A Multi-Engine OCR Framework for Accurate Text Extraction from Scanned and Printed Images Using Preprocessing Enhancements

CHANDANA B N[1], Mr SANTHOSH S G[2]
[1]*PG Student, Dept of MCA, JNN College of Engineering, Shivamogga*
[2]*Associate Professor, Dept of MCA, JNNCE, Shivamogga, and Research Scholar, Dept of ICIS, Srinivas University, Mangalore, India. ORCIDID: 0009- 0004-1587-4656*

***Abstract*—In the fields of digitization, document archiving, and automated information retrieval, the extraction of textual data from scanned printed documents continues to be an essential requirement. Traditional OCR systems often face challenges with low-quality or degraded images, where factors such as poor resolution, background noise, and uneven illumination significantly reduce recognition accuracy. The project addresses the challenges by adopting a multi-engine OCR strategy that integrates Tesseract, EasyOCR, and Keras-OCR to enhance reliability and robustness. Scanned inputs are transformed into high-contrast formats using a specialized preprocessing pipeline that includes adaptive thresholding, noise reduction, grayscale conversion, and binarization to better distinguish text from background. This method, in contrast to conventional pipelines, standardizes documents into formats with white text on a black background, making character boundaries more visible. The recognition quality, processing time, and extracted textual fidelity of each image is compared after they are independently processed by the three OCR engines. By leveraging the complementary strengths of traditional and deep learning–based OCR systems, this hybrid framework offers a more adaptable and accurate solution for diverse document types and varying scan qualities, contributing to improved digitization workflows and reliable text recognition.**

***Index Terms*—EasyOCR, Keras-OCR, Optical Character Recognition (OCR), Tesseract, Thresholding**

I. INTRODUCTION

Converting physical documents into machine-readable text is essential for effective information management and retrieval in the digital age. Optical Character Recognition (OCR) plays an important role in extraction process, especially when working with printed documents that have been scanned or degraded over time. Low resolution, background noise, skew, and inconsistent lighting all make it difficult to achieve high OCR accuracy. OCR performance is significantly influenced by preprocessing techniques like adaptive binarization, sharpening, and contrast enhancement, according to recent research. The project addresses the challenges by applying a unified preprocessing pipeline that standardizes images with a black background and white foreground text, thereby improving text visibility and ensuring better edge detection. Three well-known engines Tesseract, EasyOCR, and Keras-OCR—are utilized in order to assess the efficiency of OCR in these circumstances. The system can make use of the strengths of each engine to extract text from printed scanned images in a more reliable and comprehensive manner thanks to independent analysis and comparison of their outputs. The comparative analysis is crucial because each OCR engine comes with its own advantages: Tesseract is a well-known open-source OCR tool that is renowned for its reliability when it comes to handling printed text. Deep learning-powered EasyOCR is more adaptable to various fonts and noisy backgrounds. Architecture of modern neural network are combined in Keras-OCR to improve accuracy of recognition, specially for intricate document layouts. Moreover, the integration of preprocessing ensures that input images are optimized before recognition, significantly reducing recognition errors. Beyond accuracy, this study also explores the processing time taken by each OCR engine, as efficiency is equally important in large-scale document digitization tasks. Visual graphs, such as performance metrics

(accuracy, precision, recall, and F1-score) and time analysis, provide a complete picture of OCR performance under the proposed pipeline and are used to support the analysis. This work addresses the growing demand for reliable digitization in fields like archives, government records, academic research, and corporate document management by contributing to the practical, efficient, and scalable framework development for the text extraction from scanned printed documents.

