The text within this document is the similar to the original document.
If you read this document, please refer to my research.

C. Ratanavilisagul, L. Poempool, and S. Kosolsombat, "Improved Segmentation Technique for Handwritten Character Recognition," in Proceedings of the 14th 2026 International Electrical Engineering Congress (iEECON), 2026.

Improved Segmentation Technique for Handwritten Character Recognition

Chiabwoot Ratanavilisagul
Department of Computer and Information Sciences, Faculty of Applied Science, King Mongkut's University of Technology North Bangkok (KMUTNB)
Bangkok, Thailand
chaibwoot.r@sci.kmutnb.ac.th

Lukkana Poempool*
Department of Automated Manufacturing Engineering, Faculty of Industrial Technology, Rajabhat Rajanagarindra University
Chachoengsao, Thailand
lukkana.po@gmail.com

Somkiat Kosolsombat*
Data Science and Innovation, College of Interdisciplinary Studies, Thammasat University
Pathum Thani, Thailand
somkiatk@tu.ac.th

***Abstract*— Handwritten Character Recognition (HCR) is the task of interpreting handwritten text from documents or answer sheets. English is a common language used in general exams and programming, and many symbols are found on a standard keyboard. English characters can be easily segmented. Recently, many researchers have proposed using K-means, contour detection, line segmentation, word segmentation, and character segmentation. These techniques use the concept of space between characters. In the case of English characters and keyboard symbols, clear spaces usually appear, so these techniques can segment them easily. However, some characters and symbols cannot use the space concept for segmentation because they are divided into two parts. As a result, the segmentation results are incorrect. The process after segmentation is classification, which then produces incorrect results. Experimental results from this research indicate that line segmentation and K-means achieve the best results. This research proposes that the merge technique is executed after line segmentation and K-means clustering. In addition, performing the merge twice increases the chance of obtaining better segmentation results. This research also integrates classification experiments using EasyOCR to show that segmented images obtain better classification results than non-segmented images. Experiments were conducted on a personalized handwriting dataset collected from 50 participants. The results demonstrate that the proposed method has a lower segmentation error percentage than other compared methods. Moreover, the segmented images achieve better classification results than the non-segmented images.**

Keywords— Handwritten Character, Line Segmentation, Segmentation, Classification, K-means clustering, Machine Learning, EasyOCR

I. Introduction

English print type and Arabic numerals are important characters and are widely used in general examinations, general writing, and programming languages [1–3]. In addition to English letters and Arabic numerals, programming languages also require various symbols found on the keyboard. In general examinations or programming language examinations, handwriting on paper is still commonly used, employing English print type, Arabic numerals, and symbols found on the keyboard. These are referred to as characters in this paper. HCR is the task of interpreting handwritten text from sources such as documents and answer sheets [1]. Before interpretation, these texts must be segmented in order to split characters into individual units, which makes character classification easier. HCR has many stages such as preprocessing, segmentation, feature extraction, and classification. So, segmentation, which is an important part, divides text images into lines, words, and characters [4, 5].

Recently, many research works have proposed novel segmentation techniques. These techniques can be divided into two groups: non-overlapping characters and overlapping characters. For non-overlapping characters or characters with small overlap, such as Arabic numerals, English printed characters, keyboard symbols, and mathematical equations, segmentation is easier. These characters usually have clear spaces between them. Segmentation can be performed using image processing techniques. General methods are Line and Word Segmentation [4-10], Contour Detection [11], and K-means clustering [12]. For overlapping characters or complex spacing, such as in the Thai and Telugu languages, segmentation is difficult. Therefore, deep learning or machine learning methods are used to improve segmentation performance and obtain better results. Popular techniques include U-Net and SegNet [13], Region-Based Convolutional Neural Networks (R-CNN) [14], and You Only Look Once version 5 (YOLOV5) [15]. However, these methods require a large dataset and a long training time to classify data correctly, which may not be suitable in some cases.

This research focuses on an improved segmentation technique for non-overlapping characters in order to apply it to HCR from English printed text, Arabic numerals, and keyboard symbols. These characters are easy to segment. In this case, using machine learning may not be appropriate.

Mamatha H. R. and Srikantamurthy K. [4] proposed a handwritten text segmentation method for Indian languages. Their proposed technique uses projection profiles to separate characters. The experimental results show good accuracy.

