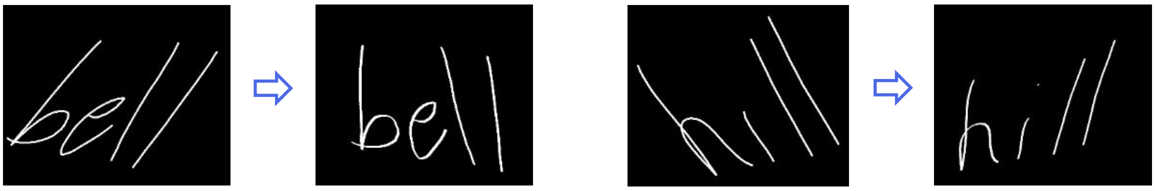


Figure 4.12 Slanted Images and Slant Corrected Images

4.1.2 Sliding Window Segmentation Result

Sliding window processing split the unqualified outer ROI into vertical slides of various sizes to find the valid character. Each slide is fed into the trained classifier to extract the valid character slide. Sliding window processing and qualified windows are shown in **Figure 4.13**.

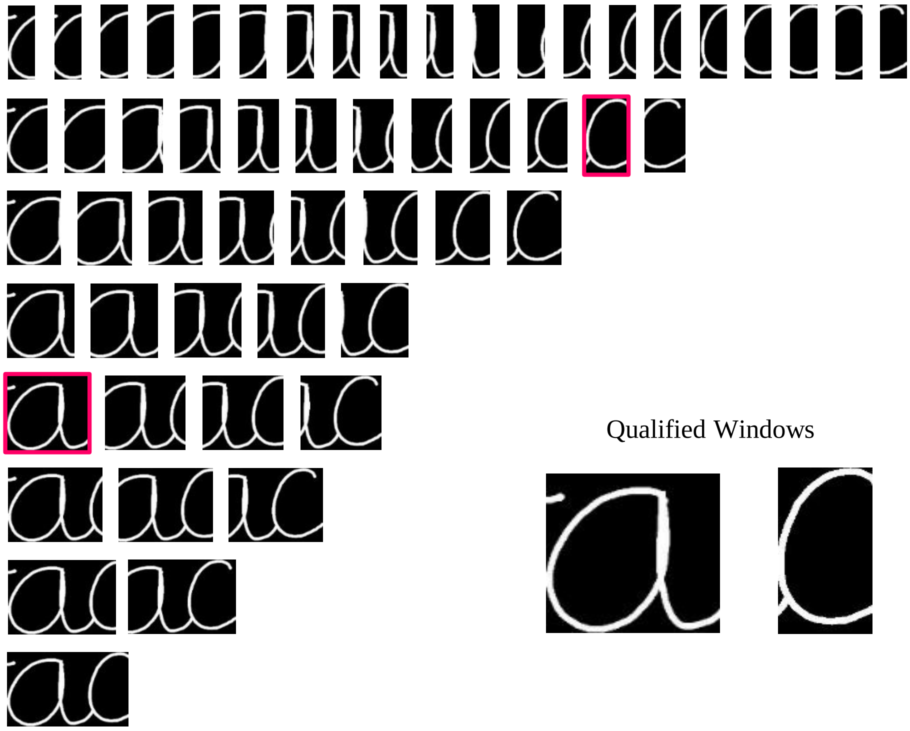


Figure 4.13 Sliding Window Processing

Segmentation results of outer ROI segmentation, sliding window segmentation processes and final segmentation are shown in **Figure 4.14**.

(a) Outer ROI segmentation Result **(b) Sliding Window Segmentation Result**

(c) Final Segmentation

Figure 4.14 Segmentation Results

4.2. Experimental Results of Handwritten Character Recognition System

The CNN trained classifier recognizes the segmentation results of both outer ROI segmentation and sliding window segmentation processes.

4.2.1 Pre-processing Before Classification

Before recognition, the segmented character is pre-processed to get the classifiable input image for CNN classifier. There are three processes: noise removal, thickening and normalization in pre-processing for final classification and recognition as in the pre-processing of outer ROI for outer ROI segmentation. In sliding window segmentation, there are only two pre-processing steps: thickening and normalization. The sample results of three pre-processing steps are shown in **Figure 4.15**.

