International Journal of Education and Social Development
ISSN: 3078-9931 (Print), ISSN: 3078-2287 (Online) | Vol. 2, No. 1, 2025

Exploration and Research on OCR Technology for Standardized Yi Character

Junping Huang[1, *]

[1]Shangsi County, Fangchenggang City, Guangxi Zhuang Autonomous Region, PRC
[*]Corresponding author: Junping Huang (Email: hip2049@126.com)

Abstract: Optical Character Recognition (OCR) refers to the process of analyzing, understanding, and recognizing textual information in image files, which is beneficial for extracting and collecting standardized Yi text information from paper materials. Using Tesseract-OCR software, based on the coding standards of standardized Yi characters and the font characteristics of standardized Yi characters, a deep learning training and recognition result validation were conducted using Long Short Term Memory Neural Network (LSTM) for standardized Yi character recognition. The Python programming language called OpenCV library, Pytesseract library, and PyQt5 library to achieve the construction of a standardized Yi character recognition system. Experiments show that the system can realize the characters recognition of Yi characters in Baiti, Songti, Heiti and Xiheiti, and has off-line operation and high recognition accuracy.

Keywords: Standardized Yi character; Optical Character Recognition; Deep learning; OpenCV; Tesseract-OCR.

1. Introduction

The "Standard Scheme for the Yi Language" was officially promulgated and implemented by the State Council in August 1980, which selected and standardized 819 basic characters and 345 characters of high pitch symbols, plus one alternative phonetic symbol, for a total of 1165 Yi language characters. In China, a large number of paper publications such as newspapers, magazines, and books related to the Yi language are issued every year. For example, Yi language newspapers include "Liangshan Daily"; Yi language publications include bimonthly "Liangshan Literature" and quarterly "Nationality"; Yi language classics, literary works, legal system, popularization of science, and Yi language translations of the National People's Congress and Chinese People's Political Consultative Conference white papers; Educational books such as Yi language textbooks, teaching materials, teaching aids, and reference books for primary and secondary schools. The large-scale publication and distribution of Yi language publications have promoted the promotion, learning, and use of Yi language. However, due to the fact that most publications are paper materials, it is not convenient to extract, collect, and build a corpus of Yi language text information. Nevertheless, Optical Character Recognition technology can effectively solve these problems. The concept of OCR was first proposed by German inventor Tausheck in the late 1920s, and related technological research began in the 1950s. OCR refers to the process of analyzing, understanding, and recognizing text information in image files. It has been widely used in character recognition scenarios such as license plate recognition, bank card recognition, and ID card recognition.

In China, domestic scholars have conducted relevant research on OCR technology for Yi character. Liu Sai and Li Yidong [1] used text segmentation technology to recognize scanned images of Yi character fonts; Wu Bing [2] discussed the national annotation scheme and international standard scheme for Yi character coding, and conducted measurement, data statistics, and feature analysis research on eight Yi character fonts. In recent years, the rapid development of Artificial Intelligence technologies such as artificial neural networks and computer vision in the field of image recognition has provided new technical support for Yi character recognition. Based on this, a set of Yi character recognition system is constructed by calling third-party libraries through Python programming language, in order to provide technical support for the informatization of Yi language.

2. Experimental Design

2.1. Experimental process

The process of standardized Yi characters Optical Character Recognition mainly includes image acquisition, image preprocessing, character segmentation, character recognition, and recognition result output. Optical Character Recognition training includes designing character datasets, generating initial models, character recognition pre-training, fine-tuning training, and merging training files. The process is shown in the following figure (Figure 1):

