A TESSERACT-BASED OCR SYSTEM FOR BILINGUAL CHINESE-SERBIAN GROCERY ORDER PROCESSING

Marko Ercegovac[1]*,
[0009-0009-3629-3240]

Yao Zheng[2],
[0009-0002-6638-2323]

Angelina Njeguš[3]
[0000-0001-8682-7014]

[1]Student,
Singidunum University,
Belgrade, Serbia

[2]Agricultural University,
Beijing, China

[3]Singidunum University,
Belgrade, Serbia

Abstract:

This paper presents a new method for the ordering process in Chinese grocery stores using Tesseract OCR for bilingual Chinese-Serbian translation. The system is implemented as a Telegram chatbot, allowing users to capture images of orders, which are then automatically processed into CSV delivery lists. To ensure high performance on low-power devices, we used the Tesseract LSTM engine. To improve accuracy, all images are pre-processed before being passed to the OCR engine, and regular expressions (regex) are used to extract and normalize units and quantities from the recognized text. A domain-specific dictionary containing approximately 3,000 translations serves as a robust error-handling mechanism to mitigate OCR misclassifications. This hybrid approach, combining neural recognition with expert linguistic rules, ensures high reliability even when processing low-quality images from mobile chat applications. The final output provides structured data, including company names, item lists, and units, in order to reduce manual entering CSV data for drivers and logistics.

Keywords:

Tesseract OCR, LSTM, Bilingual Translation, Telegram Chatbot, Logistics Automation.

INTRODUCTION

Working with a large number of orders is quite exhausting. Many food stores use managers who receive orders and manually create and organize them into documents such as Excel tables. Customers are accustomed to this way of ordering, but behind the scenes is a highly organized and orchestrated group of people delivering within tight deadlines. A domain specialist manager in a medium-sized grocery store needs around five hours to prepare everything for printing and delivering to drivers.

The process is quite challenging if the shop is owned by foreigners, such as certain Chinese grocery stores. They must provide orders to employees who are often not Chinese and are not familiar with the Simplified Chinese language. These employees must prepare the goods for orders that are not in their native language.

Correspondence:
Marko Ercegovac

e-mail:
marko.ercegovac.25@singimail.rs

DOI: 10.15308/Sinteza-2026-220-225

Drivers require a delivery list, especially for long-distance deliveries. This workflow is difficult for employees to maintain high customer service standards, and as the business grows, preparation time increases significantly.

The focus of this paper is the implementation of an automated method for a Chinese grocery store in Serbia. Also, the solution can be applied to similar businesses. The key to this solution lies in adapting Google Tesseract to handle Simplified Chinese text from orders. In our specific case study, customers are used to ordering via chat applications like WeChat, but this paper introduces a similar approach using a Telegram Chat Bot. We propose a new method for these foreign groceries to automate the receiving process, process text from deliveries, and generate delivery documents for drivers.

2. OCR WITH NEURAL NETWORKS

OCR (Optical Character Recognition) has wide application in various areas, from processing text in images to recognizing handwritten documents. It was first introduced to assist blind people and convert characters into Braille [1] Code. Early OCR systems were used in banking systems for automated document processing, relying primarily on structural pattern matching without the use of neural networks. A major paradigm shift toward the use of neural networks happened in 1989 when a CNN named LeNet-1 [2] was used to read digits.

2.1. TESSERACT OCR

Tesseract was developed at HP Laboratories [2] and, in its early stages, did not utilize deep neural networks. It remained in this form until its adoption by Google Inc.

They reworked the main Tesseract OCR engine and extended it to support more than 100 languages in 2006 [3] introducing compatibility with the Han script (used for Chinese, Japanese, and Korean). Furthermore, they adopted the Russian Cyrillic script, which paved the path to supporting the Serbian Cyrillic script [3]. In 2018, Tesseract Engine 4.0 [4] was introduced with support for LSTM (Long Short-Term Memory) as a recurrent neural network. By utilizing a Bidirectional LSTM (BiLSTM) architecture, the engine performs a double-pass verification of the input sequence. This is particularly critical for the Han script, as the model captures long-range dependencies and context from the entire line of text, significantly reducing ambiguity in complex character recognition.

2.2. LONG SHORT-TERM MEMORY (LSTM) CELL

Figure 1. illustrates the internal architecture of a Long Short-Term Memory (LSTM) cell, which serves as the core computational unit in Tesseract 4.0 [4].