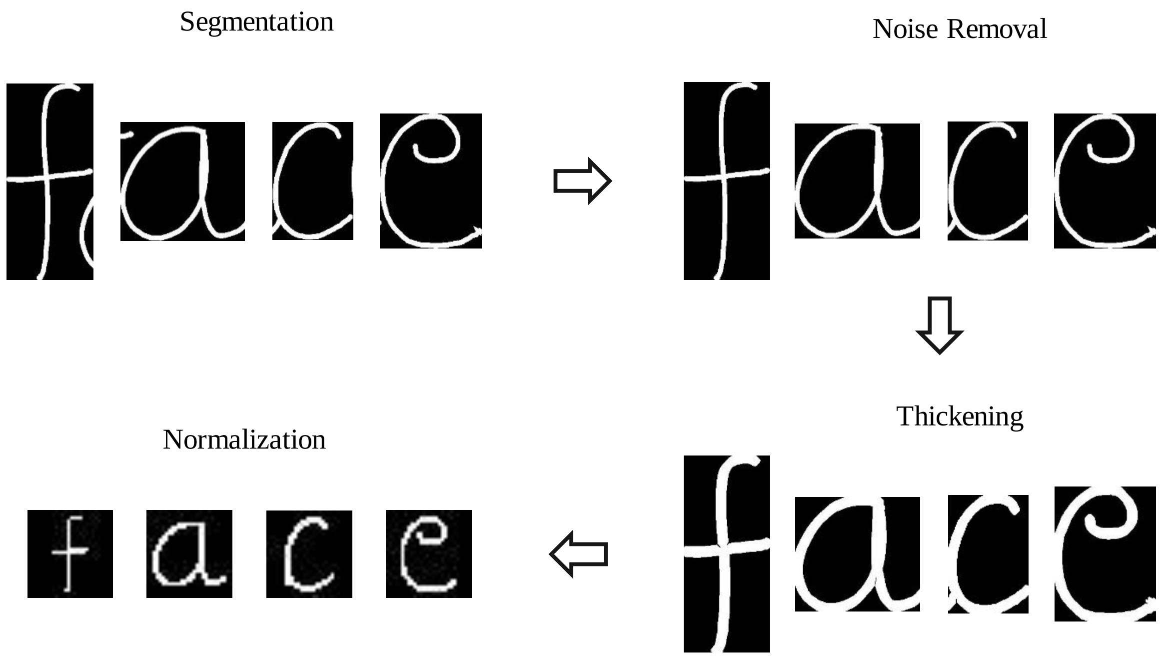


Figure 4.15 Results of Pre-processing Steps

4.2.2 Recognition Result

The system classified and recognized the classifiable pre-processed images of segmented results and printed out the machine printed characters of recognition results. Some images that include wrong recognized character are shown in **Figure 4.16** and sample correct recognition results are shown in **Figure 4.17**.

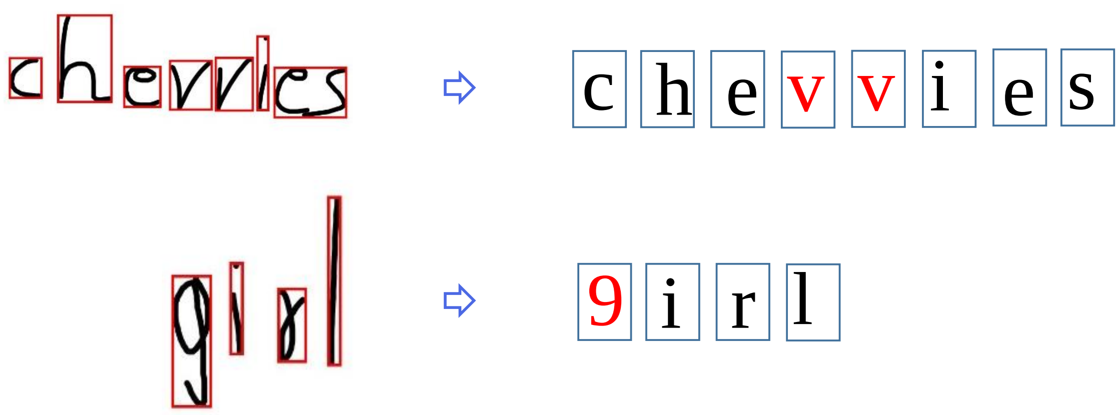


Figure 4.16 Wrong Recognition of Some Segmented Characters

The segmentation and recognition results of the images that including the split characters, ‘5’ and ‘y’ are shown in **Figure 4.18**. Characters, ‘5’, ‘y’, ‘b’, ‘p’, and ‘y’ are split into two ROIs. The small ROI from the split ROIs is not the character because their dimensions are too small to become one character by comparing with other ROIs. They are not the small parts of the character ‘i’ or ‘j’ and also not small enough to remove in small dot removing step. The noises are completely arrived during the main region boundaries (i.e. noises are not touched with the boundary of the main region) because they are not the parts of the another character. So, those noises are not removed in the noise removal before classification of the split character. In **Figure 4.19**, character ‘w’ is split into two ‘v’ characters and character ‘v’ is into two ‘l’. They are completely split and can be recognized as ‘v’ and ‘l’ respectively.

