Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 8, August : 2024

Text Information Extraction from Images

Samit Arun Ingole[1], Shitalkumar A Jain[1]
[1]MIT Academy of Engineering, Alandi, Pune, India
samit.ingole@mitaoe.ac.in, sajain@mitaoe.ac.in

Abstract: Text extraction from images is a fundamental task in document digitization, automated data processing, and real-time information retrieval. Various techniques have been explored for this purpose, including deep learning-based models, traditional Optical Character Recognition (OCR) engines, and hybrid approaches that integrate machine learning with image preprocessing. This study offers a thorough analysis of several research projects emphasizing several text extraction methods. Though they need great computer power, deep learning techniques like Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM), and Transformers have demonstrated extraordinary accuracy. Conversely, while they may have difficulty with handwritten writing and complicated backdrops, OCR-based approaches like Tesseract OCR, OpenCV, and image preprocessing provide lightweight and effective alternatives. Hybrid methods, especially those integrating Tesseract OCR with OpenCV and Pillow, have greatly enhanced text recognition accuracy and efficiency. This study looks at the pros and cons of each method and finds that using Tesseract-OCR, OpenCV-Python, and Pillow together is the best way to quickly and effectively extract text in different areas. The findings of this paper serve as a foundation for future research in developing more robust, scalable, and accurate text extraction systems.

Keywords: Convolutional Neural Networks, Tesseract, Optical Character Recognition, Long Short-Term Memory, Text Extraction.

1. INTRODUCTION

Document image analysis is a crucial subfield of computer vision and natural language processing, which is successfully used in different areas like medicine, commuting, finance, and policies. Due to the emergent role of digital documents, optical character recognition to read text from paper documents and electronic media has assumed a very important role to play in streamlining the process, easing the flow of documents, and eliminating or somehow reducing human errors that may arise while keying documents. OCR has been a revolutionary topic in this respect, as it has facilitated computers to read text from scanned images and photographs as well as scenes in a way that can be processed. However, some of the challenges faced in the text extraction include the poor signal and image quality, a different type of font size and style, the intensity of the light under which the document was taken, poor quality of the scanned document, different writing styles, and complicated background images. In recent years, various efforts have been made to advance text extraction procedures, such as the use of the normal OCR technique, deep learning techniques, and other techniques based on the fusion of different approaches [1]. Of these, Tesseract OCR has received much attention due to its effectiveness, ability to work on different languages, and document structure and content recognition capabilities. However, because of its low quality of images, noise, and difficult structure of the text it produces, it has a very low recognition rate. That is why there is the usage of filtering, such as converting the images to grayscale, noise removal, contrast enhancement, and many others using OpenCV and Pillow to enhance the images for text recognition, which in turn improves the OCR[2].

I. TRADITIONAL OCR VS. DEEP LEARNING-BASED APPROACHES

Rule-based techniques and feature extraction techniques are both traditional techniques that have been widely used in optical character recognition, where the characters to be recognized are first tuned according to certain rules and character templates, which, in combination with statistical models, are used to recognize the text characters. These methods work quite well for the documents with non-distorted printing and clean, black fonts but fail for handwritten data, low-contrast images, and cursive text. Tesseract OCR, for instance, operates using adaptive thresholding and pattern recognition algorithms but often requires extensive preprocessing to handle variations in input images. In contrast, text extraction models that use deep learning, like CNNs, Recurrent Neural Networks (RNNs), and Transformers, have shown great success in reading text from complicated images. These models learn patterns from big data and are capable of generalizing the recognition to different styles, fonts, and directions of the handwriting. For instance, to establish both spatial and temporal features in text pictures, CNN-BiLSTM structures have been exploited to enhance the recognition rate. However, deep learning approaches need computation power, an abundant amount of labeled data, and a long training time, which makes them unsuitable for lightweight as well as real-time applications.

II. THE ROLE OF HYBRID APPROACHES

Taking into consideration the drawbacks of common OCR and modern deep learning models, scientists search for their combination with machine learning and image recognition algorithms. Hybrid models utilize the fast speed of Tesseract, OpenCV, and Pillow for pre-processing and segregated deep learning segments for improving the rate of recognition. It also enables more effective recognition

of noisy images, different lighting conditions, and writing styles of handwritten text. Tesseract OCR in combination with OpenCV for image preparation and Pillow for image post-processing is more favorable for recognition in such fields as medical prescriptions, license plate number recognition, the use of digital receipts, and digitizing historical papers.