Banumathi K. L. and Jagadeesh Chandra A. P. [6] showed that line segmentation has good performance in solving Kannada handwritten problems, where the Kannada language has overlapping and limited spacing. Therefore, word segmentation is not effective for solving Kannada handwritten problems.

*Corresponding author

979-8-3315-5925-0/26/$31.00 ©2026 IEEE

W.D.R. Fernando [7] proposed a four-stage approach for Sinhala handwritten character segmentation, including line segmentation, word segmentation, overlapping character segmentation, and classification.

Poojarini Mitra et al. [9] proposed line, word, and character segmentation for handwritten Bangla text. They emphasize that accurate segmentation is essential for improving Optical Character Recognition (OCR) performance.

Arwa Alghamdi et al. [10] investigated line and character segmentation for Arabic handwritten text using projection profile methods on a historical manuscript. Line segmentation shows good results, while character segmentation performs poorly due to overlapping and language complexity.

Sakshi and Vinay Kukreja [11] proposed using contour detection from OpenCV to segment handwritten mathematical expressions. Their experiments achieved good segmentation accuracy.

Abhishek Jha et al. [12] proposed character segmentation for Indian license plates. They used image preprocessing, K-means clustering, Otsu, and Niblack methods. The experimental results show that K-means clustering obtains good results.

Kunnapat Thipparaphonkul et al. [13] proposed a deep learning–based segmentation algorithm for offline Thai handwritten character recognition. They used U-Net and SegNet models for horizontal and vertical link segmentation. The proposed method achieves high performance.

Kanchan Varpe, Sachin Sakhare, and Rahul Joshi [14] proposed an R-CNN technique to improve handwritten character segmentation for OCR systems. Experimental results show that the proposed method effectively handles overlapping and varied writing styles.

M. Kiran Sastry and S. Aparna [15] proposed a model for the segmentation and classification of Telugu handwritten characters using Adaptive YOLOv5 for segmentation.

From the previous mention, characters (English print type, Arabic numerals, and symbols found on the keyboard) have clear spacing. Therefore, the techniques used for segmentation can be applied easily. It is not necessary to use machine learning or deep learning, which would result in the loss of resources for training data. Hence, this research proposes image processing techniques, including line segmentation to separate lines and K-means to separate characters in each line into individual characters.

However, these techniques have weaknesses. Some characters, when using space-based segmentation, are separated into two characters. For example, the character “i” may be segmented into “l” and “.” In the case of handwritten text, the height of characters is not equal, so character height cannot be used to classify the data. To address this issue, this research proposes the merge technique to combine characters that are close to each other into a single character, in order to improve segmentation performance.

From the previous mention, results from segmentation affect classification. Therefore, good results from segmentation can increase classification performance. To verify it, this research compares results using EasyOCR, which is an impartially benchmarked OCR engine [16, 17], by reading text directly from paper images without segmentation and by reading individual characters after applying the proposed segmentation method.

Experiments were conducted on an individual handwriting dataset collected from 50 participants. The results indicate that the proposed merge technique can improve the segmentation technique for HCR. The proposed technique gets a lower error percentage than other compared similar methods. Moreover, the results from the improved segmented images enhance the performance of classification and obtain better results than the non-segmented images.

The rest of this document is organized as follows. Section 2 explains the background. Section 3 describes related works. Section 4 explains the proposed method. Section 5 shows the experimental results. Finally, Section 6 concludes this research and suggests future work.

II. Background

A. K-means clustering

K-means [12, 18] is a clustering algorithm that assigns n observations into 2 clusters (k=2) for a black-and-white image. In this case, one cluster represents the foreground (character pixels) and the other represents the background pixels. In character segmentation, each pixel is treated as an observation. Initial cluster centers are randomly selected from the image data. Each pixel is then assigned to the nearest cluster center based on distance. After assignment, the cluster centers are updated by calculating the mean value of each cluster. This process is repeated until the cluster centers no longer change. As a result, pixels belonging to the same character are grouped into one cluster.

B. Contour Detection

Contour Detection Segmentation [6] is used to identify the boundaries of objects in an image. The process can be summarized as follows: after preprocessing the image, edge information is extracted to highlight object boundaries. Contours are detected by finding connected boundary points. Each contour represents a character. Small contours are removed based on size or shape. The remaining contours are used to segment individual characters from the image.