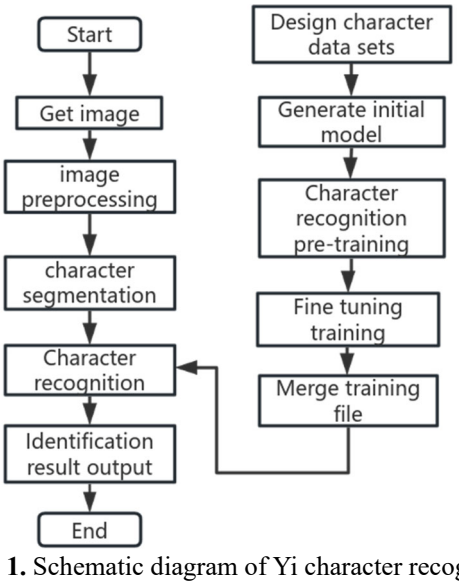


Figure 1. Schematic diagram of Yi character recognition process

2.2. Design of Yi character Dataset

The "Universal Multiple-Octet Coded Character Set" for Yi Language is an international standard scheme for Yi language, which was completed and formulated in December 1993. From 1994 to 1999, after multiple rounds of international conference discussions and voting, it was finally approved and included in the 2000 edition of the International Information Standards Collection. The international standard scheme specifies that the encoding range of 1165 Yi characters in Unicode is A000 to A48C.

Due to the lack of a pre-trained Yi character recognition library on Tesseract's official website, in order to achieve image recognition of Yi characters, it is necessary to train the Yi character recognition library from scratch. Firstly, a Yi character dataset needs to be designed, which includes the following three parts:

(1) Pre-training character dataset: The pre-training character set includes 10 Arabic numerals, 52 uppercase and lowercase English letters, 49 punctuation marks, and 1165 Yi characters, totaling 1272 characters;

(2) Text dictionary file and fine-tuning training character dataset: Approximately 14000 Yi language words were obtained from "Yi Language Vocabulary Definition" and "Yi Han Dictionary", with a total of 40100 Yi language characters included in all words.

(3) Pre-training and fine-tuning training character data set: 1520196 Yi characters, numbers, letters and punctuation marks, including 1440646 Yi characters, were obtained from Yi e-books and newspapers such as Selected Reading of Yi Folklores, Lu Han's Story, Mother's Daughter, Mamuteyi, Civil Code, the Novel Coronavirus Public Psychological Protection Manual (Yi-Han Edition), the Novel Coronavirus Infection Protection (Yi-Han Edition), Liangshan Daily (Yi Language Edition), etc. Select four commonly used Yi character fonts, including Baiti, *Songti*, Heiti, and Xiheiti, as the research objects. The four types of Yi character fonts are shown in the following figure (Figure 2):

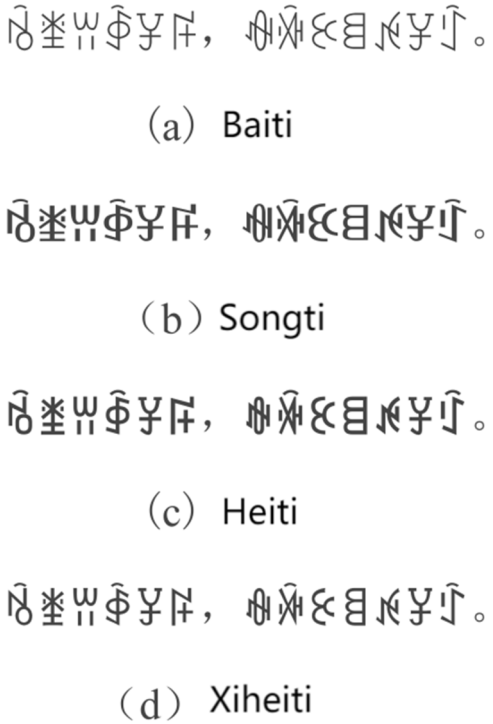


Figure 2. Four types of Yi character fonts

2.3. Software and hardware configuration

Before the design and deployment of Yi Optical Character Recognition system, the software and hardware system environment should be built. The software and hardware configuration environment is shown in the following table (Table 1):

