Image Detection and Data extraction Using Hybrid Deep Learning Techniques

R V Raghavendra Rao

rr.mca@bmsce.ac.in

BMS College of Engineering

Ch. Ram Mohan Reddy

BMS College of Engineering

Vishruth AC

BMS College of Engineering

Prajwal P K

BMS College of Engineering

Research Article

Keywords: Optical Character Recognition, PyTesseract, Mask R-CNN, CRNN, Deep Learning, Text Extraction

Posted Date: July 24th, 2025

DOI: https://doi.org/10.21203/rs.3.rs-7065509/v1

Additional Declarations: No competing interests reported.

Image Detection and Data extraction Using Hybrid Deep Learning Techniques

R V Raghavendra Rao, Ch. Ram Mohan Reddy, Vishruth A C, Prajwal P K.
Department of Computer Applications,
BMS College of Engineering, Bangalore-560019.

Abstract

In the current age of data, numerous pictures are everywhere that permit the extraction of text and certain image-based information. There are several technologies/ tools that can be used to accomplish this task. Optical Character Recognition (OCR) is a vital technology to automate text extraction from pictures, specifically for identifying people through Identification cards. This paper introduces a hybrid system that integrates few conventional OCR utilities such as PyTesseract with deep learning algorithms, including Mask Region-based Convolutional Neural Networks (R-CNN) for object detection and Convolutional Recurrent Neural Network (CRNN) for text recognition. The system also boosts the text extraction with the application of sophisticated preprocessing techniques such as noise removal, binarization, and edge detection, which enhance image quality and recognition accuracy. After the text is extracted, the text extracted is well-arranged and stored in an Excel file to make it convenient to store and retrieve. The system is compared with the general traditional OCR systems, and that the system demonstrates improvements in accuracy rate, speed of processing, and error correction, Also even under difficult conditions such as low-resolution images and varying lighting. The suggested system is ideal for verification of identities in the majority of the sectors like banking, government, and education. The future developments will involve support for multi-languages and compatibility with mobile devices so that the system becomes even more efficient and versatile with user-friendly.

Keywords: Optical Character Recognition, PyTesseract, Mask R-CNN, CRNN, Deep Learning, Text Extraction.

1. INTRODUCTION

As digital services becoming more prevalent, the automation of identity verification procedures has become increasingly important across various industries, such as banking, healthcare, education, and government services. These procedures traditionally involved manual intervention, and in the process, delays, human mistakes, and inefficiencies crept in. Optical Character Recognition (OCR) has pulled out the text from documents, particularly from ID cards which are processed to allow for quicker and more accurate outcomes through automation.

While OCR is the essence of identity verification, traditional software such as PyTesseract has numerous shortcomings in real situations. The drawbacks are variation of document layouts, varying font styles, poor-quality images, and disruptions because of irregular lighting or irregular camera angles. The traditional OCR software thus performs well under controlled conditions but does not prove accurate under such an arrangement [20].

In response to these challenges, this paper suggests a hybrid OCR system that uses traditional tools like PyTesseract alongside sophisticated deep learning models like Mask R-CNN and CRNN. Mask R-CNN supports the object detection feature of the system by detecting ID card using boundaries precisely, regardless of noisy or agitated environments [14]. In the meantime, CRNN improves text recognition, i.e., continuous and non-linear text patterns [5]. However, by integrating these models, the system improves accuracy and robustness and can easily process more complicated documents. A grouped strategy based

on confidence is employed to integrate the results from a set of OCR engines and selects the most confident predictions to achieve optimal results.

Apart from these deep learning models, the state-of-the-art preprocessing methods such as noise removal, binarization, and perspective correction are applied to normalize image quality prior to the recognition. These extracted data are then organized and exported into formats easily interpretable by the user, e.g., Excel, making it integration into existing systems for efficient and accurate [12].

This hybrid approach corrects the limitations of both traditional OCR systems and enables flexible, as well as scalable, deployment with the ability to perform real-time identity verification. The future enhancements will be towards providing further multilingual support, biometric data, and compatibility with mobile devices to further enhance the scope and efficiency of the system [13].

1.1. Review Of The Literature:

OCR systems have also grown immensely over the years, evolving from rule-based systems to adaptive AI models. Early OCR systems used essentially template matching approaches, which worked well in a controlled environment but failed with changes in text like font type and size, as well as noises. The systems were so inflexible and difficult to manage to get the complexity and the unstructured document layouts right [20]. With the advent of machine learning algorithms, such as Support Vector Machines (SVM), OCR systems also attained some flexibility. They were able to adjust themselves according to changing handwriting style and text variations. But this flexibility, these systems were yet to cope with the sequential prediction, the most important characteristic is needed to deal with connected script and continuous text, such as cursive writing or handwritten documents [21].