C. Projection Profile

Projection Profile [4] is a simple image processing method used for text segmentation. It works by summing pixel values along one direction of the image. The Projection Profiles that are commonly used for text segmentation are the Horizontal Projection Profile and the Vertical Projection Profile. The Horizontal Projection Profile calculates the sum of pixels in each row. Low values indicate empty rows, which represent spaces between text lines. The Vertical Projection Profile calculates the sum of pixels in each column. Low values indicate spaces between words or characters.

D. Line Segmentation

Line Segmentation [7–11, 19] is the process of dividing an image into text lines. For example, if an image has 5 lines, it will be divided into 5 separate images. This is done by scanning the image horizontally at the pixel level, from top to bottom. When the scanning finds a space between text lines, it separates the lines. Line segmentation is popularly used with the horizontal projection profile to find empty rows. These empty rows are considered the spaces between text lines.

E. Word Segmentation

Word segmentation [7–11, 19] is executed by searching for the space between words. This is executed by scanning each line vertically at the pixel level, from left to right Word segmentation can be performed using methods such as the vertical projection profile. Word segmentation must be executed only after line segmentation. It performs well because it uses the distance between words to separate words within each line.

F. EASYOCR

EasyOCR [16, 17] is an open-source OCR tool that reads text from images. It works by first detecting text areas in an image, and then recognizing the characters in those areas using deep learning models. It supports multiple languages and can handle different fonts and layouts. The EasyOCR process can be summarized as follows: First, Text Detection scans the image to find areas that contain text. Then, Text Recognition uses a deep learning model to read and convert the detected text areas into characters. Finally, it outputs the text in a readable form. Experiments in [16] show that EasyOCR can detect text with high accuracy.

III. PROPOSED METHOD

As mentioned previously in this paper, K-means clustering [12] is called the K-means technique, Contour Detection [6] is called the Contour technique, and Line and Word segmentation [11] is called the Word technique. These three techniques use the principle of separating characters based on spaces. English characters and Arabic numerals mostly use spaces to separate characters. Therefore, these techniques perform well for segmentation.

However, some characters cannot be separated using spaces, such as “i” and “j”. They have spaces within a single character. If space is used, one character is often separated into two. For example, the character “i” is separated into “l” and “.”, and the character “j” is also separated into “l” and “.” Moreover, in the case of handwritten text, character size can rarely be used for segmentation.

As shown in Fig. 1, the image has already been processed by the preprocessing step. The character “i” in the rectangle is the original character processed by Contour Detection, Line and Word Segmentation, and K-means clustering, respectively. The segmentation results from Contour Detection, Line and Word segmentation, and K-means clustering are shown in Fig. 2, Fig. 3, and Fig. 4. All three techniques have problems with the character “i”. Since the segmentation results are wrong, the classification results become incorrect as well. In this example, the classified results are “l” and “.”, but the correct result should be “i”.

In addition, other symbols on the keyboard such as : ; ? ! % and = have similar problems. When these symbols are segmented using the previous techniques, they are separated into more than one character, which leads to incorrect classification results. To solve this problem, this research proposes the merge technique.

Based on the experimental results, among the three segmentation techniques, K-means clustering gives the best results. Therefore, this research proposes applying a merge technique after the K-means clustering step. The merge technique must be applied after K-means clustering. Each character has horizontal coordinates (X-axis), as shown in Fig. 5. If the X-axis coordinate of one character is within the boundary of another character, these two characters will be merged. In the example in Fig. 5, the dot character has X-axis points between 598 and 612, and the line character has X-axis points between 599 and 608. Both have points within each other’s boundaries. The line character has a point at 599, which is between 598 and 612, so they should be merged into a single character. The result becomes the character “i”, as shown in Fig. 6. The merge technique can also be applied to characters such as j, :, ;, ?, !, %, and =.