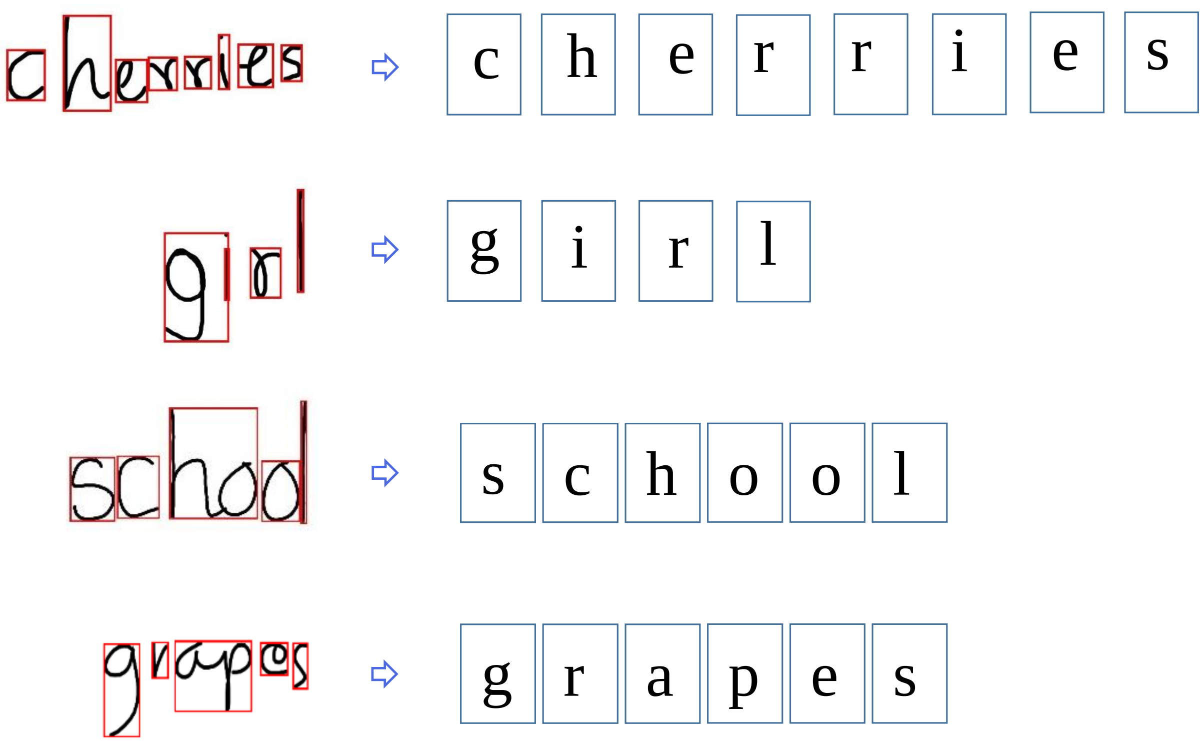


Figure 4.17 Some Correct Recognition Results

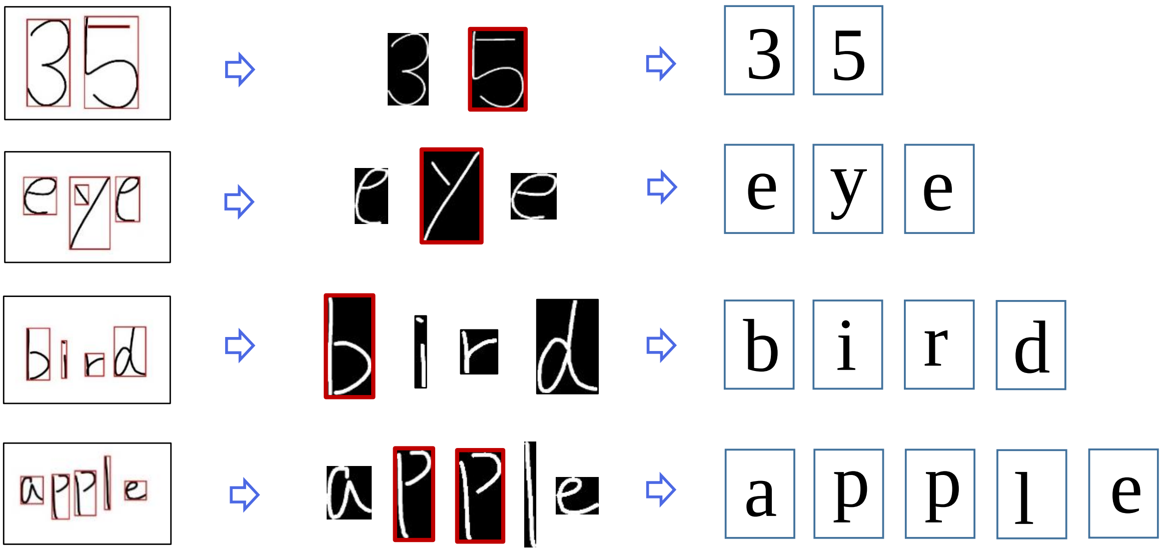


Figure 4.18 Successful Segmentation of Split Characters

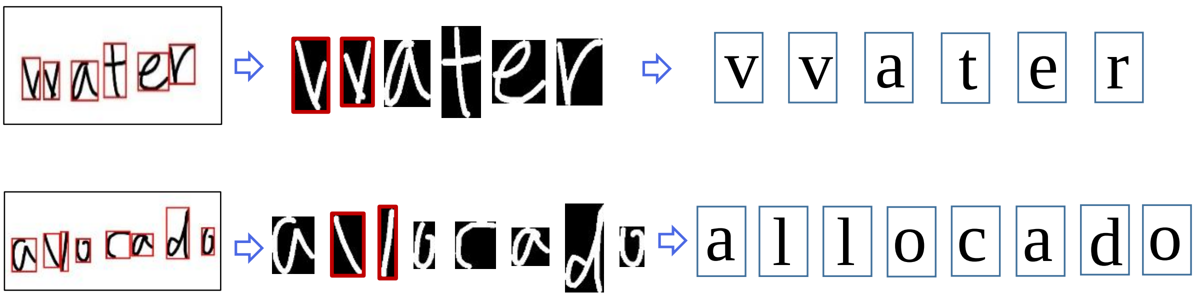


Figure 4.19 Splitting into Two Characters

4.2.3 Convolutional Neural Network (CNN) Architecture

There are seven layers in the architecture of the CNN. Layer 1 is convolutional layer and ReLU (Rectified Linear Unit). Layer 2 is max pooling layer. Layer 3 is the second convolutional layer with ReLU. Layer 4 is the second max pooling layer. Layer 5 is the third convolutional layer and layer 6 and layer 7 are the fully connected layer. The structure of those seven layers are described in **Table 4.3**.

Table 4.3 Layer Structure of the CNN

Layer No.	Layers	Input Size	Filter Size	Stride	Output
Layer 1	Conv	28x28	5x5	1	32@ 24x24
Layer 2	Max Pooling	24x24	2x2	2	32@ 12x12
Layer 3	Conv	12x12	5x5	1	32@ 8x8
Layer 4	Max Pooling	8x8	2x2	2	32@ 4x4
Layer 5	Conv	4x4	4x4	1	64@ 1x1
Layer 6	FC	1x1	-	-	256@ 1x1
Layer 7	FC	1x1	-	-	62 classes