The gating mechanism is represented which includes forget gate (ft), this gate determines whether to keep the information or to discard it [6]. The current input and the previous hidden state are processed by a sigmoid layer. This layer outputs a value between 0 and 1. If the output value is closer to 1, then keep the information. Else, forget the information. The output value of forget gate is computed by formula (1)

$$f_t = \sigma(W_t \cdot [a_{t-1}, X_t] + b_f) \quad (1)$$

Where f_t, denotes the activation vector of the forget gate at time t, σ represents the sigmoid activation function that maps values between 0 and 1, W_t is the weight matrix associated with the gate, a_{t-1} signifies the hidden

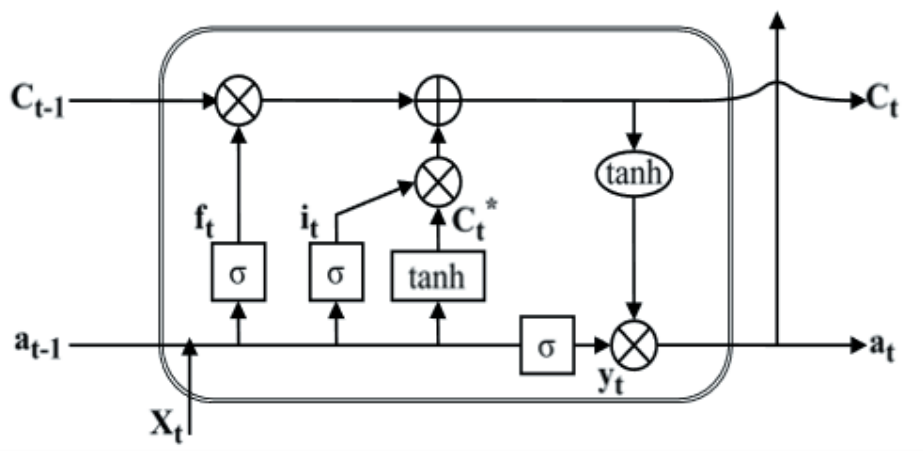

Figure 1. Architecture of an LSTM memory cell [5]

state activation from the previous time step, X_t refers to the current input vector, and b_t is the bias vector added to the linear transformation.

Input gate is updating cell state. Current input and the previous hidden state are processed by a sigmoid and a tanh separately. Sigmoid converts a data value to a value between 0 and 1 while tanh converts a data value to a value between -1 and 1. A point-wise multiplication operation is performed on the outputs of the sigmoid layer and the tanh layer. The new cell state value is then computed. The output of the input gate is computed by formula (2) where Wi is the weight matrix associated with the input gate.

$$i_t = \sigma(W_i \cdot [a_{t-1}, X_t] + b_i) \quad (2)$$

The output of tanh is computed by using formula (3)

$$C_t^* = tanh(W_c \cdot [a_{t-1}, X_t] + b_c) \quad (3)$$

The new cell state is computed by using formula (4)

$$C_t = f_t \cdot C_{t-1} + i_t \cdot C_t^* \quad (4)$$

The output gate determines the next hidden state and computes the output using formula (5)

$$y_t = \sigma(W_y \cdot [a_{t-1}, X_t] + b_y) \quad (5)$$

The next hidden state is computed by formula (6)

$$a_t = y_t \cdot tanh\,(C_t) \quad (6)$$

Unlike a standard LSTM, Tesseract 4.0 is implemented with a Bidirectional LSTM (BiLSTM) architecture [7], which processes the input sequence in both directions to enhance recognition accuracy through double-pass verification. For Bidirectional LSTM, two independent hidden layers of these cells process the input sequence in both forward and backward directions. This allows the model to capture context from the entire line of text, which is crucial for accurately identifying complex characters in the Han script.

2.3. INTELLIGENT DOCUMENT PROCESSING (IDP)

The Intelligent Document Processing workflow begins with an image preprocessing using the Pillow library. Input of neural network are images where operations such as resizing, noise reduction, and binarization are applied to optimize the input. After the process of text extraction, raw recognized text is post-processed using Regular Expressions (Regex) with domain-specific dictionary.

3. STATE OF THE ART (SOTA)

The current state-of-the-art in OCR is Baidu's Paddle OCR (v5) [8], which leverages hybrid Transformer-based architectures [9]. It is a highly optimized ecosystem supporting PyTorch (via conversions) and NVIDIA GPU acceleration, enabling the recognition of multi-language, complex-structured documents with high precision. The latest versions utilize Vision Transformer (ViT) technology and offer lightweight models optimized for edge computing and mobile devices. These models are exceptionally fast, often completing recognition tasks in under 300ms.