The merge technique can be written in mathematical form as follows. Let two characters be C_1 and C_2. Each character has a horizontal coordinate range on the X-axis, as shown in Eq. (1). For the merging condition, the two characters are merged if the X-axis range of one character overlaps with the X-axis range of the other character. This condition is expressed in Eq. (2).

$$C_1 = \left[X_1^{min}, X_1^{max}\right], C_2 = \left[X_2^{min}, X_2^{max}\right] \quad (1)$$

$$\left[X_1^{min}, X_1^{max}\right] \cap \left[X_2^{min}, X_2^{max}\right] \neq \emptyset \quad (2)$$

The merging result makes the character size larger, which increases the chance of merging with other nearby characters. If characters are written with enough space between them, this technique can greatly improve segmentation performance. However, each merge makes the character larger and may increase the chance of merging with other characters again. Therefore, merging multiple times can increase the chance of obtaining a more complete character. Experiments show that merging twice gives better results than performing the merge only once. For more than two merges, the results are the same as those for two merges. One should not merge many times because it wastes time resources. Hence, this research proposes executing the merge twice to improve segmentation performance.

To show that segmentation results can improve classification performance, this research performs classification using EasyOCR on images that have passed the proposed segmentation step, as shown in Fig. 11. The results are compared with images that have passed only the preprocessing step, as shown in Fig. 8, which were also classified using EasyOCR.



Fig. 1. Exmaple of the character “i” on the image has already been processed the preprocessing step.



Fig. 2. The character “i” is processed by Contour Detection.



Fig. 3. The character “i” is processed by Line and Word Segmentation.



Fig. 4. The character “i” is processed by K-means clustering.

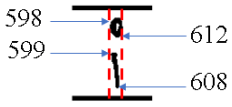


Fig. 5. The character “i” and horizontal coordinates on the image has already been processed the preprocessing step.

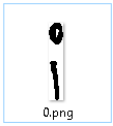


Fig. 6. The character “i” is processed by K-means clustering and the merge technique.

The proposed segmentation technique for solving HCR includes data collection, preprocessing, segmentation using line segmentation, K-means clustering, and merge techniques applied twice, and finally classification using EasyOCR. It is called the merge technique. The proposed technique consists of a total of four steps, summarized as follows:

Step 1: Data collection. Volunteers wrote text in their own handwriting on answer sheets using ballpoint pens with a tip size of at least 0.5 mm and only black or blue ink. The answer sheets contained predefined boxes to keep characters separated, and the volunteers were required to write characters only inside these boxes. The answer sheets were then scanned. Finally, volunteer names and other irrelevant information were cropped out, as shown in Fig. 7. The dataset can be downloaded from [20].

Step 2: Preprocessing. The input image is converted to a grayscale image. Then, it is converted to a binary image using a thresholding method. After the conversion, the watermark background is completely removed. However, in some cases, small dots or noise may remain, so a noise removal process is applied to clean the background and improve image quality. Next, the thickness of the character strokes is increased using the dilation technique, which makes the characters darker and clearer, as shown in Fig. 8.

Step 3: Segmentation. Line segmentation is executed to separate characters into individual lines to increase the accuracy of character segmentation, as shown in Fig. 9. In the next step, each line is processed using K-means clustering to separate characters one by one, as shown in Fig. 10. The results in Fig. 10 show that K-means clustering produces many fragmented components (such as dots). Then, the merge technique is executed. This technique starts by sorting characters from left to right. The current character is compared with the next character. If the X-axis coordinate of one character is within the boundary of the X-axis coordinate of the other character, these two characters are merged. The merge process is executed twice. The final results are shown in Fig. 11, which indicates that the dots can be merged into characters that correctly match the original characters from Fig. 8.

Step 4: Classification by EasyOCR. Each character obtained from the segmentation in Step 3 is classified using EasyOCR.

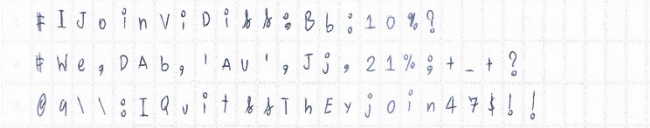


Fig. 7. One part example of the individual's handwritten Characters dataset from volunteers.

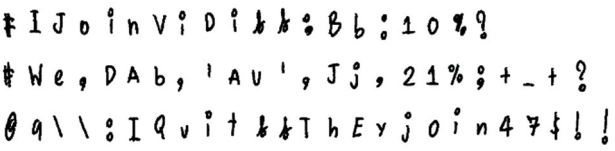


Fig. 8. The image example after the preprocessing step executed.