II. PRIOR ART

Several studies have advanced OCR research by focusing on preprocessing, recognition, and binarization techniques. Nazeem et al. [1] conducted a comparative evaluation of open-source OCR engines such as Tesseract, EasyOCR, and PaddleOCR on Indian and Arabic scripts, reporting high printed-text accuracy but limited analysis of complex layouts. Santhosh et al. [2] developed a smart intrusion prevention system by combining Random Forest classifiers with Flask-based deployment, demonstrating the practical integration of machine learning models into real-time, web-accessible environments. Their work, though centered on network security, highlights the broader adaptability of machine learning for real-time applications. In another study, Santhosh et al. [3] proposed enhanced machine learning methods for robust text classification of bilingual documents, addressing script variability and improving real-time adaptability in multilingual contexts. Kshetry [4] proposed a modified adaptive thresholding method to enhance text clarity through intensity-based segmentation, while Rani et al. [5] introduced a CNN-assisted preprocessing approach for degraded documents, which significantly improved binarization performance, especially in historical manuscripts. Foundational work by Gatos et al. [6] on adaptive document binarization has been widely applied, though it remains disconnected from modern deep learning advancements. Similarly, Su et al. [7] combined local and global thresholding to improve text visibility in historical documents but the results are not compared with contemporary OCR engines like EasyOCR. Ayyalasomayajula [8] leveraged deep neural networks (FCN and PD-Net) for handling severe document degradation, focusing on binarization rather than direct OCR accuracy. Jain et al. [9] provided a comprehensive overview of document layout analysis, essential for structured OCR, though not specifically aligned with engines such as Keras-OCR or PaddleOCR. Huang et al.
[10] introduced DeepErase, a weakly supervised technique for ink and noise removal, which improved OCR accuracy though its generalization to modern printed content remained limited. Beyond document-focused approaches, Mishra et al. [11] explored bottom-up and top-down cues for scene text recognition, while Liao et al.
[12] proposed TextBoxes, a fast single-network detector for text localization. Tian et al. [13] introduced a Connectionist Text Proposal Network for detecting text in natural images, and Gupta et al. [14] leveraged synthetic data to enhance training for text localization tasks. Zhan et al. [15] presented ESIR, an end-to-end recognition framework using iterative image rectification to handle distorted scene text, whereas He et al. [16] achieved accurate text localization through a cascaded convolutional text network, further advancing OCR performance in natural scenes.

III. PROPOSED METHODOLOGY

The proposed system is implemented for meaningful text extraction from printed or scanned images—especially those found in real-world complex scenarios like advertisement boards—using multiple OCR engines to ensure robust performance. The core idea is to preprocess the input image to enhance textual features, and then feed the cleaned version into different OCR systems (Tesseract, EasyOCR, KerasOCR) to compare results in form of confidence and accuracy. By integrating preprocessing techniques like grayscale conversion, binarization, and sharpening, the method aims to reduce background noise, increase contrast, and isolate text from visual clutter. These enhancements improve the readability and recognition rate of OCR systems, especially for skewed or low-quality scans. The use of a Streamlit-based interface allows for intuitive user interaction, visual comparison of OCR results, and presentation of confidence scores. This approach aims to offer a generalized pipeline that can adapt to various OCR engines and image types while highlighting the importance of preprocessing in

improving recognition accuracy.

3.1 Dataset Collection

The dataset used in this project was mostly made up specifically to closely match real-world situations in which OCR systems are commonly used. It is made up of a diverse assortment of images with printed and scanned text from public banners, street signs, advertisement boards, and other similar sources. The pictures were chosen carefully to show a range of difficulties, like backgrounds with a lot of complexity or visual noise, different text orientations, and different resolutions. Because of this diversity, the OCR engines are tested in real-world situations rather than controlled lab environments. While inspiration was drawn from established open datasets such as ICDAR 2015 and COCO-Text, which are well-known benchmarks in the OCR domain, the majority of the dataset was manually collected and curated to align with the project's focus on real-world use cases. This method ensures that the evaluation is based on actual conditions, making the results more applicable to everyday tasks like document digitization and text recognition.

Fig 1. Dataset Collection

3.2.1 Image Upload

The Streamlit interface, which serves as the entry point for the OCR system, is used by users to initiate the process by uploading a scanned or printed image. The image is temporarily stored in a designated folder after it has been uploaded to preserve the raw version for subsequent processing. The platform is flexible enough to handle a variety of document types because it supports multiple formats like PNG, JPG, and JPEG. The system starts a preprocessing pipeline right after the upload to fix common problems with scanned documents like noise, low contrast, and uneven illumination. In order to improve recognition, this preparation ensures that the OCR engines receive standardized and clean input. Thus, the stage of uploading images not only kicks off the workflow, but it also lays the groundwork for text extraction that is accurate and dependable.

1.2.2 Image Preprocessing

Preprocessing transforms the raw scanned image into a cleaner, more structured format that enhances the ability of OCR systems to accurately recognize text. This stage is critical because scanned documents often suffer from issues such as background noise, faded ink, uneven illumination, or distortions introduced during the scanning process. Techniques commonly applied include grayscale conversion, which reduces unnecessary color details and highlights the textual regions; adaptive thresholding or binarization, which improves contrast by separating text from the background; and sharpening filters, which emphasize character edges to make them more distinguishable. Noise reduction methods, such as morphological operations or median filtering, help eliminate speckles and irregularities that could otherwise be mistaken as text. Additionally, resizing and normalization ensure that characters maintain a consistent scale across the document, improving the recognition accuracy of OCR engines. In some cases, to create black background and white text inversion is used, which further enhances character visibility. Together, these preprocessing steps create a high-quality input that bridges the gap between raw scanned images and reliable text extraction.