There is another CNN structure constructed aiming for this work even though that is not used in this system. The structure of that network is also described in **Table 4.4.** That unused CNN is constructed with 4 convolutional layers, 2 max pooling layers, and 1 fully connected layer unlike the structure using in this system. Filter sizes are also different between them.

Table 4.4 Layer Structure of the Unused CNN

Layer No.	Layers	Input Size	Filter Size	Stride	Output
Layer 1	Conv	28x28	3x3	1	8@ 26x26
Layer 2	Max Pooling	26x26	2x2	2	8@ 13x13
Layer 3	Conv	13x13	3x3	1	16@ 11x11
Layer 4	Max Pooling	11x11	2x2	2	16@ 5x5
Layer 5	Conv	5x5	3x3	1	64@ 3x3
Layer 6	Conv	1x1	3x3	1	64@ 1x1
Layer 7	FC	1x1	-	-	62 classes

4.2.4 Training Process

The training and testing data are prepared and labeled using the EMNIST dataset. There are 62 classes in the training and testing dataset. 26 capital letters, A-Z, 26 small letters a-z, and 10 numbers, 0-9 are including in the dataset. The sample data images of EMNIST dataset are shown in **Figure 4.20**.

The training progress of the convolutional neural network is shown in **Figure 4.21.** The validation accuracy of the training is 86.04%. The elapsed time is 3628 minutes and 27 seconds. It is taken about 2 days and half to train the CNN.