III. Scope and Contribution of this Review

The process of extracting text from images has been researched in detail, and there are various ways of doing that using deep learning models, traditional OCR standalone, or utilizing image processing with OCR engines. This section presents several papers that are related to the mentioned methodologies and their advantages and limitations. We categorize the literature into three broad sections. (A) Deep learning-based approaches; (B) traditional OCR and image processing methods; and (C) hybrid approaches that integrate OCR with machine learning and image processing techniques. Each subsection highlights key findings from multiple studies, systematically comparing different techniques to evaluate their strengths and weaknesses. Thus, by the end of the given review, it is possible to conclude that Tesseract OCR combined with OpenCV and Pillow offers the best results in terms of performance and accuracy for processing and extracting text from real-world images.

2. LITERATURE REVIEW

Opening text from images as a problem has been investigated for many years, and as such, there are lots of solutions in the context of deep learning approaches, optical character recognition, and hybrid models that combine the image processing with OCR engines. This section provides an analysis of such papers, which are focused on different methodologies, their benefits, and drawbacks. The entire research has been divided into three main sections: (A) Deep learning-based papers, (B) optical character recognition and image processing work, and (C) papers that use both OCR and machine learning and image processing techniques. All the subsections include a review of several studies wherein the authors make direct comparisons between various approaches in order to expose their advantages and limitations. Therefore, Tesseract OCR, OpenCV, and Pillow show the best versions in terms of both time and accuracy that can be used for the real-world application software.

A. Deep Learning-Based Approaches for Text Extraction:

Deep learning models have revolutionized the text extraction process by enhancing the accuracy in terms of accurately extracting texts that are noisy, handwritten, and even complex, multilingual texts. They apply convolutional neural networks, recurrent neural networks, long short-term memory networks, transformers, and other deep learning network structures to recognize and establish relationships in the textual images. The CNN- Bi-Directional LSTM (BiLSTM) -based text extraction model [1] employs CNNs for feature extraction and BiLSTM for sequential text modeling. It is also quite possible to identify both handwritten and printed texts, respectively, recognition of which reached 88.58% and 90.8% for a set of samples. The authors enhance context understanding through BiLSTM, where, according to them, CNN captures spatial information while BiLSTM captures sequence information in text. However, it demands more computational power and a large training set that makes the real-time application of systems a great issue. The authors identify a two-step process involving the use of a Convolutional RNN (CRNN) model and apply the model in the process of extracting medical lab reports by localizing the text and then reading it off. The sensitivity of the Chinese character recognition depending on the CRNN framework is 99.5%, including English and Chinese characters. This is quite useful for handling medical documents but comes with the downside of needing a large dataset for a model and a computing asset to run on; it is, therefore, not nearly feasible to run in a real-time health care system. One of the recent state-of-the-art models with Transformer architecture aims at recognizing text in multiple scales [3], which utilizes self-attention to capture text in complicated environments. While working with CNN-based models has some drawbacks, such as not analyzing the long-range dependencies, the transformers seem to perform well when it comes to text localization and recognition. The results show that the model is very effective for multiple languages for text detection, but the program has a heavy computational cost and is therefore not very suitable for mobile devices. Another important improvement comes from contrastive learning for image-text retrieval [4], where CLIP-based models make text detection in images better by using improved methods to measure similarity. By modifying how the system understands image-text relationships, accuracy improved by 5.32% in classification tasks, and retrieval precision increased by 45.2%. While this method benefits search engines and automated captioning, it does not focus on direct text extraction from images. Despite their impressive performance, deep learning-based methods require extensive computational power, specialized hardware (such as GPUs), and large datasets for training. These usually take longer times than traditional OCR methods to draw their inferences making them useless in real-time applications.

B. Traditional Approaches OCR and Image Processing-Based Approaches

As the name implies, Tesseract OCR is one of the most versatile and accurate OCR engines designed to process structured documents. Such techniques entail operations such as binarization, thresholding, and noise removal to enhance the quality of text, which is helpful in OCR. The use of OCR to digitize prescriptions presents research on how prescription prints, handwritten or typed, can be scanned using the Tesseract OCR with adaptive threshold settings. The system achieves 98% accuracy by implementing image resizing and contrast enhancement, which guarantees the correct extraction of medicine names and dosages. However, handwritten text with illegible characters remains a challenge. In Automatic Number Plate Recognition (ANPR) [5], the authors use grayscale conversion, morphological operations, and Tesseract OCR to extract alphanumeric characters from vehicle