Grayscale Conversion: by converting the image to shades of grey, it reduces computational complexity and removes color distraction. This formula makes sure that the grayscale image that is converted is very close to how bright the original-colored image looks to people.

$$Gray = 0.299 \times R + 0.587 \times G + 0.114 \times B \quad \textbf{(1)}$$

Binarization, such as Otsu's thresholding, transforms grayscale images into a binary black-and-white format, improving the contrast between text and background to facilitate more accurate text extraction.

Formula:

Otsu's Method selects a threshold t that minimizes

intra- class variance:

$$\sigma^2(t) = \omega 1(t)\sigma^2(t) + \omega 2(t)\sigma^2(t) \quad \textbf{(2)}$$

where ω is class probability and σ^2 is class variance.

Sharpening: Enhances text edges using a kernel-based convolution filter.

$$\text{Sharpened Image} = \text{Original} + \alpha(\text{Original} - \text{Blurred}) \quad \textbf{(3)}$$

where α is a sharpening factor.

These preprocessing steps help clean noisy backgrounds and improve contrast, which boosts OCR engine performance.

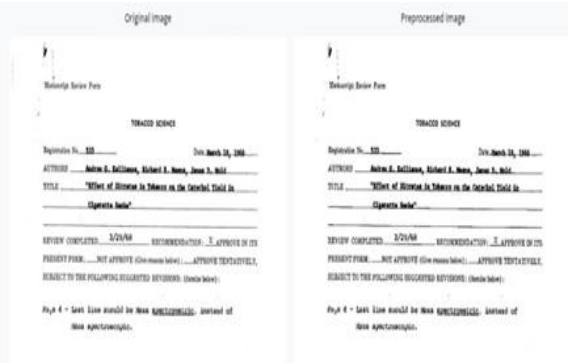

Fig 2. Image Conversion Original into Preprocessed

3.2.3 OCR Text Extraction

Tesseract OCR is a rule-based engine designed for high- quality, clean printed documents, but it is less effective on backgrounds that are noisy or complex. EasyOCR, a deep learning–based framework, supports 80+ languages and diverse fonts, handling variations in alignment, orientation, and image quality. For precise text detection and recognition in a variety of fonts and intricate layouts, Keras-OCR makes use of a pipeline with convolutional and recurrent layers. Combined, these OCR engines offer complementary strengths for robust text extraction across diverse document types and image conditions.

3.2.4 Confidence Score Evaluation

The system collects and displays confidence scores for each OCR output (if provided by the engine), helping compare accuracy levels across tools. These scores reflect the engine's certainty about the recognized text.

$$Average\ Confidence = \sum_{i=1}^{n} c_i \quad \textbf{(4)}$$

Ci= Confidence score of the ith detected text segment

n = Total number of valid detected text segments

System Design

This project's system design incorporates preprocessing, text recognition, and result comparison within a multi- engine OCR framework. To make the text clearer, uploaded scanned or printed images first undergo enhancement methods like resizing, grayscale conversion, and binarization. The images which are processed are then passed to three OCR engines Tesseract, EasyOCR, and KerasOCR for independent extraction. Finally, the outputs, with their accuracy scores and processing times, are compared and visualized, enabling users to evaluate the best-performing engine for their use case.