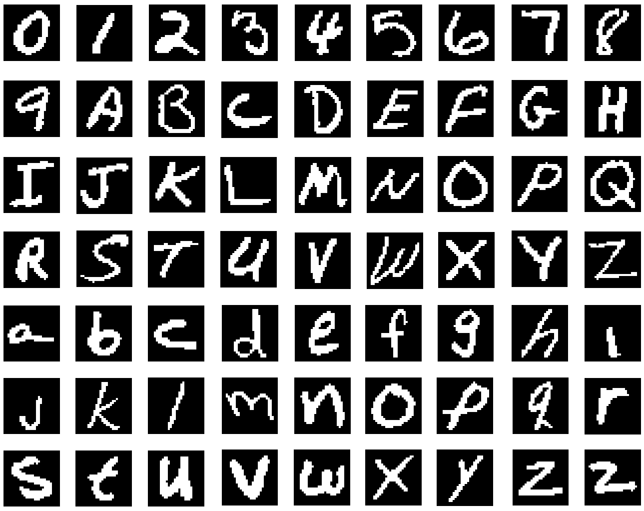


Figure 4.20 Sample Data of Training and Testing Dataset

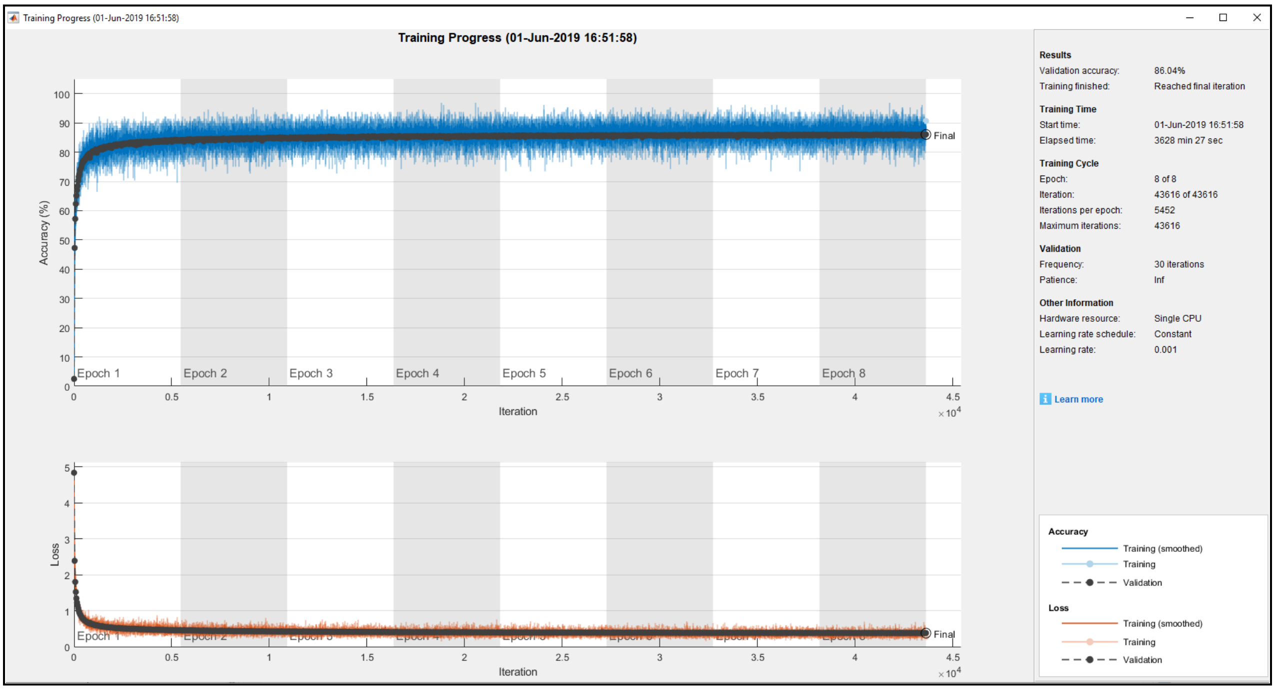


Figure 4.21 Training Progress

The training progress of unused CNN is also shown in **Figure 4.22**. It took 6568 minutes and 34 seconds in training progress, and validation accuracy is 81.78%