license plates. The proposed model ideally detects 83.3% character accuracy under ideal lighting but fails in the case of different styles of font, reflections, and low picture quality through motion blur in real-world examination. The paper on digitizing Javanese scripts [6] explains the difficulties in the non-Latin scripts for OCR. According to the bounding box modification and training of Tesseract on a given dataset, the authors are able to obtain an accuracy of 97.5%. However, the model does not perform well in the case of handwritten Javanese text, thus signifying the fact that there is room for improvement in terms of training the OCR for the low-resource language. A real-time license plate recognition system [7] is incorporating OpenCV with Tesseract OCR for better characterization of characters by applying Gaussian Blur, Adaptive Threshold, and Contour Detection. Although this improves the identification of characters and shapes, it decreases the speed for the identification of small texts and low-quality images. In the case of receipt text extraction, the work done in [8] uses Tesseract OCR along with image splitting, which provides better text separation in lengthy receipts. This method works best in identifying dates, prices, and taxes, while it has some issues when it comes to identifying faded ink, handwritten notes, and any sort of non-standard fonts. The study on getting text from scanned images [9] presents a system that finds information using Tesseract OCR and OpenCV to detect shapes. While this method efficiently locates words in scanned documents, highly distorted or low-contrast images remain problematic. While traditional OCR methods provide lightweight and efficient solutions, they are highly dependent on image quality and struggle with handwriting, curved text, and complex document layouts.

C. Hybrid Approaches: Integrating OCR with Machine Learning and Image Processing

Enhancements of the hybrid methods over OCR enhance the established engine use of machine learning, object detection, and real preprocessing procedures. These approaches are intended to address three factors, namely accuracy, efficiency, and the real-time nature of the implementation. Another work with YOLOv3-based OCR uses an object detection-based approach in which YOLOv3 detects text regions that are then passed to Tesseract OCR. It helps to overcome the false positive issue and also makes it applicable for background profiles, which usually have a lot of noise around the object of interest. Nevertheless, YOLOv3 makes the better improvement but still needs domain-specific fine-tuning in order to advance the better and reasonable lead. In the study that deals with text extraction using Gamma Correction [10], they employ the techniques of adaptive thresholding and the Gray Level Co-occurrence Matrix (GLCM) analysis before OCR. This approach reduces execution time by 67% while maintaining high accuracy, making it suitable for embedded systems. A text extraction model that uses Radial Wavelet Transform (RWT) improves the process of getting text from low-contrast images by using wavelet entropy methods. However, as the computational cost will be high for real-time applications, it is more effective for weak signal detection. The comic text extraction system [11] uses a method based on the Connected Component Labeling (CCL) approach with 94.82% accuracy. As for the application of this approach to digital comics, it is generally helpful as it can only process texts that are laid horizontally and is useless when there are overdrawing characters. An Android-based system using Tesseract and LSTM breakthrough proves to be 85-98 % efficient through LSTM-based character distinction. Nevertheless, due to the complicated background, it is still demanding to decipher handwritten text and images.

D. Summary of Key Findings

However, the deep learning-based models give high accuracy but require high computational resources and a large dataset. Traditional OCR methods like Tesseract are efficient but need preprocessing to handle noise and complex layouts. Combining OCR with machine learning and preprocessing tools like Tesseract, OpenCV, and Pillow greatly improves accuracy and reliability. From this analysis, the best choice for extracting text in real-life situations is a hybrid method that uses Tesseract OCR, OpenCV-Python, and Pillow because it works well, can grow with needs, and adapts to different text recognition tasks [12-16].

3. MATHEMATICAL MODEL

However, notwithstanding the development in the text extraction techniques, a few challenges and limitations still exist in all these techniques. These limitations fall under the following categories we broadly can categorize:

A. Limitations of Deep Learning-Based Text Extraction Approaches:

In the earlier years, deep learning models like CNN BiLSTM, CRNNs, and others have shown very high accuracy in text recognition. But there are the following factors that hamper their applicability:

Dependency on Large-Labeled Datasets

High Computational Requirements

Such models based on deep learning are not usable in real-time applications on low-power devices like mobile phones and embedded systems; they need high-end GPUs and TPUs. Operating the large-scale deep learning models requires hardware and infrastructure, which considerably raise the operational costs.

Unlike rule-based OCR systems, deep learning models require massive annotated datasets for training, such as IAM for handwriting and Synth Text for printed text. The availability of diverse, high-quality datasets is a significant challenge, particularly for low-resource languages and specialized domains (e.g., historical scripts and niche medical documents).

Inconsistent Performance on Handwritten Text and Unstructured Documents

Yet printed text is what deep learning models do best at, while they tend to mangle the handwritten text, stylized fonts, and messed up characters. This decreases the recognition accuracy of real world scenarios due to the fact that many state of art models struggle with multi line, multi oriented, and overlapping text structures.