Table 1. Software and Hardware Configuration Table

Software Name	Version	Others
Operating system	Windows10(64bit)	
Python	3.1	Computer programming language
Tesseract-OCR	V5.3.0	Open source Optical Character Recognition software
PyTesseract	0.3.10	Python Optical Character Recognition Library
OpenCV	4	Computer Vision Library
PythonQT(PyQt)	5	Graphic User Interface Library
Hardware name	**Version**	**Others**
Lenovo Workstation	ThinkStation E32 TVT	
Memory	HSGM DDR3 1600MHz 8GB*4	32G
Hard disk	WD10EURX-63FH1Y0	1TB
Digital composite machine	Canon iR-ADV 4025/4035	Used for scanning files

3. Image Preprocessing

The quality of image preprocessing results directly affects the results of text recognition. Character images are generally obtained through methods such as scanning with a scanner, taking photos with a camera, and taking screenshots. During the image acquisition process, there may be issues such as uneven exposure, excessive noise, and tilt in the character image, which require targeted image preprocessing. OpenCV is a cross platform computer vision library that can be used to process various types of images and has a large number of programming language interfaces such as Python, Java, and Matlab. OpenCV has a wide range of applications, including image processing, character recognition, facial recognition, machine vision, and other fields, making it widely used. Given the excellent performance of the OpenCV computer vision library, it can be called through the Python programming language to perform image preprocessing such

as grayscale, binarization, denoising, and tilt correction on character images.

3.1. Image Grayscale

Image grayscale is the process of transforming an image from color to gray. Color images are composed of three components: RGB (red, green, blue) with values ranging from 0 to 255, and grayscale is the process of making the three color components equal. After completing the grayscale transformation of the image, pixels with higher pixel values are brighter (the maximum value is 255, which is white), while those with lower pixel values are darker (the minimum value is 0, which is black). Due to the varying sensitivity of the human eye to color, the character image can be grayscale processed using the weighted average method according to equation (1):

$$Gray(i, j) = 0.299 * R(i, j) + 0.578 * G(i, j) + 0.114 * B(i, j) \quad (1)$$

The comparison image of the color image before and after grayscale processing (Figure 3) is as follows:

(e)Color image (f)Grayscale image

Figure 3. Image Grayscale Processing

3.2. Image binarization

Image binarization is the process of setting the grayscale values of all pixels in a grayscale image to 0 or 255. The binary image has only two states for each pixel: pure white (pixel grayscale value of 255) and pure black (pixel grayscale value of 0). Otsu algorithm (OTSU algorithm) is an algorithm that determines the threshold for binary image segmentation. It was proposed by Japanese scholar Otsu in 1979 and is also known as the maximum inter-class difference method. The image can be binarized using a determined segmentation threshold. The principle of Otsu method: Assuming there is image $I(x, y)$, the binary segmentation threshold of the image is denoted as τ, the proportion of pixels larger than the segmentation threshold in the entire image is denoted as ω_1, and the average grayscale is denoted as μ_1; The proportion of pixels below the segmentation threshold in the entire image is denoted as ω_2, the average grayscale is denoted as μ_2, the total average grayscale of the image is denoted as μ, and the inter-class variance is denoted as g. The size of the image is $M \times N$, the number of pixels in the image with grayscale values less than the threshold τ is denoted as K_1, and the number of pixels with grayscale values greater than the segmentation threshold τ is denoted as K_2, then there are:

$$\omega_1 = K_1 / M \times N \quad (2)$$

$$\omega_2 = K_2 / M \times N \quad (3)$$

$$M \times N = K_1 + K_2 \quad (4)$$

$$\omega_1 + \omega_2 = 1 \quad (5)$$

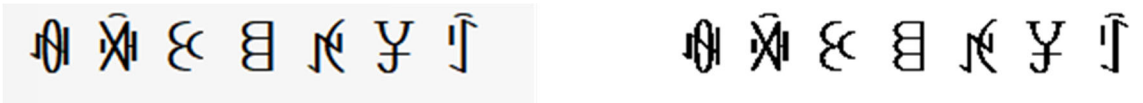
$$\mu_1 \times \omega_1 + \mu_2 \times \omega_2 = \mu \quad (6)$$

$$g^2 = \omega_1 \times (\mu - \mu_1)^2 + \omega_2 \times (\mu - \mu_2)^2 \quad (7)$$