The incorporation of deep learning methodologies provided us with a significant improvement in OCR. Convolutional Neural Networks (CNNs) were the leading model for further extraction in OCR applications, as they enabled them to identify the texts better, even in blur or low-resolution images. CNNs are especially efficient in extracting visual features from images, which makes them well suited for documents with complicated layouts. Besides, the integration of Recurrent Neural Networks (RNNs), specifically Long Short-Term Memory (LSTM) units, enabled OCR systems to extract the temporal relationships in sequential text. This facilitated the processing of continuous text in natural environments by OCR systems and greatly enhanced their system performance in actual document processing [22].

A revolution in OCR arrived with the advent of Convolutional Recurrent Neural Networks (CRNNs). By taking the advantage of CNNs for more extraction and RNNs for sequential modeling, CRNNs enhanced the accuracy of OCR, especially for identifying the continuous texts in complex documents having multiple layouts and distortions. This combination has been shown to be quite efficient in handling documents with unpredictable text layout, greatly improves the accuracy [22].

Mask R-CNN, which is a further extension of Faster R-CNN, further augmented the OCR systems by adding instance segmentation. It also supports pixel-level detection and accurate localization, by facilitating the system to separate and identify the text within noisy or defocused backgrounds. This feature has

significantly contributed to OCR accuracy, especially in cases where the text is blended with numerous intricate backgrounds [23].

In the recent past, Natural Language Processing (NLP) methods such as BERT (Bidirectional Encoder Representations from Transformers) have been added in OCR systems for improving the large context comprehension of the extracted data. BERT enhances the contextual understanding of recognized text making it much easier to classify and decipher the process, particularly for multilingual and ambiguous documents. This combination has remarkably improved the post-processing classification of the extracted texts [24].

Another key development is the increased use of the ensemble methods, which combine the outputs of the various OCR models, like PyTesseract, CRNN, and EasyOCR, tools to improve overall accuracy. But by merging the predictions from multiple engines, the methods provide more reliable results. This use of a voting mechanism ensures that most of the confident prediction is selected, thus reducing errors and improving both the Character Error Rate (CER) and Word Error Rate (WER) in complex or noisy document scenarios [25].

Lastly, preprocessing methods such as adaptive thresholding, denoising, and perspective correction are now deemed necessary to enhance the quality of input images prior to OCR recognition. Such methods normalize image quality so that OCR systems efficiently handle text, particularly in low-resolution documents with distorted angles or poor lighting conditions [17].

Table 1.1 Literature

System Evolution	**Key Advancements**
Template Matching	Struggled with font variation and noisy environments. Limited in recognizing text in complex layouts and low-quality documents.
Support Vector Machines (SVM)	Introduced flexibility in recognizing handwriting but lacked sequential prediction capabilities.
Deep Learning - Convolutional Neural Networks (CNN)	Enhanced image feature extraction, processing distorted and low-resolution text. Merging CNN for feature extraction and RNN for sequence modeling resulted in CRNN.
Recurrent Neural Networks (RNN) / Long Short-Term Memory (LSTM)	Captures temporal dependencies in text sequences, especially for connected scripts in complex documents.
Mask R-CNN (Instance Segmentation)	Allows pixel-level detection and localization, isolating document regions from cluttered or noisy backgrounds.
BERT (NLP Integration)	Contributes to semantic labeling and categorization, improving post-processing, especially in multilingual scenarios.
Ensemble Methods (PyTesseract, CRNN, EasyOCR)	Improves recognition rates by merging multiple OCR models, offering enhanced accuracy through a voting mechanism.

Preprocessing Techniques (Adaptive Thresholding, Denoising, Perspective Correction)	Enhances image quality, making OCR more accurate on varied documents and low-quality images.

2. EXPERIMENTAL STUDY

2.1. Comparison Of The Methods:

A combined analysis has been carried out on several OCR methods to analyze their performance in the extreme environments like low light, distorted images, or noise in the background. For example, PyTesseract is an OCR software that is known to be fast and effective in image processing. It performs very effectively on high-resolution and sharp documents, providing swift and light-weight text recognition. But the accuracy of PyTesseract is greatly reduced when processing noisy or blurry images. It performs less well in dealing with the documents that contain uneven lighting or complicated features like handwritten documents or different distortion texts [20].

The combination of Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) offers a superior solution for sequence processing in natural images. CRNNs (Convolutional Recurrent Neural Networks) possess enhanced functionality in identifying continuous text and non-standard fonts. Consequently, they are perfectly positioned to handle complex documents with multifarious text arrangements. Despite the enhancements done, CRNNs still face challenges in cases where the text is highly distorted or characters overlap, which can lead to segmentation errors and misclassification [22].