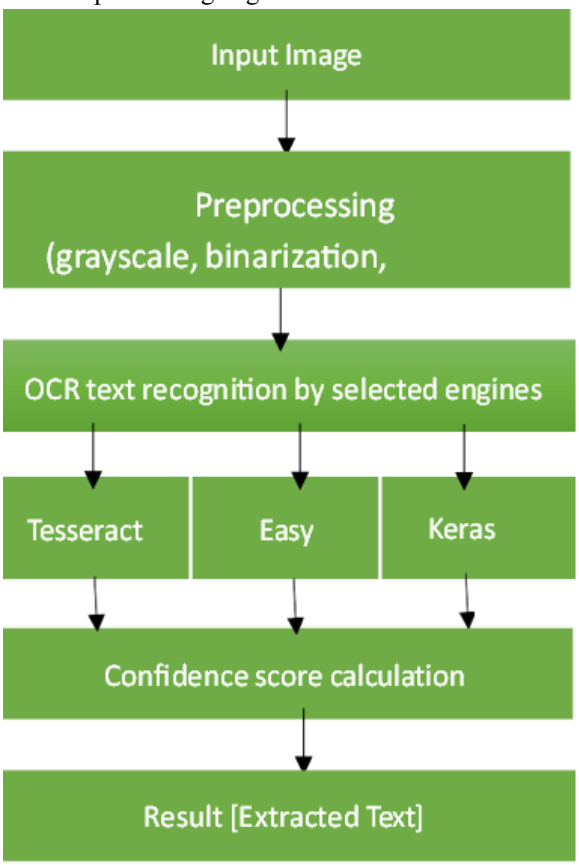

Fig 4. System Design Flow

The OCR system's overall procedure is depicted in this flowchart. To improve text clarity, the captured image is users first subjected to preprocessing steps like grayscale conversion, binarization, and sharpening. For independent text recognition, the enhanced image is then processed by three OCR

engines—Tesseract, EasyOCR, and KerasOCR. The extracted text is then presented as the final result for user evaluation after confidence scores are calculated.

3.2.5 Result Display and Comparison

Both the captured and pre-processed versions of the image are presented side by side within the interface, allowing Users to clearly observe the improvements introduced by preprocessing. Users are provided with a quantifiable measure of accuracy thanks to the system's inclusion of the confidence score for each OCR engine in addition to the extracted text. This dual presentation makes it possible to gain a deeper comprehension of how preprocessing affects recognition quality, particularly in images that are blurry or noisy. Tesseract, EasyOCR, and Keras-OCR outputs can be compared and contrasted for clarity, character spacing, and text completeness. The application enables an option for user to select the OCR engine that is more convenient for their particular use case by combining visual inspection with analytical metrics. This enables users to make well-informed choices regarding the engine's speed, precision, or adaptability to various document qualities. This side-by-side comparison thus creates a transparent and practical evaluation environment for effective OCR performance assessment.

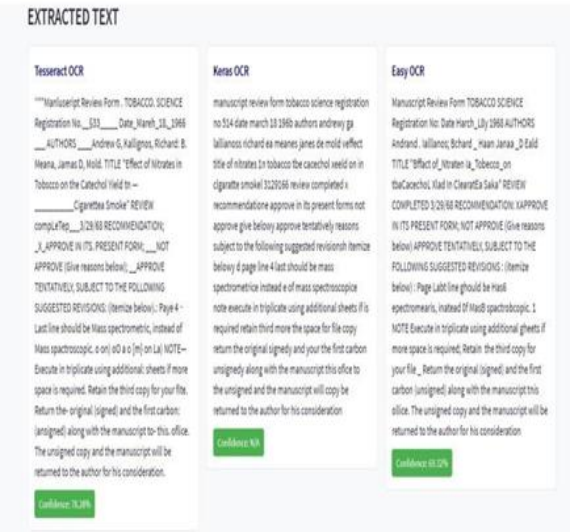

Fig3. Output Display of Extracted Text

This image presents a comparative analysis of text extraction results obtained from three different OCR engines: Tesseract OCR, Keras OCR, and EasyOCR. After processing the same input image, the textual output produced by the respective engine is displayed in each section. Where confidence scores are available, they are displayed alongside the extracted text, indicating the engine's confidence in the recognition accuracy. For example, Tesseract OCR reports a confidence score of 78.79%, while EasyOCR achieves a higher confidence score of 82.36%. Keras OCR, in this instance, does not provide a confidence metric. The three tools' formatting, word segmentation, and recognition accuracy all differ, highlighting differences in how each engine interprets character interpretation, spacing, and punctuation. For determining which OCR engine performs best for particular document types and conditions, this side-by- side display is helpful.