Substitute equation (6) into equation (7) to get formula (8), as follows:

$$g^2 = \omega_1 \times \omega_2 \times (\mu_1 - \mu_2)^2 \quad (8)$$

Use traversal method to traverse the entire image, equation (8) can be used to calculate the maximum inter-class variance value τ, which is the threshold for binary segmentation of the image. Through experiments, it has been found that the Otsu method has a good effect on binarizing character images. The comparison image of the character image before and after binarization processing (Figure 4) is as follows:



(g)Grayscale image (h)Binarization image

Figure 4. Image binarization processing

3.3. Image Denoising

During the process of image acquisition and processing, various noise points are generated in the image due to various reasons. Most image noise is imperceptible to the human eye, but it can easily have a negative impact on character recognition in character images. Liang Hao [3] pointed out that both granular noise and thermal noise belong to Gaussian

white noise . Gaussian filtering is a type of linear smoothing filter, which is a type of mean filtering suitable for eliminating Gaussian noise in images and smoothing them. Gaussian filtering requires weighted averaging of the entire image, where the value of each pixel is obtained by weighted averaging its own pixel value with the values of other pixels in the neighborhood.

The formula for one-dimensional Gaussian function is:

$$G(x)=\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{x^2}{2\sigma^2}} \tag{9}$$

The formula for the two-dimensional Gaussian function is:

$$G(x,y)=G(x)\cdot G(y)=\frac{1}{2\pi\sigma^2}e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{10}$$

Gaussian filtering has good denoising properties and can be used to denoise character images. The comparison of the character image before and after Gaussian filtering (Figure 5) is as follows:

(i)Binarization image (j)Gaussian filtered image

Figure 5. Image Gaussian filtering processing

3.4. Image Correction

Tilted character images can have various impacts on character image segmentation, which in turn affects character recognition results. Therefore, tilted character images need to be pre-corrected for tilt. Hough transform is a commonly used image correction method. Firstly, the image pixels are mapped from the Cartesian coordinate system to Hough space, and points with many intersecting lines are calculated from Hough space. Then, these points are mapped back to the Cartesian coordinate system, and the straight line segments in the character image can be extracted. The edge lines are fitted in the straight line segments, and the tilt angle and direction of the image can be estimated based on the distribution angles of these straight line segments.

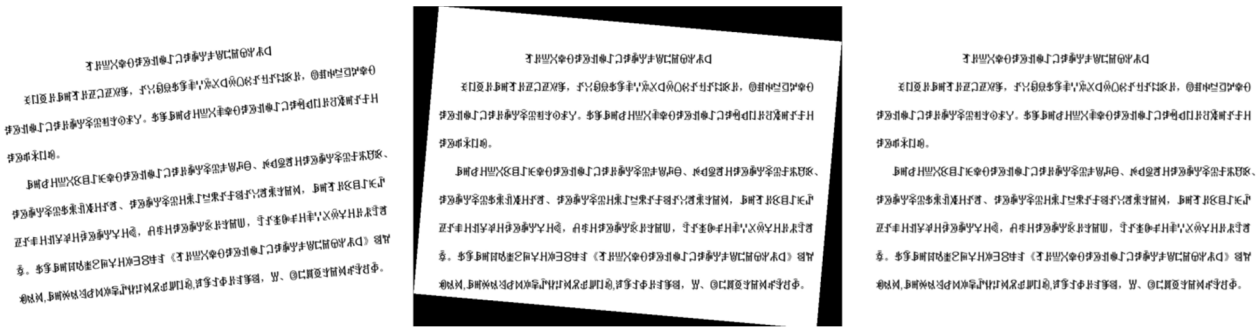
The Hough transform utilizes the correspondence between lines and points in the Cartesian coordinate system and Hough space to determine the tilt angle of an image. In the Cartesian coordinate system, the linear expression is $y=kx+b$, where x and y are the variables, k is the slope, b is the intercept, and k and b are the variables in Hough space. When N points in the Cartesian coordinate system are distributed on the same straight line $y=k_1x+b_1$, and these N points correspond to the Hough space, N straight lines passing through point (k_1,b_1) will be formed; On the contrary, the more lines a certain intersection point passes through in Hough space, the more lines it corresponds to in the Cartesian coordinate system, resulting in a line that passes through more points.