High Latency in Real-Time Applications
In fact, transformer-based models, while being quite accurate, offer extremely slow inference, thus not being suitable for real-time use cases such as license plate recognition or USS document scanners. Compared with the original OCR techniques, deep learning models usually entail iterative and iterative improvement, which prolongs the time of text processing.

B. Limitations of Traditional OCR and Image Processing Techniques:
For example, Tesseract OCR engines are preferred for their performance as well as for their not being larger in size than other types. Nevertheless, these methods have some drawbacks:

Poor performance on noisy and low-resolution images. This is because OCR engines have to analyze the quality of an image in terms of the extent of light in an image and the sharpness of the characters. However, the most critical challenges seen in the testing process include blurred, low-resolution, and shadowed images, which resulted in wrong characters being extracted.

Difficulties of Handwritten and Cursive Text Recognition Tesseract OCR fails to work with difficulties in the cursive, slanting, or joined-up writings, and that is why it could not perform well with the handwritten texts. They, therefore, reduce the readability of the handwritten texts, as the engine is meant for printed texts only.

Uses of Preprocessing in Traditional OCR Several OCR approaches need prior preprocessing of the image documents, which may involve binarization, thresholding, or erosion to remove noise, for instance. Shiny and complex backgrounds also hinder the OCR performance; therefore, different kinds of filters are required in this case. Failure to perform well for multilingual and multi-script text One of the language supports of the Tesseract is the ability to handle multiple languages, but it results in confusion when it has multiple different languages on a single document. For such types of writing systems like Javanese, Arabic, and Chinese script, the trained model for OCR is not always available.

Problems with non-standard font Non-standard font types and sizes remain a huge challenge for OCR engines to read all the contents that, too, are having layouts and formats that are complex with other objects like tables, images, etc.
Thus, such areas as text extraction from the digital comics, using decorative fonts, or appreciable lettering is still an unworthy obstacle.

C. Limitations of Hybrid Approaches Integrating OCR with Image Processing and Machine Learning:
A new approach to overcome these issues is to use a hybrid model where OCR is used in conjunction with image processing techniques such as OpenCV and Pillow and object detection algorithms such as YOLO and CRNNs. They are accurate in their findings, but in return, they have new sorts of restrictions:

They are complex. The hybrid models have many processing levels: they have the preprocessing, the object detection level, OCR, and the post-processing level, which makes them harder to optimize and maintain. It is for this reason that common hybrid models require more computational time than the standalone OCR engines do.

As will be seen shortly, hybrid techniques for extracting information from tables are not easily portable to other applications and/or documents, unlike general OCR systems that can be trained for and applied to different contexts fairly easily. For example, the model that was trained for identifying vehicle license plates would not be as efficient in the task of optical character recognition on historical manuscripts.

The issues with the application of real-time deployment of YOLO for text region detection and Tesseract OCR for text recognition are that extra fine-tuning is needed for the different datasets and different lighting conditions. This is in addition to the overall processing pipeline, which can present bottlenecks, which are unacceptable for real-time applications such as video-based text detection.

Limitations Since implementing hybrid approaches in other languages, scripts, and orientations requires altering the size of the dataset and training the model. The fact of using several methods sequentially (deep learning, OCR with potential image preprocessing) results in a higher time and space complexity.

D. Challenges in Multilingual and Domain-Specific OCR:
In various practical scenarios, there is a necessity of using OCR systems for recognizing texts in different languages and specializing in certain fields (reports, legal documents, manuscripts, etc.). However, several challenges persist:

Lack of robust multilingual OCR models. It is pertinent to mention that most of the OCR models are trained specifically for the English language, and they do not have a very good performance on languages having complicated characters, such as Japanese, Arabic, and Indic scripts. He cites the need to have the appropriate language-specific training data set, which is often not available.

Specific difficulties Common issues include the identification of mathematical equations, chemical formulas, and tables. Handwritten medicines and doctors' reports are often not reliably understandable because several doctors use different styles of writing.

Randomness and ambiguity are also common for OCR engines because most of the time they are not well equipped for regional dialects and low-resource languages since they lack enough labeled data for training. It is observed that the performance of the multilingual OCR is very poor when the document contains many languages in a single sentence.