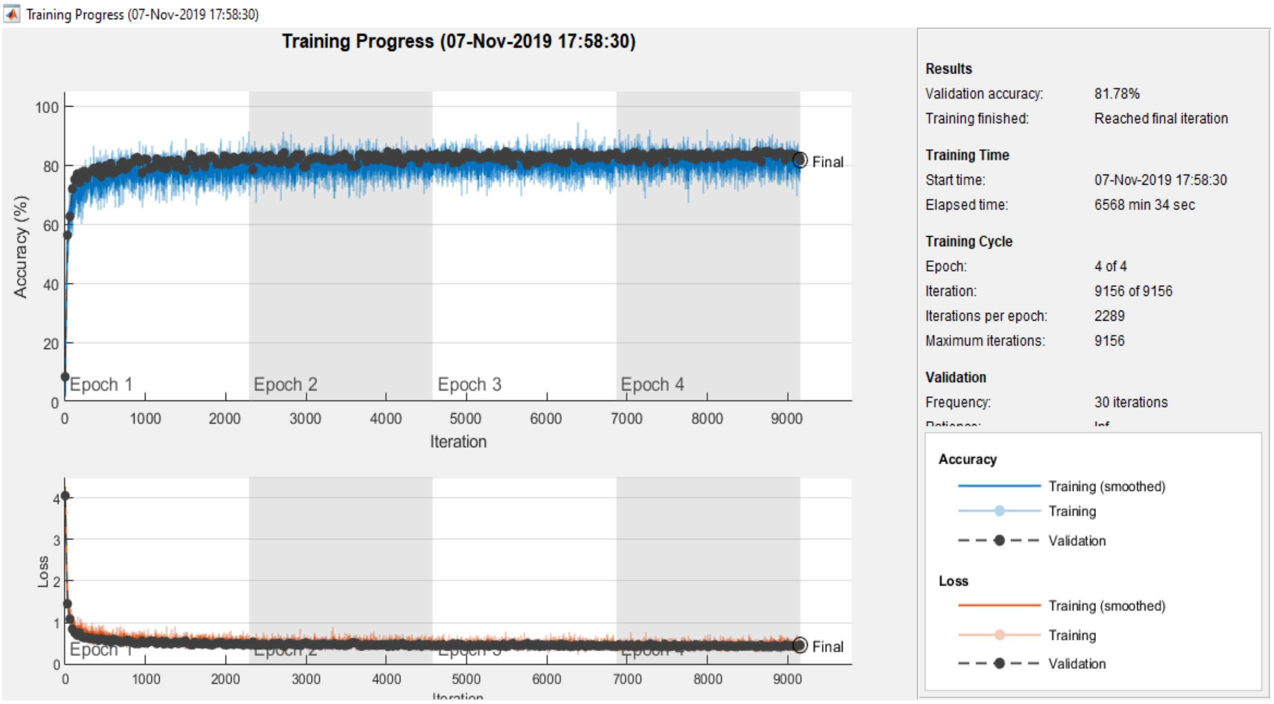


Figure 4.22 Training Progress of unused CNN

4.3 Performance Evaluation

The system correctly segmented 1507 images when testing the total handwritten images of 1684 and the performance accuracy of segmentation is 89.49%. According to experimental results, the proposed system of recognition based segmentation performed effectively.

The performance of recognition is 83.78%. When testing 6997 segmented character images, 5862 character images can be correctly recognized. The system miss-classified the characters with the similar contours (e.g. u and v). But, the recognition accuracy achieved the reliable performance. **Table 4.5** describes the recognition accuracy of EMNIST dataset and own collected dataset. The accuracy of own dataset is calculated and described by separating into two: the data collected from laboratory members and from primary school student.

Table 4.5 Recognition Accuracy of Two Datasets

Data	Recognition Accuracy (%)
EMNIST	86.04
Data from Lab members	83.63
Data from primary school student	87.45

This proposed system performance of segmentation on own collected dataset is described in **Table 4.6**. Segmentation accuracy of laboratory members' handwriting and Myanmar primary school students' handwriting is 90.19% and 76.54%. This system can segment successfully if characters are not linked to each other.

Table 4.6 Segmentation Accuracy

Data	Total Words Images	Successfully Segmented Images	Segmentation Accuracy (%)
Data from Lab members	1603	1445	90.19
Data from primary school student	81	62	76.54

On the other hand, recognition accuracy is 83.63% and 87.45% respectively as described in **Table 4.7**. This system gave reasonable segmentation and recognition accuracy results.

Table 4.7 Recognition Accuracy

Data	Total Words Images	Successfully Segmented Images	Recognition Accuracy (%)
Data from Lab members	6734	5632	83.63
Data from primary school student	263	230	87.45

CHAPTER 5

CONCLUSION AND FURTHER EXTENSION

There are many applications of handwriting for portable touch screen tablet to write down the important things in relevant industry in our real life purposes. Recognition of handwritten characters automatically on tablet like human's brain is also necessary to be more convenient in real life. Precisely, it can also be used in kindergarten education for learning in English, and many other related works.