In order to avoid the slope problem caused by lines $x=b$ or $y=d$ in the Cartesian coordinate system, the polar coordinate equation is introduced in the Cartesian coordinate system, where ρ represents the distance from coordinate point (x,y) to the origin, θ represents the angle between the point on the line and the horizontal axis, and the polar coordinate equation is expressed as:

$$\rho=x\cos(\theta)+y\sin(\theta) \tag{11}$$

In image correction processing, assuming there is an image of size $M\times N$, the Hough transform needs to be used to detect the tilt angle in the image, and the pixels in the image can be represented by polar coordinate point (ρ,θ). First, create a two-dimensional array accumulator with a size of $\rho\times\theta$, initialize all values to 0, represent rows as ρ, and list as θ. The range of θ values is $[0^\circ,180^\circ]$, and the precision of θ values in the array determines the accuracy of the results. If the angle precision of θ is set to 1°, 180 columns are required. The maximum value of ρ is the distance $\sqrt{M^2+N^2}$ between the diagonal lines of the image. If you want to achieve pixel level accuracy, the size of the rows should be equal to the distance between the diagonal lines of the image. Take the first point (x_0,y_0) on the image and input it into the polar coordinate equation (Equation 11). Then iterate through θ values: 0, 1, 2... 179, 180, and calculate the corresponding ρ value for each θ value. If this value has a corresponding position in the accumulator, add 1 to that position. Take the second point (x_1,y_1) on the image, repeat the above operation, and update the value in the two-dimensional array accumulator. Traverse all pixels in the image and update the value in the accumulator each time. Finally, search for the larger values in the accumulator and find the polar coordinate points (ρ,θ) corresponding to these values. Then, these values correspond to the polar coordinate equation in the Cartesian coordinate system, which is the straight line being calculated. Finally, calculate the tilt angles of all lines, take their average, and perform rotation correction on the image based on the calculated average tilt angle.

Compared with other image correction methods, Hough transform has better anti-interference ability against incomplete lines and noise in images, so Hough transform can be used to correct images. The comparison image of the character image before and after Hough correction processing (Figure 6) is as follows:



(k)Tilted image (l)Hough transform image (m)After removing borders

Figure 6. Image Hough Transform Processing

4. Character Recognition Training

Tesseract OCR is an open-source OCR library with high accuracy and flexibility, which can recognize specified types of characters through training. PyTesseract is an OCR recognition library that encapsulates Tesseract OCR with Python API. Therefore, before using PyTesseract, Tesseract OCR needs to be installed and the running environment configured. Tesseract OCR V5.3.0 has implemented a deep learning recognition engine based on long short-term memory (LSTM) neural network, which greatly improves the accuracy of character recognition. Based on the above advantages of Tesseract OCR V5.3.0, it is used for Optical Character Recognition training and character recognition of commonly used Yi characters.