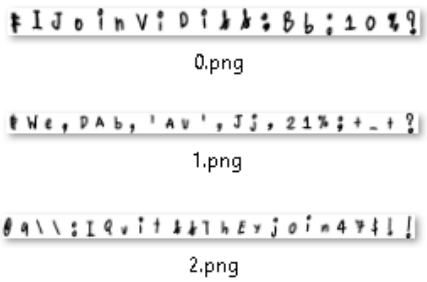


Fig. 9. The image example after Line segmentation executed.



Fig. 10. The image example after K-means clustering executed.

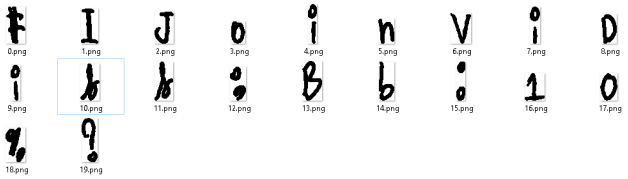


Fig. 11. The image example after merge technique executed.

IV. Experiments and Results

A. Datasets

This research uses one dataset for all experiments. The dataset consists of 50 sets collected from 50 volunteers. Each volunteer wrote characters in their own handwriting on a form designed by the researchers, which is the size of A4 paper. The dataset includes the following characters: digits 0 to 9 with an average of 4 samples per digit, uppercase letters A to Z with an average of 4 samples per character, and lowercase letters a to z with an average of 7 samples per character, except for the characters “i” and “j”. The character “i” has 43 samples and “j” has 14 samples to clearly show the performance of the merge technique and the impact of segmentation problems. In addition, special symbols are included: + @ # $ % ^ & * ( ) [ ] { } - _ = \ | : ; ' ! < > ? . , / with an average of 6 samples per symbol, except for % : ; ' ! ? which have an average of 11 samples each. The scanned papers were processed using a clean, dust-free scanner to reduce interference. The scanned files are saved in “.png” format with a resolution of 2480 × 3507 pixels, and each paper contains 596 characters. Some examples of the dataset are shown in Fig. 7 and it can be downloaded from [20]. As shown in Fig. 7, the designed form

uses faint lines to separate each character, which are visible enough for volunteers to see and write characters inside the boxes. After the preprocessing step, the background is completely removed, so it does not interfere with the segmentation process. The dataset is used to compare the performance of algorithms and to demonstrate the practical applicability of the results. This research was conducted on a personal computer with an Intel i7-14700F CPU running at 2.1 GHz, 16 GB of RAM, and Python 3 as the programming language.

B. The Measures of Algorithm Performance

The performance metrics used in the experiments are described as follows. The Average Recognition Rate (ARR) is calculated as the ratio of correctly identified predictions to the total amount of data across all runs. ARR indicates the classification performance of the algorithm, where a higher ARR value means better classification performance.

The Average Percent Error (APE) is calculated based on the segmentation accuracy. To measure the segmentation performance, the number of correct segmentations is divided by the total number of characters and then subtracted from 100% to find the error, as shown in Eq. (3). APE indicates the segmentation performance of the algorithm, where a lower APE value means better segmentation performance. All segmented images are compared with the correct images. If a character is segmented incorrectly, the count of correct segmentations is not increased. For example, if the segmented text is “IJoin” and the true count is 5, but the segmentation result is “I J o l . n”, the measured correct value is 4 because one character (“i”) was segmented incorrectly.

$$Percent\ Error = \left|\frac{Measured\ Value - True\ Value}{True\ Value}\right| \times 100 \qquad (3)$$

SD, which stands for Standard Deviation, is used to measure the reliability of the segmentation algorithm. A smaller SD value indicates higher reliability of the segmentation results.

C. Experiments and Results

TABLE I. COMPARATIVE APE AND SD OF CONTOURS, WORD, K-MEAN AND THE MERGE TECHNIQUES ON DATASETS.

Technology	APE	SD
Contours [6]	18.52	4.45
Word [11]	10.39	4.29
K-mean [12]	7.28	2.24
Merge (1 times)	3.88	1.84
Merge (2 times) or Proposed Technique	1.13	1.35
Merge (3 times)	1.13	1.35
Merge (4 times)	1.13	1.35

TABLE II. COMPARATIVE ARR OF EASYOCR WITH IMAGES WITHOUT SEGMENTATION AND EASYOCR WITH IMAGES WITH SEGMENTATION ON DATASETS.