Handwritten segmentation and recognition are subjects of much attention due to the presence of many difficulties. English language is used by a much higher percentage of the world's population. People will be benefited all over the globe if an automation system is designed for off-line handwriting recognition. The off-line handwriting recognition system enables the automatic reading and processing of a large amount of data printed or handwritten in English language. Although such automated systems for recognizing off-line handwriting have already existed, the scope of further improvement is always there.

Character segmentation is the most crucial step for any OCR (Optical Character Recognition) system. The selection of segmentation algorithm being used is the key factor in deciding the accuracy of OCR system. If there is a good segmentation of characters, the recognition accuracy will also be high. Segmentation of words into characters becomes very difficult due to the cursive and unconstrained nature of the handwritten script.

There are three approaches of segmentations - explicit, implicit, and holistic segmentation. In the explicit segmentation, the input word image is segmented into sub images of individual characters, which are then classified. Vertical segmentation is one of the approaches of explicit segmentation. Implicit segmentation is also called recognition based segmentation. Holistic is segmentation free. This approach is to implement implicit segmentation using sliding windows.

First of all, pre-processing operations are performed on the tablet based application input handwritten image. In the second, the connected elements are labelled in the pre-processed binary image. Bounding rectangle boxes are defined on the pre-processed image based on the labelling to define ROI boxes. After that, the segmented classifiable character is recognized through the trained classifier. The classifier determines the labels with class scores of the segmented character. Thus, the characters presenting in the images can be classified as: class 0 to 61.

Finally, the system output the machine printed characters from handwritten image of tablet-based application. This system recognized the segmented characters after both outer ROI segmentation and sliding window segmentation. The recognition can be finished at the same time with the segmentation process but the final segmentation after all segmentation steps improved the performance accuracy of recognition rate because noise removal is not included in the pre-processing of classification in sliding window segmentation. There are three pre-processing stages: noise removal, thickening and normalization before final recognition process as in the outer segmentation steps.

In this proposed system, the segmentation process is focused because segmentation errors led to the recognition errors. In the real world of handwritten recognition environment, there are the various challenging problems such as:

- Variation in shapes and writing styles of different writers
- Cursive nature of handwriting i.e. two or more characters in a word can be connected to each other while writing
- Some characters in handwriting can have similar contours (e.g. ‘r’ and ‘v’)
- Some characters can give the illusion of presence of two similar characters (e.g. ‘w’ can be segmented into two ‘v’.

If characters are not touched to each other in the input image, this system can successfully segment each character from that image. The limitations of this system are that the system can recognize only one and two digits numbers (e.g. 9,18), and if both capital letter and small letter have the similar features (e.g. ‘*W*’ and ‘*w*’, ‘*V*’ and ‘*v*’), those similar characters will be recognized into small characters (e.g. ‘*w*’, ‘*v*’). The experiments accomplish the outstanding results of segmentation although this system still need to achieve the goal. This study focuses on segmentation process for better segmentation accuracy of handwritten character recognition.

For performing the experiments, the own dataset collected from Japanese and Myanmar students by using handwritten on tablets are used. The dataset can be classified into seven categories:

- **Animals** - ant, bird, cat, cow, dog, fish, fox, zebra
- **Body Parts** - ear, eye, face, head, mouth
- **Education** - book, classroom, homework, school, student, teacher
- **Fruits** - apple, avocado, banana, blueberries, cherries, coconut, durian, grapes, guava, lemon, mango, orange
- **Drinks** - coffee, juice, milk, tea, water
- **Numbers** - 18, 24, 35, 60, 97
- **Others** - bell, birthday, blue, box, candy, girl, good, hill, quarter, queen, vegetable, yard, zoo.

In the future, more experiments on more cursive handwritten data will be executed by emphasizing on recognition the whole word to implement the real time application.

REFERENCES

[1] H. A. Al Hamad, R. A. Zitar, "Development of an Efficient Neural-based Segmentation Technique for Arabic Handwriting Recognition,Pattern Recognition", Vol. 43, No, 8, pp. 2773-2798, August 2010.

[2] T. Saba, A. Rehman, G. Sulong, "Cursive Script Segmentation with Neural Confidence", *International Journal of Innovative Computing, Information and Control*, vol. 7, no 8, pp. 4955-4964, 2011.

[3] L. C. Lin, V. E. Nzai, "Using iPad Apps to Enhance Literacy Skills of English Language Learners with Special Needs," *International Journal of Languages and Literatures, USA*, vol. 2, no. 1, pp. 21-29, March 2014.

[4] G. Cohen, S. Afshar, J. Tapson, A. Van Schaik, "EMNIST: Extending MNIST to Handwritten Letters," *2017 International Joint Conference on Neural Networks (IJCNN)*, Anchorage, AK, pp. 2921-2926, 2017.