4.1. Training and Evaluation

4.1.1. Generate initial model

According to Tesseract OCR official website [4], to construct a Yi language recognition model, the first step is to generate starter traineddata (initial model). The steps for generating the model are as follows:

(1) Generate word.txt dictionary file: Save 14000 Yi language words obtained from "Yi Language Vocabulary Definition" and "Yi-Han Dictionary" to the word.txt dictionary file, with one word per line;

(2) Generate a number. txt dictionary file: Save 10 Arabic numerals, including 0-9, one digit per line to the number. txt dictionary file;

(3) Generate punc.txt punctuation dictionary file: Save 55 punctuation marks, one per line, to the punc.txt punctuation dictionary file;

(4) Generate character set lstm-unicharset file: The lstm-unicharset file contains 1165 Yi characters, Arabic numerals 0-9, 52 English uppercase and lowercase letters, and 55 punctuation marks, totaling 1272 characters;

(5) Merge and generate the starter traineddata initial model from the word.txt dictionary file, number.txt dictionary file, punc.txt punctuation dictionary file, and lstm-unicharset file.

4.1.2. Pre-training and Evaluation of Character Recognition

The Microsoft Yi Baiti font, which comes with the Windows 10 operating system, contains numbers, English letters, punctuation marks, and 1165 Yi characters, while the Yi fonts designed by Peking University Founder only contain Yi characters. Therefore, Microsoft Yi Baiti font is selected as the training font for Yi recognition pre-training. The pre-training and evaluation steps for Yi character recognition are as follows:

(1) Preparation of pre-training character dataset: 1272 characters consisting of numbers, letters, punctuation marks, and Yi character, first copy 1272 characters 200 times each to obtain 254800 characters. In order to increase the complexity and disturbance of the text, 254800 characters are randomly sorted in the newly created train.txt file;

(2) Use the text2image command, set ptsize=50 for rendering, and generate Tif images and Box files from the train.txt file;

(3) Generate training file lstm.train;

(4) Create a new training list file train_istfile.txt;

(5) Pre-training for Yi character recognition, specific information is shown in Table 2;

(6) Generate evaluation file: The text used for evaluation should include as many training characters as possible. Save the 1272 characters from step (1) to the e: \ eval.txt evaluation file;

(7) Generate Tif images and Box files from the evaluation file eval.txt;

(8) Generate evaluation eval.lstmf file;

(9) Generate evaluation list file: Create a new eval.listfile.txt file with the content "e: \ eval.lstmf" ;

(10) Evaluate the pre-training results of Yi character recognition, as shown in Table 2;

(11) Merge the pre-trained output_checkpoint file with the starter traineddata initial model to generate a traineddata character recognition file, thus completed the character recognition pre-training.

Table 2. pre-training and Evaluation Information Table for Yi character Recognition

S/N	Yi character font	Number of characters	Training duration	Training completion criteria	Identification error rate assessment
1	Baiti	254800	7 days, 6 hours, and 29 minutes	Target error rate<0.001%	0.209%

4.1.3. Character recognition fine-tuning training and evaluation

Select four Yi character fonts designed by Peking University Founder, including Baiti, Songti, Heiti, and Xiheiti, for character recognition fine-tuning training. Due to the fact that the font library of these four Yi fonts only contains 1165 Yi characters, the characters used for fine-tuning training do not include Arabic numerals, English letters, punctuation marks, and other characters. The fine-tuning training steps for the four Yi character fonts are as follows:

(1) Preparation of fine-tuning character dataset: Create a train.txt file and copy the 40100 Yi characters used for fine-tuning training to the train.txt file, with approximately 65 characters per line;

(2) Retrieve LSTM file: Extract the LSTM file from the pre-trained character recognition file traineddata;

(3) Set the font of Yi characters in the train.txt file and generate Tif images and Box files;

(4) Generate training file lstm.train;

(5) Fine-tuning training for Yi character recognition, specific information is shown in Table 3;

(6) Generate evaluation file eval.txt: The eval.txt file contains 1165 Yi characters;

(7) Generate Tif images and Box files from the evaluation file eval.txt;

(8) Generate evaluation eval.lstmf file;

(9) Generate evaluation list file: Create a new eval.listfile.txt file with the content "e: \ eval.lstmf" ;

(10) Evaluate the fine-tuning training results of Yi character recognition, as shown in Table 3;

(11) Merge the outputted checkpoint file generated by fine-tuning training into the traineddata character recognition file to complete fine-tuning training.