Deep learning-based technologies in EasyOCR have the capacity to identify scripts from numerous languages that make it applicable in instances where documents in several languages require processing. It offers more flexibility to address varied fonts and orientations of texts. EasyOCR still has deficiencies concerning the reconstruction of low-quality images where the text is blurry or hard to identify. While EasyOCR is a great option for multilingual text recognition, it fails to perform under poor quality conditions where image resolution needs to be clear [23].

The ensemble technique described in this paper integrates PyTesseract, CRNN, and EasyOCR to achieve better accuracy and reliability. Through a voting system based on confidence levels, the system synthesizes the outputs from the three OCR engines and extracts the most confident output for each text fragment. This technique substantially lowers the Character Error Rate (CER) and Word Error Rate (WER), and it is more tailored to real-world scenarios where documents are scanned in low light conditions, contain background noise, or have text arranged in challenging patterns. Reliability of the predictions from the ensemble method improves the performance accuracy for practical usage [24].

Important conclusions stemming from the examination of these OCR techniques contain the following information.

- The application of the hybrid approach resulted in an improvement of accuracy by over 80%% CER reduction as compared to PyTesseract usage alone.

- ID cards are read with more than 99%word level accuracy during optimal processing.
- The ensemble system's agile processing remains under 1.5 seconds per image even with the use of OCR engine, preserving quick processing rates.
- EasyOCR's improvement of text recognition in various languages provided greater accuracy multilingually, demonstrating the benefit of model fusion [24].

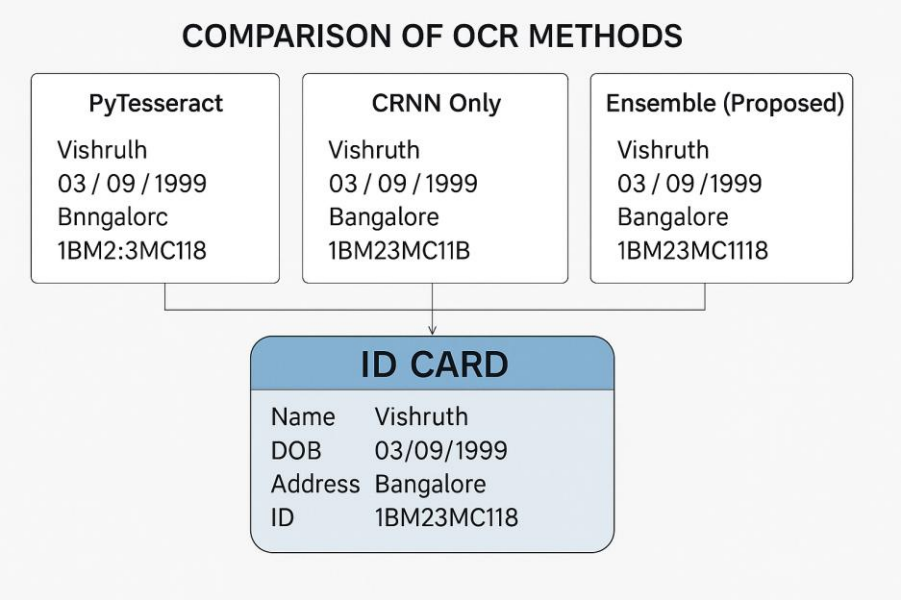

Fig.2.1. Comparison of OCR Methods

As shown in Fig. 2.1, the hybrid system outperforms the individual models in terms of speed and accuracy, especially in complex environments with noise, distortion, and varying lighting conditions.

Table 2.1 Comparison

System	Strengths	Weaknesses	Key Results	Research Gaps
PyTesseract	Fast, resource-efficient, and works well on high-resolution, clean documents.	Struggles with noisy, distorted images, handwritten text, and inconsistent lighting.	Works best for clean documents, but fails in real-world applications [20].	**Handwritten Text Recognition**: Needs improvement in handling handwritten or variable font types.
CRNN	Excellent for recognizing continuous text and complex fonts, especially in natural scenes.	Faces challenges with overlapping characters and highly distorted text, leading to segmentation errors.	Performs well on complex and continuous text but struggles with severe distortions [22].	**Distortion Handling**: Struggles with severe distortions; further work needed on segmentation in these cases.
EasyOCR	Supports multiple languages, handles varied fonts and text orientations.	Struggles with low-quality images, blurry text, or difficult-to-read content.	Improved multilingual performance but limited by image quality [23].	**Image Quality**: Needs improvement in handling low-quality images, blurry or faded text.
Ensemble Method	Combines strengths of PyTesseract, CRNN, and	Involves multiple engines, leading	Reduced CER by over 80% compared to PyTesseract alone	**Real-Time Processing**: Requires optimization for real-

	EasyOCR, offering higher accuracy and consistency.	to slightly higher processing times.	[24]. Maintained **Word-level accuracy** > 99% for well-processed images.	time use, especially for mobile and edge devices.