E. Addressing These Limitations: Why Tesseract OCR + OpenCV + Pillow is the Optimal Choice

From the reviewed literature, it is clear that there are pros and cons in each technique, and thus none of them is without shortcoming. Nonetheless, when it comes to providing optimized performance based on accuracy, time consumption, and feature versatility, Tesseract OCR in conjunction with OpenCV-Python and Pillow is the best option:

Tesseract OCR is very good OCR software for extracting printed as well as handwritten text in multiple languages. Some of the most used preprocessing techniques employed by OpenCV to prepare an image for a subsequent OCR are discussed as follows. To that end, it supports image operations in handling images to enhance the accuracy of the OCR.

It is not based on deep learning, which allows the combination to run real-time applications on low-power devices. The scope of further research should therefore be invested in the improvement of hybrid models through fine-tuning the preprocessing and increasing the efficiency of the OCR approach on different layouts, handwritten text, and documents in many languages.

4. SYSTEM ARCHITECTURE

It is not based on deep learning, which allows the combination to run real-time applications on low-power devices. The scope of further research should therefore be invested in the improvement of hybrid models through fine-tuning the preprocessing and increasing the efficiency of the OCR approach on different layouts, handwritten text, and documents in many languages. The process begins with the input layer, where images containing text are acquired from various sources, such as scanned documents, photographs, or screenshots. These images may vary in quality, resolution, and text clarity. To improve how well the text can be recognized, the preprocessing layer uses OpenCV and Pillow to change the image to grayscale, reduce noise, set thresholds, and adjust contrast. These techniques refine the image, removing distortions and improving text visibility for accurate extraction.

The OCR processing layer employs Tesseract-OCR to detect and recognize characters, words, and lines once preprocessing is complete. Tesseract converts the processed image into a machine-readable text format, utilizing pattern recognition and language models to improve accuracy. However, the text extracted from raw format could bear errors because of fonts, some levels of skewness or noise, and thus requires subsequent post-processing such as spell-checking, punctuation, and right text formatting.

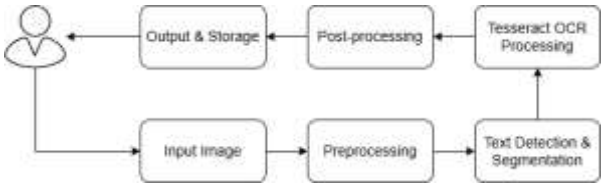

Fig. 1 Process Model

Lastly, in the Output Generation Layer, the best structure and coordinates are used to improve the improved text in the output format, such as text, JSON, or as a database entry, and displayed or exported. This organized, step-by-step method makes sure the system provides very accurate and efficient OCR-based text extraction, which is useful for things like turning documents into digital formats, automatic data entry, and analyzing text in real-time in fields like education, healthcare, and business automation.

A. Input Layer

The input layer is primary to the OCR-based text extraction, where it is responsible for image capture and compatibility of image formats. Sources of input to the system can be text that has been scanned, photos taken by the camera, or screenshots. This is because the system was developed to accept various types of formats, such as JPEG, PNG, TIFF, and BMP, for versatility. Secondly, in addition to the input images being captured from actual or scanned forms of handwritten materials from the environment, the system can also append input data from direct captures through camera units or from scanned writings; this makes the system usable in a number of applications, such as automated data entry and document conversion. After an image is input, the system checks for compatibility of the input file format and also looks at image resolution so that OCR can be done properly. Some of the images are

of low quality, reducing text recognition capabilities, and thereby the user is advised to upload high-quality images where necessary. Moreover, if the text is excessive or tilted in the image, then at the time of preprocessing, the system can proceed through the techniques of deskewing. If the input layer is to handle a number of image formats and sources, the system is set for a solid foundation for the preprocessing and text extraction processes that follow it. This approach enhances the accuracy and reliability of the OCR process and also increases the extraction of text from images [17].

B. Preprocessing Layer

Specifically in the case of the OCR pipeline, the preprocessing layer of the classifier, where images are preprocessed to the optimal condition in order to achieve the best text recognition results, is paramount. Finally, raw images may include noise, varying brightness, distortion, as well as complex backdrops, which negatively affects the OCR operations' efficiency. These problems are then mitigated using various image enhancement techniques with the help of OpenCV and Pillow. The steps below follow to process the images before passing them to the OCR engine. This is the first step of the image pre-processing; we remove color from the images to make the most of the OCR; most of the OCR works optimally for single-channel images. Further, we proceed to apply some denoising algorithms, such as Gaussian blurring or median filter, to reduce other distortions that may hinder the visibility of text contents. The next step involves the use of a thresholding tool for purposes of converting the picture into an image with only two shades separating the text from the background. Some of the common approaches contain Otsu's thresholding and adaptive thresholding, where the contrast varies in accordance with the image illumination. The deskewing technique is mostly used on images where the text is written skewed or not aligned horizontally. Further, the morphological operations, such as dilation or erosion, improve the text structure to ensure thin characters are easily distinguished, or sometimes there are artifacts that are removed. Scaling and cutting are also crucial for the purpose of achieving a proper size of the text suitable to make it more understandable to the OCR software. With these preprocessing steps, the system is able to increase the text detection quality and hence reduce the OCR mistakes in the next text extraction process.