Table 3. Information Table for Fine-tuning Training and Evaluation of Yi character Recognition

Fine-tuning training epochs	Yi character font	Training duration	Training completion criteria	Identification error rate assessment
Round 1	Baiti	3h24m	Target error rate<0.001%	0.124%
Round 2	Songti	1h34m		0.428%
Round 3	Heiti	1h53m		1.106%
Round 4	Xiheiti	3h2m		1.041%

4.2. Character Recognition Verification

The Yi character recognition verification character dataset includes four Yi character fonts: Baiti, Songti, Heiti, and Xiheiti, each with 20069 Yi characters. The character image is first processed through grayscale, binarization, denoising, and correction, and then subjected to Yi Optical Character Recognition. Through analysis and statistical analysis of the recognition results of Yi character, it is found that: (1) the character recognition speed of the four Yi character fonts varies slightly, ranging from 130 characters/second to 140 characters/second; (2) The misidentification rate is the percentage of the number of Yi characters that are misidentified to the total number of Yi characters. The misidentification rate for Baiti and Songti is around 6%, while the misidentification rate for Heiti and Xiheiti is relatively small, both slightly less than 2.7%; (3) The unrecognized rate is the percentage of unrecognized Yi characters to the total number of Yi characters. The unrecognized rates of the four Yi fonts are relatively small, ranging from 0.2% to 0.6%; (4) The recognition accuracy of Yi character Baiti and Songti is between 93% and 94%, while the recognition accuracy of Yi character Heiti and Xiheiti is relatively high, both slightly greater than 97%. The Yi character recognition verification information table (Table 4) is as follows:

Table 4. Yi character Recognition Verification Information Table

S/N	Yi character font	Recognition speed	Error rate	Unrecognized rate	Recognition accuracy
1	Baiti	132 c/s	5.725%	0.558%	93.716%
2	Songti	136 c/s	6.423%	0.239%	93.338%
3	Heiti	135 c/s	2.681%	0.304%	97.015%
4	Xiheiti	131 c/s	2.691%	0.299%	97.010%

5. Software System Design and Implementation

On the Windows 10 (64 bit) operating system, Python is used as the development language to call the OpenCV library (responsible for image preprocessing), Pyteseract library+Tesseract OCR (responsible for Yi character recognition), and PyQt5 library (responsible for GUI interface design and construction). Install and configurethe above third library and software in Python to design a Yi character recognition system.

5.1. PyQt5

PythonQt5 (PyQt5) is a toolkit for developing graphical user interfaces (GUI) using the Python programming language that integrates the Qt library. It is suitable for mainstream operating systems such as Windows, Linux, Unix, and MacOS. PyQt5 has excellent visualization effects, rich functional modules, and diverse classes. The functional modules used include QtCore, QtGui, QtDesigner, UIC, and QtWidgets, as well as classes QApplication, QWWidget, QFrame, and QMainWindow. The functional modules, classes, and their functions used are shown in the following table (Table 5):

Table 5. PyQt5's functional modules, classes, and their respective function tables

Module Name	Module functionality
QtCore module	Covering the core non GUI functions of the package
QtGui module	A class that includes multiple basic graphic functions
QtDesigner module	The included classes allow the use of PyQt to extend Qt Designer
UIC module	The included classes are used to handle. ui files
QtWidgets module	Contains UI control elements for creating desktop applications
Class name	**Class Function**
QApplication class	Used to manage the control flow and main settings of graphical user interface applications
QWidget class	Base class for all user interface objects
QFrame class	It is a base class and a framed window control
QMainWindow class	Provide the main application window with menu bar, anchor window, status bar, etc