[5] A. Vedaldi, K. Lenc, "MatConvNet: Convolutional Neural Networks for MATLAB," *Proceedings of the 23rd ACM International Conference on Multimedia,* pp. 689-692, Brisbane, Australia, Oct. 26-30, 2015.

[6] F. Kurniawan, et al., "Region-based Touched Character Segmentation in Handwritten Words", *International Journal of Innovative Computing Information and control*, Vol. 7, No. 6, pp. 3107-3120, June 2011.

[7] A. Choudhary, R. Rishi, S. Ahlawat, "A New Character Segmentation Approach for Off-Line Cursive Handwritten Words", *International Conference on Information Technology and Quantitative Management*, vol. 17, pp. 88-95, 2013.

[8] Y. H. Tay, P. M. Lallican, M. Khalid, C. Viard-Gaudin, S. Kneer, "An Offline Cursive Handwritten Word Recognition System", *Proceedings of IEEE Region 10 International Conference Electrical and Electronic Technology, TENCON 2001*, Singapore, vol. 2, pp. 519–524, 2001.

[9] V. Alexandrov, "Using Critical Points in Contours for Segmentation of Touching Characters," *Proceedings of the 5th International Conference on Computer Systems and Technologies,* 2004.

[10] A. Rehman, T. Saba, "Performance Analysis of Character Segmentation Approach for Cursive Script Recognition on Benchmark Database," Digital Signal Processing, vol. 21, no. 3, pp. 486-490, 2011.

[11] T. Saba, A. Rehman, G. Sulong, "Cursive Script Segmentation with Neural Confidence," *International Journal of Innovative Computing, Information and Control*, vol 7, no. 8, 2011.

[12] P. D. Gader, M. P. Whalen, M. J. Ganzberger, D. Hepp, "Handprinted Word Recognition on a NIST Data Set," *Machine Vision and Its Applications,* vol. 8, pp. 31-40, Jan. 1995.

[13] N. Jain, K. Rahul, I. Khamaru, A. K. Jha, A. Ghosh, "Hand Written Digit Recognition using Convolutional Neural Network (CNN)," *International Journal of Innovations & Advancement in Computer Science, IJIACS*, vol. 6, no. 5, May 2017.

[14] S. B. Hallale, G. D. Salunke, "Twelve Directional Feature Extraction for Handwritten English Character Recognition", *International Journal of Recent Technology and Engineering (IJRTE)*, vol. 2, no. 2, May 2013.

[15] A. Choudhary, R. Rishi, S. Ahlawat, "Off-Line Handwritten Character Recognition Using Features Extracted from Binarization Technique", *AASRI Conference on Intelligent Systems and Control,* vol. 4, pp. 306-312, 2013.

[16] J. Pradeepa, E. Srinivasana, S. Himavathib, "Neural Network Based Recognition System Integrating Feature Extraction and Classification for English Handwritten", *International Journal of Engineering,* vol. 25, no. 2, pp. 99-106, May 2012.

[17] R. M. O. Cruz, G. D. C. Cavalcanti, T. I. Ren, "An Ensemble classifier for Offline Cursive Character Recognition Using Multiple Features Extraction Technique", *The 2010 International Joint Conference on*

Neural Networks (IJCNN), Barcelona, pp. 1-8, 2010.

[18] A. Graves, M. Liwicki, S. Fernandez, R. Bertolami, H. Bunke, J. Schmidhuber, “A Novel Connectionist System for Unconstrained Handwriting Recognition,” IEEE *Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 5, pp. 855–868, May 2009.

[19] Myat Thiri Wai, Thi Thi Zin, Mitsuhiro Yokota, Khin Than Mya, “Recognition Based Segmentation of Handwritten Alphanumeric Characters Entry on Tablet PC”, *Memoirs of the Faculty of Engineering, University of Miyazaki*, to be published in vol. 49, Sept. 2020.

[20] H. A. Alwzwazy, H. M. Albehadili, Y. S. Alwan, N. E. Islam, "Handwritten Digit Recognition using Convolutional Neural Networks", *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 4, no. 2, pp. 1101-1106, Feb. 2016.

[21] Y. Lecun, L. Bottou, Y. Bengio, P. Haffner, “Gradient-Based Learning Applied to Document Recognition”, *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

PUBLICATION

[1] Myat Thiri Wai, Thi Thi Zin, Mitsuhiro Yokota, Khin Than Mya, "Handwritten Character Segmentation in Tablet Based Application", *Proceedings of the 8th IEEE Global Conference on Consumer Electronics,* Osaka, Japan, Oct. 15-18 2019.