IV. RESULT AND DISCUSSION

Based on analysis of recent research, integrating preprocessing techniques like binarization and sharpening enhances OCR accuracy significantly, especially for degraded scanned documents. Comparative studies indicate that combining traditional OCR (Tesseract) with deep learning-based models (EasyOCR, Keras-OCR) yields improved recognition across diverse print styles and layouts. The snapshots in this study visually demonstrate the transformation of a scanned printed image into machine-readable text using OCR techniques. The original image, often affected by scanning noise, low contrast, or background interference, presents a typical challenge for standard OCR systems. To address this, the image undergoes preprocessing, where techniques like grayscale conversion, binarization, and sharpening are applied. This converts the image into a format with a dark background and clearly defined white text, making character edges more distinguishable and enhancing contrast. The resulting pre-processed image improves the performance of OCR engines. The final snapshot presents the extracted text generated by Tesseract, EasyOCR, and Keras-OCR. Each engine interprets the enhanced image to produce textual output, confirming that preprocessing plays a major role in recognition accuracy. By comparing the snapshots, it becomes obvious that the preprocessing pipeline bridges the gap between raw scanned input and reliable text extraction in real-world scenarios.

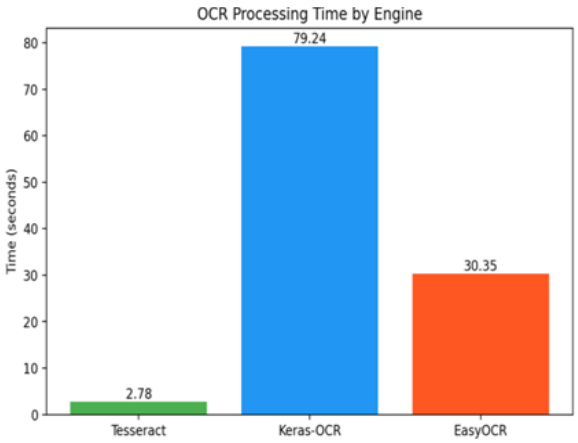

Fig 5. OCR Processing Time by Engine

The bar graph illustrates the comparative processing times of the three OCR engines—Tesseract, Keras-OCR, and EasyOCR—when applied to the uploaded image. Tesseract demonstrates the fastest performance, completing text extraction in approximately 2.78 seconds, highlighting its efficiency for quick recognition tasks. EasyOCR requires more processing time, around 30.35 seconds, reflecting its more complex recognition pipeline that balances accuracy and speed. Keras-OCR, while powerful in handling diverse fonts and orientations, takes the longest, with about 79.24 seconds, indicating a higher computational demand due to its deep learning-based architecture. This analysis emphasizes the trade-off between execution time and model complexity, showing that while Tesseract is optimal for rapid tasks, EasyOCR and Keras-OCR may be preferable when handling more challenging image conditions where accuracy is critical despite longer processing times.

V. CONCLUSIONS

In conclusion, text extraction from scanned printed documents is significantly more reliable and precise when traditional and deep learning-based OCR models— Tesseract, EasyOCR, and Keras-OCR—are combined with a robust preprocessing pipeline. The preprocessing techniques like grayscale conversion, adaptive binarization, and sharpening play a vital role in increasing contrast, decreasing noise, and making character boundaries more distinct, which in turn makes it easier to recognize characters with greater precision. EasyOCR and Keras-OCR outperform Tesseract in comparison when it comes to handling various font styles, varying lighting, and intricate page layouts, while Tesseract performs well with text that is clean and structured. A hybrid OCR framework with adaptive preprocessing provides a comprehensive and practical solution for extracting high- quality text from degraded or low-quality scans, as demonstrated by this project, which confirms previous research's findings. The system improves recognition accuracy and robustness for real-world document digitization and retrieval of information from applications by merging the strengths of multiple OCR engines.

VI. FUTURE WORK

There are several advanced methods which can be explored to further improve the performance and flexibility of text extraction from print-scanned images. Adaptive thresholding and contrast-limited histogram equalization can significantly improve image clarity by balancing lighting variations and improving contrast in low-quality scans, making preprocessing is one of the important stages. The acceptance of cutting-edge deep learning architectures like transformer-based OCR models and Vision-Language frameworks, which are capable of comprehending contextual semantics and handling complex document layouts more effectively than traditional methods, holds enormous potential beyond preprocessing. A noise detection and quality assessment module can also be added to automatically analyses the input image and dynamically adjust preprocessing parameters like denoising strength, sharpening filters, and thresholding techniques to give each document the best enhancement. On the recognition side, an ensemble framework that combines conventional engines like Tesseract with cutting-edge models like EasyOCR, Keras- OCR, or even transformer-based OCR can help take advantage of their respective strengths. By weighting engine outputs according to their confidence levels, this can be improved to produce a more accurate and context- aware final text output. Incorporating natural language processing (NLP) tools like grammar checkers, domain- specific dictionaries, and context-based error correction to refine the recognized text can also improve post-processing methods, particularly in situations where

ambiguous characters or degraded fonts are present. Adding script-specific preprocessing and multilingual OCR support can also make the system more adaptable to a variety of real-world applications. Together, these enhancements not only promise a substantial improvement in recognition accuracy but also pave the way for a more scalable, intelligent, and reliable OCR system capable of handling the diverse challenges posed by real-world document digitization and information retrieval tasks.