Technology	ARR
EasyOCR with Images without Segmentation	5.94
EasyOCR with Images with Segmentation (Proposed Technique)	39.77

The experimental results in Table 1 show that the merge technique (with one merge) has better segmentation performance than the contour, word, and K-means techniques because the APE of the merge technique is lower. Therefore, the merge technique improves segmentation performance for HCR). In addition, the merge technique (with one merge) is more reliable than the contour, word, and K-means techniques because its SD is lower. Therefore, the merge technique also improves reliability for HCR.

From the experimental results, performing two merges has better segmentation performance and reliability than performing only one merge because the APE and SD for two merges are lower. Thus, executing the merge twice improves both performance and reliability. When increasing the number of merges to three or four, the results are the same as those of two merges. Since the APE and SD values are equal, increasing the number of merges beyond two provides no additional benefit and wastes computational resources.

The experimental results in Table 2 show that using EasyOCR with images processed by the proposed segmentation technique yields better classification performance than using images processed only by preprocessing. The ARR of EasyOCR applied to the proposed method is about six times higher than the non-segmented approach. This shows that the segmentation step is crucial for HCR. By using the same EasyOCR classifier for both cases, the results prove that effective segmentation can significantly improve classification performance.

V. CONCLUSION

In this research, we propose a novel technique to enhance segmentation performance for solving HCR by using line segmentation and K-means clustering, respectively, applying the merge technique twice, and then performing classification using EasyOCR. Line segmentation separates characters into rows. It can increase segmentation accuracy because the area to be considered is reduced. K-means clustering separates characters by considering the spaces between them, so characters that are not connected can be separated.

However, this causes problems for some characters, such as "i" and "j", and some symbols, such as "?", "!", "%", because when space is considered, these characters are separated into two or three parts.

To solve this problem, the merge technique is applied to merge these characters back into their original form. Each merge increases the character size and increases the probability of obtaining correct characters. This research experiments with the number of merges to find a suitable value. From the experimental results, two merges are sufficient to solve the problem for HCR. In addition, the experimental results show that the segmentation step is very important for solving HCR and strongly affects the classification step. The performance of the proposed method was evaluated on a personalized handwriting dataset. The experimental results show that the proposed technique outperforms other techniques, such as Contour, Word, and K-means.

In future work, we plan to develop segmentation methods that can solve the problem of overlapping handwritten characters and correctly separate them into individual characters by applying machine learning or deep learning techniques. In addition, we plan to apply this technique to more complex languages or languages with many character forms, such as the Thai language. From the experimental results of this research, it is shown that EasyOCR should not be used in practical applications for handwritten character

classification because the accuracy is low, at only about 40%. In practical use, the accuracy should be higher than 95%, especially for grading exams. Therefore, this researcher plans to develop machine learning methods, such as Support Vector Machines (SVM), to classify English printed letters, Arabic numerals, and symbols.

ACKNOWLEDGMENT

This research was funded by Faculty of Applied Science, King Mongkut's University of Technology North Bangkok, Thailand contract no. 690006.

REFERENCES

[1] Chiabwoot Ratanavilisagul. 2025. Modified Particle Swarm Optimization and Machine Learning for Solving Handwritten English Characters. In Proceedings of the 2024 10th International Conference on Robotics and Artificial Intelligence (ICRAI '24). Association for Computing Machinery, New York, NY, USA, 45–50. https://doi.org/10.1145/3728985.3728997.

[2] C. Ratanavilisagul, "Modified Particle Swarm Optimization for Solving Handwritten Digits," in 2024 4th International Conference on Computer Communication and Information Systems (CCCIS), Phuket, Thailand, 2024, pp. 52-58, doi: 10.1109/CCCIS63483.2024.00017.

[3] C. Ratanavilisagul and S. Kosolsombat, "Modified particle swarm optimization and support vector machine for solving water level measurement," ICIC Express Letters, Part B: Applications, vol. 17, no. 1,pp. 31-39, Jan. 2026.

[4] M. H. R. Mamatha and K. Srikantamurthy, "Morphological operations and projection profiles based segmentation of handwritten Kannada document," International Journal of Applied Information Systems (IJAIS), vol. 4, no. 5, pp. 13–?, Oct. 2012.