5.2. Implementation of Yi character recognition system

The implementation of the Yi character recognition system first involves importing the PyQt5 library into the Python programming language to design and deploy the graphical user interface. Then, the OpenCV computer vision library is used to preprocess the imported character images. During the character image recognition process, the Pyteseract character recognition library calls the Tesseract OCR engine through the API interface to complete the character recognition. The final character recognition result is output to the "Yi character recognition result" display bar on the user interface. The button "Copy Recognition Result" can be used to copy the recognized Yi character text. The character recognition performance of the Yi character recognition system is shown in Figure 7 as follows:

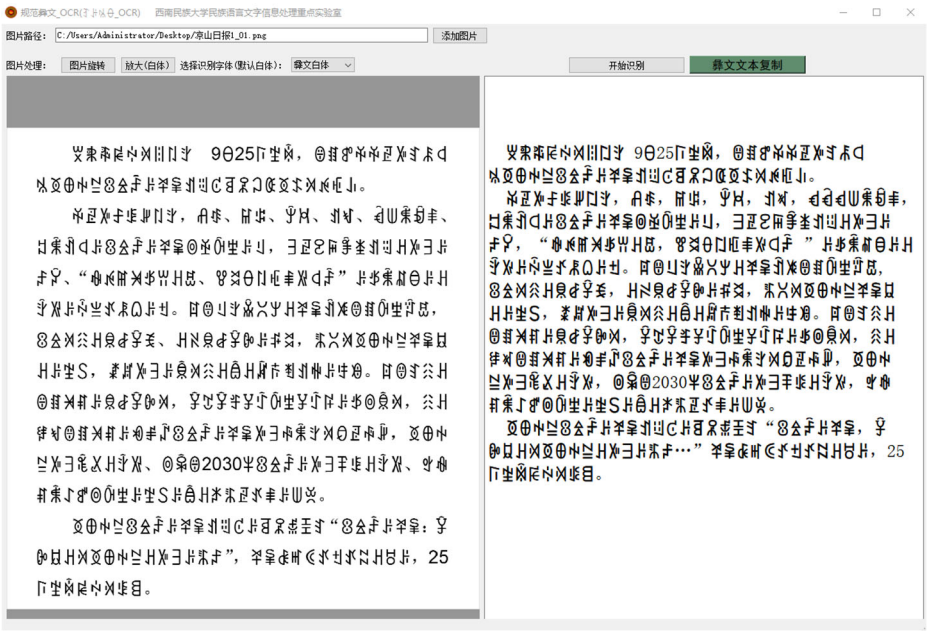


Figure 7. Yi character recognition system

6. Conclusion and Prospect

By using long short-term memory networks for deep learning training of Yi character recognition, and utilizing Python programming language to call OpenCV computer vision library, Pyteseract library+Tesseract OCR engine, and PyQt5 library, a Yi character recognition system is constructed, which includes image import, image preprocessing, character recognition, and recognition result output. The system has offline operability and high recognition accuracy in recognizing printed Yi characters such as Baiti, Songti, Heiti, and Xiheiti. It can also be applied to schools, libraries, publishing houses, and other places to perform character recognition and text output on various scanned or photographed Yi character images.

References

[1] Liu Sai ,Li Yidong.Design and Realization on CharacterSegmentation Method for Yi Language[J]. Journal of South-Central University for Nationalities (Nat. Sci. Edition). 2007(03): 70-72.

[2] Wu Bing.Research on the Analysis of Standardized Yi characters from the Perspective of Character Recognition[J]. Journal of Southwest Minzu University(Humanities and Social Sciences Edition). 2018, 39(09):46-53.

[3] Liang Hao .Study of Nonliner Gaussian Filters and its Application to CNS/SAR/SINS Integrated Navigation [D]. Harbin Institute of Technology,2015.

[4] https://github.com/tesseract-ocr/tesseract.