REFERENCES

[1] Long, S., Ruan, J., He, X., & Huang, L. (2021). *Scene Text Detection and Recognition: The Deep Learning Era*. ACM Computing Surveys, 54(6), 1–35.

[2] S. G. Santhosh, et al, "Enhancing Machine Learning Methods for Robust Real-Time Text Classification of Bilingual Documents," International Journal of Innovative Research in Technology (IJIRT), vol. 12, no. 3, pp. 42–48, Aug. 2025, ISSN: 2349-6002.

[3] S. G. Santhosh, et al, "Smart Intrusion Prevention System Using Integrated Machine Learning Models (Random Forest Classifier with Flask)," International Journal of Innovative Research in Technology (IJIRT), vol. 12, no. 3, pp. 2872–2877, Aug. 2025, ISSN: 2349-6002.

[4] Sharma, R., & Chaurasia, V. (2020). *A Review on Optical Character Recognition and Its Applications*. Procedia Computer Science, 167, 1160–1167.

[5] Smith, R. (2007). *An Overview of the Tesseract OCR Engine*. In Proceedings of the Ninth International Conference on Document Analysis and Recognition (ICDAR), 629–633.

[6] Jain, P., & Ghafoor, K. (2021). *A Comparative Study of OCR Tools for Text Extraction from Images*. Journal of King Saud University - Computer and Information Sciences.

[7] Borisyuk, F., Gordo, A., & Sivic, J. (2018). *Rosetta: Large Scale System for Text Detection and Recognition in Images*. Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining.

[8] Singh, A., & Roy, P. (2021). *Text Detection and Recognition in Natural Scene Images Using Deep Learning Techniques: A Review*. Multimedia Tools and Applications, 80(10), 15691–15724.

[9] Busta, M., Neumann, L., & Matas, J. (2017). *Deep TextSpotter: An End-to-End Trainable Scene Text Localization and Recognition Framework*. In ICCV.

[10] Mishra, A., Alahari, K., & Jawahar, C. V. (2012). *Top-down and Bottom-up Cues for Scene Text Recognition*. In CVPR.

[11] Liao, M., Shi, B., Bai, X., Wang, X., & Liu, W. (2017). *TextBoxes: A Fast Text Detector with a Single Deep Neural Network*. AAAI Conference on Artificial Intelligence.

[12] Tian, Z., Wang, W., Huang, T., & He, C. (2016). *Detecting Text in Natural Image with Connectionist Text Proposal Network*. ECCV.

[13] Gupta, A., Vedaldi, A., & Zisserman, A. (2016). *Synthetic Data for Text Localisation in Natural Images*. CVPR.

[14] Zhan, F., Lu, S., & Zhu, H. (2019). *Esir: End-to-end Scene Text Recognition via Iterative Image Rectification*. CVPR.

[15] He, T., Huang, W., Qiao, Y., & Yao, C. (2016). *Accurate Text Localization in Natural Images with Cascaded Convolutional Text Network*. ACCV.

[16] Shi, B., Bai, X., & Yao, C. (2017). *An End-to-End Trainable Neural Network for Image-Based Sequence Recognition and Its Application to Scene Text Recognition*. IEEE TPAMI.

[17] Ma, J., Shao, W., Ye, H., Wang, L., Wang, H., Zheng, Y., & Xue, X. (2018). *Arbitrary-Oriented Scene Text Detection via Rotation Proposals*. IEEE Transactions on Multimedia.

[18] Liu, Y., Jin, L., Zhang, S., & Zhang, C. (2018). *DenseRAN for Offline Handwritten Chinese Character Recognition*. AAAI.

[19] Wan, Z., Wang, K., Huang, Y., & Guo, W. (2020). *Text-Image Matching Based on Self-Attention and Dual Encoding*. IEEE Access.

[20] You, S., Yang, W., Liu, F., & Shan, Y. (2020). *Scene Text Detection and Recognition: The Deep Learning Era*. Pattern Recognition, 110, 107304.