C. OCR Processing Layer

Specifically in the case of the OCR pipeline, the preprocessing layer of the classifier, where images are preprocessed to the optimal condition in order to achieve the best text recognition results, is paramount. Finally, raw images may include noise, varying brightness, distortion, as well as complex backdrops, which negatively affects the OCR operations' efficiency. These problems are then mitigated using various image enhancement techniques with the help of OpenCV and Pillow. The steps below were followed to process the images before passing them to the OCR engine. This is the first step of the image pre-processing; we remove color from the images to make the most of the OCR; most of the OCR works optimally for single-channel images. Further, we proceed to apply some denoising algorithms, such as Gaussian blurring or median filter, to reduce other distortions that may hinder the visibility of text contents. The next step involves the use of a thresholding tool for purposes of converting the picture into an image with only two shades separating the text from the background. Some of the common approaches contain Otsu's thresholding and adaptive thresholding, where the contrast varies in accordance with the image illumination. The deskewing technique is mostly used on images where the text is written skewed or not aligned horizontally. Further, the morphological operations, such as dilation or erosion, improve the text structure to ensure thin characters are easily distinguished, or sometimes there are artifacts that are removed. Scaling and cutting are also crucial for the purpose of achieving a proper size of the text suitable to make it more understandable to the OCR software. With these preprocessing steps, the system is able to increase the text detection quality and hence reduces the OCR mistakes in the next text extraction process.

D. Output Layer

Specifically in the case of the OCR pipeline, the preprocessing layer of the classifier, where images are preprocessed to the optimal condition in order to achieve the best text recognition results, is paramount. Finally, raw images may include noise, varying brightness, distortion, as well as complex backdrops, which negatively affects the OCR operations' efficiency. These problems are then mitigated using various image enhancement techniques with the help of OpenCV and Pillow. The steps below are followed to process the images before passing them to the OCR engine. This is the first step of the image pre-processing; we remove color from the images to make the most of the OCR. Most of the OCR works optimally for single-channel images. Further, we proceed to apply some denoising algorithms, such as Gaussian blurring or median filter, to reduce other distortions that may hinder the visibility of text contents. The next step involves the use of a thresholding tool for purposes of converting the picture into an image with only two shades separating the text from the background. Some of the common approaches contain Otsu's thresholding and adaptive thresholding, where the contrast varies in accordance with the image illumination. The deskewing technique is mostly used on images where the text is written skewed or not aligned horizontally. Further, the morphological operations, such as dilation or erosion, improve the text structure to ensure thin characters are easily distinguished, or sometimes there are artifacts that are removed. Scaling and cutting are also crucial for the purpose of achieving a proper size of the text suitable to make it more understandable to the OCR software. With these preprocessing steps, the system is able to increase the text detection quality and hence reduce the OCR mistakes in the next text extraction process. If the extracted text is part of a scanned document, the system can reconstruct paragraphs and align text formatting accordingly. Moreover, in automated data entry applications, the extracted information can be directly fed into management systems, reducing manual effort. The output layer's flexibility allows the recognized text to be used in different applications, such as turning documents into digital format, processing forms automatically, and using AI for analysis, making the system a strong tool for changing text from images into useful digital information [18].

5. WORKFLOW

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 8, August : 2024

The workflow of this OCR-based text extraction system follows a structured sequence of steps to ensure accurate and efficient processing. It begins with the image acquisition phase, where the user uploads or captures an image containing text. The system supports multiple formats like JPEG, PNG, BMP, and TIFF, making it adaptable to various input sources, such as scanned documents, printed materials, and handwritten notes. Next, the preprocessing stage enhances the image quality using OpenCV and Pillow. This step includes grayscale conversion to simplify color variations, noise removal to eliminate distortions, and thresholding to improve text visibility.

Additionally, operations like cropping, resizing, and contrast adjustment optimize the image for accurate OCR processing. After preprocessing the image, the system employs PyTesseract and Tesseract-OCR for text detection and recognition. The OCR engine identifies text regions, extracts individual characters, and converts them into machine-readable text.