Furthermore, PP-OCRv5 provides robust support for handwritten text and includes a complete pipeline for preprocessing, including horizontal and vertical text detection. It is a high-precision and lightweight OCR system designed to perform effectively in a wide range of scenarios, supports a diverse range of scripts within a single model, including Simplified Chinese, Traditional Chinese, Chinese Pinyin, English, and Japanese. The pipeline includes image preprocessing, text region detection, text line orientation classification, and text recognition, ultimately extracting the text from images and outputting it as structured textual content.

PP-OCRv5 is a pipeline designed with different modules. The first module is an image preprocessing module designed to process the input image, enhance quality and adjust distortion or orientation issues. The second module is a text detection module that recognizes areas where text is located, even if the background is not clean or if there are problems with text complexity or orientation. It uses massive GOT-OCR2.0 [10] model as teacher and lightweight model with high precision as student.

The third module's main feature is text rotation and acts as the main bridge with detection and recognition, specifically responsible for identifying and rectifying the direction of detected text lines. If text is inverted or rotated, this model automatically corrects it to a standard, readable orientation, ensuring that the subsequent recognition engine receives properly aligned input.

The final module it uses transformers to understand text and solve difficult cases like handwritten or rare characters, PP-OCRv5 utilizes ERNIE-4.5-VL for automated high-quality data annotation and filtering, ensuring the model is exceptionally robust across diverse, real-world document formats.

While PP-OCRv5 is superior in terms of raw accuracy and speed due to its massive pre-training, Tesseract OCR remains a competitive engine because it allows for predefined domain knowledge and regular expressions to improve character hitting, whereas PP-OCRv5 relies more on its deep learning architecture.

4. METHODOLOGY

For image manipulation and various preprocessing techniques, the OpenCV library is used [11]. Images are processed using different techniques such as grayscale conversion, erosion (to reduce noise), and dilation (to expand pixel areas), followed by contour detection to get the best results. The dataset consists of screenshots from WeChat and WhatsApp, and the system is developed and tested using low-resolution images. The development environment is Kaggle and Tesseract 4.1.1. For text recognition, the system uses the chi_sim pre-trained model from the official Tesseract GitHub [12] repository to support Simplified Chinese.

The contours extraction algorithm uses the previously mentioned preprocessing steps, combining Otsu's binarization [13] and Gaussian blur techniques. After all steps of the preprocessing stage are complete, Regions of Interest (ROIs) are generated (as shown in Figure 2). These regions are then passed to the Tesseract core engine, which attempts to recognize and extract the text.

Although the initial image was pre-processed for extracting box contours, the results remained poor until a second round of preprocessing was performed on each contoured text segment. Due to Tesseract's requirement for characters to be 30–40 pixels in height, the engine tends to "hallucinate" incorrect characters if the input fails to meet these dimensions. To prevent these hallucinations caused by low-resolution inputs, we implemented a dynamic scaling layer using interpolation to ensure a consistent character height. This enhancement significantly improved the engine's stability, leading to a 100% recognition rate for the evaluated Regions of Interest (ROIs) shown in Figure 2.

Post-processing involves detecting named entities such as company names, product names, units, and quantities. Regular expressions (regex) are used to extract company names, while a combination of regex and domain-specific unit dictionaries is used to extract quantities and units.

Once the text is extracted, product names are sent to a translation API powered by Google Deep Translator, for translation into Serbian. If a unit is not found in predefined dictionary, the system calls translation API to translate the Chinese units. Finally, all the gathered data, including the translated Serbian names of the products and units, is used to generate the Excel file shown in Table 1.

Figure 2. Example of generated ROI on Whatsup image

Table 1. An example of generated Excel table for company products

Chinese name of product	Serbian name of product	Unit	Quantity
鸭头	pačja glava	kg	3
鸭脖	pačji vrat	kg	3
鸭掌	pačje noge	kg	3
去皮大蒜	oljušteni beli luk	pakovanje	5

To measure the quality of the OCR, the Levenstein distance [14] is used, implemented with the Character Error Rate (CER) method. A predefined ground truth lists are compared with our unofficial data collection. The basic formula for the Character Error Rate is shown in Equation (7):

$$CER = \frac{(I + D + S)}{N} \quad (7)$$

Where:

- I = Number of character insertions
- D = Number of character deletions
- S = Number of character substitutions
- N = Total number of characters in the reference text

5. RESULTS AND CONCLUSION

As testing cases are used, small and poor-quality images are taken from WhatsApp. Tesseract was measured both with and without the application of pre-processing and post-processing techniques, as shown in Table 2.