[5] O. Yakovchuk, A. Cherneha, D. Zhelezniakov, and V. Zaytsev, "Methods for lines and matrices segmentation in RNN-based online handwriting mathematical expression recognition systems," Proc. 2020 IEEE 3rd Int. Conf. Data Stream Min. Process. DSMP 2020, pp. 255 261, 2020, doi: 10.1109/DSMP47368.2020.9204273

[6] K. L. Banumathi and A. P. Jagadeesh Chandra, "Line and word segmentation of Kannada handwritten text documents using projection profile technique," 2016 International Conference on Electrical, Electronics, Communication, Computer and Optimization Techniques (ICEECCOT), Mysuru, India, 2016, pp. 196-201.

[7] Fernando, Dinal. (2021). An Optimized Approach for Sinhala Handwritten Character Segmentation in Envelopes.

[8] Keerthi Prasad Ganeshaiah and Vinay Hegde, "Handwritten character recognition using optimization based skewed line segmentation method and multi-class support vector machine," International Journal of Advanced Technology and Engineering Exploration, vol. 10, no. 108, pp. 1476-1490, November 2023.

[9] Mitra, Poojarini, et al. "Character Segmentation for Handwritten Bangla Words using Image Processing." American Journal of Electronics & Communication, vol. 1, no. 3, May 2021, pp. 8-11.

[10] Alghamdi, Arwa, et al. "Text Segmentation of Historical Arabic Handwritten Manuscripts Using Projection Profile." 2021 National Computing Colleges Conference (NCCC), 2021, pp. 1-6.

[11] Sakshi and V. Kukreja, "Segmentation and Contour Detection for handwritten mathematical expressions using OpenCV," 2022 International Conference on Decision Aid Sciences and Applications (DASA), Chiangrai, Thailand, 2022, pp. 305-310.

[12] Jha, Abhishek et al. "Character Segmentation on Indian Vehicle License Plate using K-means, OTSU and Niblack algorithm." 2022 Third International Conference on Intelligent Computing Instrumentation and Control Technologies (ICICICT) (2022): 893-897.

[13] Kunnapat Thipparaphonkul, Watchanan Chantapakul, Chayanin Suatap, and Karn Patanukhom. 2020. Thai Handwritten Character Segmentation Based on Deep Learning. In Proceedings of the 2019 2nd Artificial Intelligence and Cloud Computing Conference (AICCC '19). Association for Computing Machinery, New York, NY, USA, 64–74.

[14] Varpe, Kanchan, Sachin Sakhare, and Rahul Joshi. "Enhancing Accuracy of Handwritten Character Segmentation Using Region-Based Convolutional Neural Networks (R-CNN)." 2025 IEEE 2nd International Conference on Cybernetics and Innovation Technologies (ICCIT). IEEE, 2025.

[15] Sastry, M. Kiran, and S. Aparna. "Developing adaptive Yolov5-based Telugu handwritten character segmentation and classification framework using Enhanced Chef-Based Optimization Algorithm and Deep Learning Networks." Computers and Electrical Engineering, vol. 126, 2025, p. 110533.

[16] Chowdhury, Abdulla Nasir et al. "Performance Analysis of Tesseract and EasyOCR for Bangla Optical Character Recognition on the Novel Bangla CrossHair Dataset." 2025 3rd International Conference on Intelligent Systems, Advanced Computing and Communication (ISACC) (2025): 716-721.

[17] Sriharsha, A. V., et al. "Efficient Text Extraction and Summarization using Easyocr and GPT-3." Grenze International Journal of Engineering and Technology, vol. 10, no. 2, Grenze Scientific Society, 2024.

[18] Kanungo, T., Mount, D. M., Netanyahu, N. S., Piatko, C. D., Silverman, R., & Wu, A. Y. (2002). An efficient k-means clustering algorithm: analysis and implementation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(7), 881–892.

[19] N. Dave, "Segmentation methods for hand written character recognition," International journal of signal processing, image processing and pattern recognition, vol. 8, no. 4, pp. 155–164, 2015.

[20] The individual's handwritten segmentation characters dataset. Retrieved from http://chaibwoot.no-ip.org/Dataset/Handwritten _Segmentation_Dataset.zip, Accessed on December 29, 2025.