To improve accuracy, the system applies post-processing techniques, such as spell checking, format adjustments, and error handling, ensuring minimal recognition errors. Finally, the system displays the extracted text to the user or saves it in formats such as TXT, CSV, or JSON during the output stage. If required, the system can store the output in a database for further analysis or integrate it with automated data entry systems. The processed text can be used for various applications, including document digitization, real-time text analysis, and AI-driven content processing, making this workflow highly efficient and versatile.

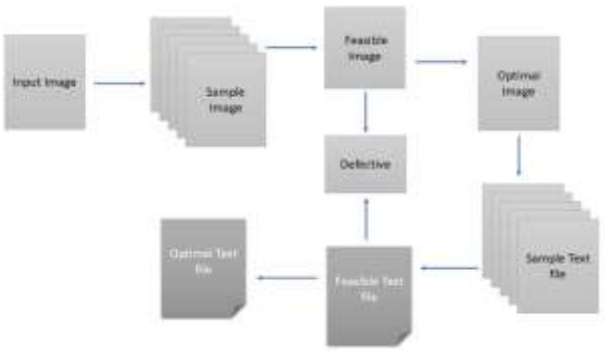

Fig.2 Workflow Diagram.

The OCR process workflow starts with an input image, which can be a scanned document, handwritten note, or any text-containing image. To enhance the accuracy of text extraction, the image undergoes a preprocessing stage using OpenCV and Pillow. This step includes grayscale conversion, which reduces noise by eliminating color information, and thresholding, which converts the image into a binary format for clearer text separation. Additional techniques such as noise removal, resizing, and contrast adjustment further refine the image quality, making it more suitable for Optical Character Recognition (OCR).

Once preprocessing is complete, the processed image is fed into Tesseract-OCR, an open-source OCR engine that detects and extracts text from the image. During this step, the software identifies characters, words, and lines, converting them into digital text. However, the extracted text may contain inaccuracies due to image distortions, font variations, or noise. We apply techniques like spell-checking, punctuation correction, and formatting adjustments to the recognized text during a post-processing stage to improve accuracy. This process ensures that the final output is clean, structured, and machine-readable.

The final step in the workflow is Text Output Generation, where the refined text is stored in a suitable format, such as a text file (.txt), JSON, or database entry. Document digitization, automated data entry, content analysis, and various real-world applications can utilize this output. The improved workflow ensures a systematic, efficient, and accurate extraction of text from images, making it useful in domains like education, healthcare, business automation, and digital archiving.

CONCLUSION

This paper presents a comprehensive review of various research studies focusing on text extraction from images using deep learning-based methods, traditional OCR techniques, and hybrid approaches. The analysis highlights the strengths and weaknesses of various methodologies, demonstrating that while deep learning models provide high accuracy, they require significant computational resources and large training datasets. On the other hand, traditional OCR methods, such as Tesseract OCR, are lightweight and efficient but struggle with handwritten text, complex backgrounds, and multi-script recognition. Hybrid models attempt to bridge this gap by integrating image preprocessing, object detection, and OCR engines to improve text recognition performance. From the literature, it is evident that a combination of Tesseract OCR, OpenCV, and Pillow provides an optimal solution for text extraction tasks. Tesseract OCR ensures efficient character recognition, OpenCV enhances image preprocessing, and Pillow provides flexible image manipulation capabilities. This approach balances accuracy, efficiency, and scalability, making it suitable for real-world applications such as medical document processing, license plate recognition, and scanned text digitization. The future scope of this project is vast, with potential enhancements and applications in various fields. Improvements in deep learning and AI-based OCR models can enhance accuracy, enabling better recognition of handwritten text and complex fonts. Integration with natural language processing (NLP) can help extract meaningful insights from recognized text. Expanding support for multiple languages and real-time

processing can make it more versatile for industries like automation, healthcare, and finance. Additionally, incorporating cloud-based OCR services and mobile applications can improve accessibility and scalability, making text extraction more efficient and widely applicable.

ACKNOWLEDGMENT

The authors would like to express their gratitude to the researchers and developers of Tesseract OCR, OpenCV, and Pillow, whose contributions have significantly advanced text extraction technologies. We thank the authors of the reviewed papers for their insightful research and findings, which provided a strong foundation for this study. Special thanks to academic mentors, colleagues, and peers for their valuable discussions and feedback, which helped refine the review and shape its conclusions. Finally, we extend our appreciation to open-source communities and organizations supporting OCR and image processing research, making tools and datasets accessible for further advancements in this field.