Table 2. An example of a generated Excel table for company products

Version	CER
Tesseract without preprocessing and postprocessing	68.98%
Tesseract with preprocessing and postprocessing	11.93%

The results presented in Table 2 represent preliminary findings based on a targeted dataset of real-world grocery orders. While these results are indicative of the system's performance, further validation on larger, official datasets will provide a more comprehensive assessment of the model's scalability.

Additionally, a Telegram bot was implemented to receive orders and process them using Tesseract. This prepares the data for a manager to generate an Excel file, which will be printed and sent to drivers.

Although the current version of the solution could be extended by creating a new LoRA (Low-Rank Adaptation) for the existing chi_sim model, this remains out of scope for the current paper. However, it is a noteworthy direction for future development. Additionally, PaddleOCR could be utilized as a state-of-the-art (SOTA) solution for recognizing Chinese characters in order processing.

The primary contribution of this research lies in the development of a multi-stage preprocessing pipeline that adapts a general-purpose OCR engine for a highly specialized bilingual logistics domain. By bridging the gap between raw neural output and structured business data through expert-defined dictionaries and regex, the system achieves a commercially viable accuracy level on edge devices.

REFERENCES

[1] X. F. Hermida, A. C. Rodríguez, and F. M. Rodríguez, "A Braille O.C.R. for blind people.".

[2] Y. LeCun, L. Jackel, L. Bottou, A. Brunot, C. Cortes, J. Denker, H. Drucker, I. Guyon, U. Müller, E. Säckinger, P. Simard, and V. Vapnik, "Comparison of learning algorithms for handwritten digit recognition.".

[3] R. Smith, D. Antonova, and D.-S. Lee, "Adapting the Tesseract open source OCR engine for multilingual OCR," in *Proceedings of the International Workshop on Multilingual OCR*, Barcelona Spain: ACM, Jul. 2009, pp. 1-8. doi: 10.1145/1577802.1577804.

[4] D. Sporici, E. Cușnir, and C.-A. Boiangiu, "Improving the Accuracy of Tesseract 4.0 OCR Engine Using Convolution-Based Preprocessing," *Symmetry*, vol. 12, no. 5, p. 715, May 2020, doi: 10.3390/sym12050715.

[5] U. B. Mahadevaswamy and P. Swathi, "Sentiment Analysis using Bidirectional LSTM Network," *Procedia Comput. Sci.*, vol. 218, pp. 45-56, 2023, doi: 10.1016/j.procs.2022.12.400.

[6] "Understanding LSTM Networks -- colah's blog." Accessed: Mar. 28, 2026. [Online]. Available: https://colah.github.io/posts/2015-08-Understanding-LSTMs/

[7] R. C. Staudemeyer and E. R. Morris, "Understanding LSTM - a tutorial into Long Short-Term Memory Recurrent Neural Networks".

[8] C. Cui, T. Sun, M. Lin, T. Gao, Y. Zhang, J. Liu, X. Wang, Z. Zhang, C. Zhou, H. Liu, Y. Zhang, W. Lv, K. Huang, Y. Zhang, J. Zhang, J. Zhang, Y. Liu, D. Yu, and Y. Ma, "PaddleOCR 3.0 technical report," 2025, *arXiv*:2507.05595.

[9] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Advances in Neural Information Processing Systems*, 2017.

[10] H. Wei, C. Liu, J. Chen, J. Wang, L. Kong, Y. Xu, Z. Ge, L. Zhao, J. Sun, Y. Peng, C. Han, and X. Zhang, "General OCR theory: Towards OCR-2.0 via a unified end-to-end model," 2023.

[11] "OpenCV: OpenCV modules." Accessed: Apr. 09, 2026. [Online]. Available: https://docs.opencv.org/4.x/

[12] "tessdata/chi_sim.traineddata at main · tesseract-ocr/tessdata," GitHub. Accessed: Apr. 09, 2026. [Online]. Available: https://github.com/tesseract-ocr/tessdata/blob/main/chi_sim.traineddata

[13] X. Xu, S. Xu, L. Jin, and E. Song, "Characteristic analysis of Otsu threshold and its applications," *Pattern Recognit. Lett.*, vol. 32, no. 7, pp. 956–961, May 2011, doi: 10.1016/j.patrec.2011.01.021.

[14] R. Haldar and D. Mukhopadhyay, "Levenshtein Distance Technique in Dictionary Lookup Methods: An Improved Approach".