REFERENCES

[1] W. Xue, Q. Li and Q. Xue, "Text Detection and Recognition for Images of Medical Laboratory Reports With a Deep Learning Approach," in *IEEE Access*, vol. 8, pp. 407-416, 2020.
[2] Z. Tang, T. Miyazaki, and S. Omachi, "A Scene-Text Synthesis Engine Achieved Through Learning from Decomposed Real-World Data," *arXiv preprint arXiv:2209.02397*, 2022.
[3] R. Mahadshetti, G.-S. Lee, and D.-J. Choi, "RMFPN: End-to-End Scene Text Recognition Using Multi-Feature Pyramid Network," *IEEE Access*, vol. 11, pp. 61892–61900, 2023
[4] G. Abdul Robby, A. Tandra, I. Susanto, J. Harefa, and A. Chowanda, "Implementation of Optical Character Recognition using Tesseract with the Javanese Script Target in Android Application," in *Procedia Computer Science*, vol. 157, pp. 499–505, 2019.
[5] Mahsa Mohammadi; Mahdi Eftekhari; Amirhossein Hassani, "Image Text Connection: Exploring the Expansion of the Diversity Within Joint Feature Space Similarity Scores," 2023.

[6] D. Vukadin, A. S. Kurdija, G. Delač, and M. Šilić, "Information Extraction From Free-Form CV Documents in Multiple Languages," *IEEE Access*, vol. 9, pp. 84559–84575, 2021.
[7] V. Kumar, P. Kaware, P. Singh, R. Sonkusare, and S. Kumar, "Extraction of Information from Bill Receipts Using Optical Character Recognition," in *2020 International Conference on Smart Electronics and Communication (ICOSEC)*, Trichy, India, 2020, pp. 72–77.
[8] T. Geng, "Transforming Scene Text Detection and Recognition: A Multi-Scale End-to-End Approach With Transformer Framework," *IEEE Access*, vol. 12, pp. 40582–40596, 2024
[9] Y. Watanabe, R. Togo, K. Maeda, T. Ogawa, and M. Haseyama, "Text-Guided Image Manipulation via Generative Adversarial Network With Referring Image Segmentation-Based Guidance," *IEEE Access*, vol. 11, pp. 42534–42545, 2023.
[10] S. K. Alhabeeb and A. A. Al-Shargabi, "Text-to-Image Synthesis With Generative Models: Methods, Datasets, Performance Metrics, Challenges, and Future Direction," *IEEE Access*, vol. 12, pp. 123456–123470, 2024
[11] Y. Wang, "Extraction Algorithm of English Text Information From Color Images Based on Radial Wavelet Transform," in *IEEE Access*, vol. 8, pp. 160050-160064, 2020.
[12] M. Ponnuru, S. P. Ponmalar, L. Amasala, T. Sree, and G. C. Garikipati, "Image-Based Extraction of Prescription Information using OCR Tesseract," *Procedia Computer Science*, vol. 227, pp. 932–938, 2024.
[13] A. Y. Sugiyono, K. Adrio, K. Tanuwijaya, and K. M. Suryaningrum, "Extracting Information from Vehicle Registration Plate using OCR Tesseract," in *Procedia Computer Science*, vol. 227, pp. 932–938, 2023.
[14] S. Lee, J. Lee, C. H. Bae, M.-S. Choi, R. Lee, and S. Ahn, "Optimizing Prompts Using In-Context Few-Shot Learning for Text-to-Image Generative Models," *IEEE Access*, vol. 11, pp. 12345–12358, 2023
[15] M. Sinthuja, C. G. Padubidri, G. S. Jayachandra, M. C. Teja, and G. S. P. Kumar, "Extraction of Text from Images Using Deep Learning," *Procedia Computer Science*, vol. 227, pp. 932–938, 2024.
[16] Okechukwu Ogochukwu Patience, E. M. Amaechi, O. George, and O. N. Isaac, "Enhanced Text Recognition in Images Using Tesseract OCR within the Laravel Framework", Asian J. Res. Com. Sci., vol. 17, no. 9, pp. 58–69, Sep. 2024.
[17] Sandeep Dwarkanath Pande, Pramod Pandurang Jadhav, Rahul Joshi, Amol Dattatray Sawant, Vaibhav Muddebihalkar, Suresh Rathod, Madhuri Navnath Gurav, Soumitra Das, "Digitization of handwritten Devanagari text using CNN transfer learning – A better customer service support", Neuroscience Informatics, Volume 2, Issue 3, 2022.
[18] Pande, Sandeep Dwarkanath et al. 'Shape and Textural Based Image Retrieval Using K-NN Classifier', Journal of Intelligent & Fuzzy Systems, vol. 43, no. 4, pp. 4757-4768, 2022.