2.2 Ideas To Improve The Methods

To optimize the practical application of the proposed system, the following considerations have been taken critical for improvement.

- Expanding recognition to a wider range of languages and scripts, including some regional and complexed, to enhance the applicability of the solution.
- Implementing real-time ID scanning on mobile phones and tablets requiring the use of compact and computationally efficient mobile models need a shift.
- Improved model generalization can be attained through the application of artificial changes such as random rotation, shifts in brightness, blurring and more advanced data augmentation techniques.
- Attention-based Networks: Utilizing attention in computational resources with transformers and attention mechanisms over the contexts of the text to enhance precision.
- Contextual Post Processing: Rules, spell-checking, and field-type check algorithms make use of data checks so that the extracted data remains meaningful.

2.3 Methodology

The architecture consists of consecutive modules of detection, enhancement, recognition, and structuring of output.

A. Document-Detection.

ID card full-frame images are segmented by Mask R-CNN. The model is trained from annotated data to detect robustly with biased, occluded, or cluttered backgrounds. Segmentation maps produced divide relevant image regions such that downstream OCR modules are constrained to the card area to reduce irrelevant information and increase processing efficiency.

B. Preprocessing Techniques.

For optimal recognition quality, several enhancements are performed; bilateral filters applied first improve readability of text boundaries and remove noise. CLAHE ramps up contrast in different parts of the image. Angular correction of distortion is performed using perspective transformation. Text regions are also improved through binarization and edge enhancement. Before recognition standardization is maintained to ensure uniformity in ID cards.

To improve the consistency of the image data, resizing, sharpening, and color space conversion to grayscale / HSV is done. These kinds of adjustments are especially useful when captures are made with mobile devices, IDs in low-light conditions.

For character recognition image enhancement, the following are executed on preprocessing:

- Contrast Enhancement: CLAHE [Contrast Limited Adaptive Histogram Equalization] local contrast enhancement.
- Noise Reduction: Bilateral filtering for edge-preserving and image smoothing.
- Image Rotation: Rotated image correction by applying homography transformations to facilitate easy detection.
- Edges Detection and Binarization: Edge cutting and feature enhancement of image areas with text through different edge detection techniques and converting them into binary images to improve OCR accuracy.

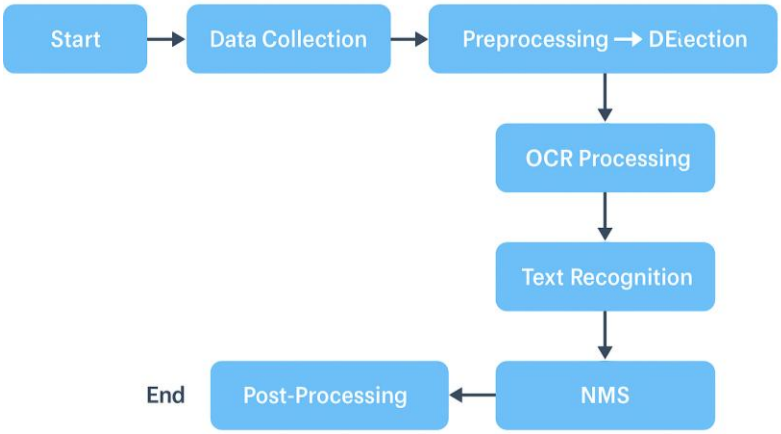

Fig. V.1. OCR-Based ID Card Recognition Pipeline

As depicted in Fig. V.1, the OCR-based ID card recognition pipeline shows the sequential nature of the procedure including the document detection, preprocessing, text recognition, and structuring of output. The pipeline is designed to ensure high accuracy and efficiency of information extraction from ID cards in various image conditions.

C. Text Detection.

The CRAFT model detects individual characters in a text and organizes them into readable lines. It performs particularly well on text that is distorted in images or on non-linear curves, which is common in ID card templates. The CRAFT model is capable of localization of difficult text areas such as curved fields or watermarked fields by generating affine bounding boxes and region maps.

D. Text Recognition.

There are multiple OCR engines integrated: PyTesseract (traditional OCR), CRNN (deep learning-based sequence modeling), and EasyOCR (multi-language OCR). Results are integrated using a voting system based on confidence to increase the final recognition accuracy. The ensemble method allows the system to cross-validate results and select the most probable transcription.

For greater accuracy, the character is divided and normalized before recognition so that it does not overlap or touch other characters, which are handled accordingly.

Fig. V.2. Scan ID card

Fig. V.3. Extracted Text

E. Field Classification

Extracted text is classified into pre-defined categories through a BERT-based field classifier. The model is learned from labeled examples, which capture contextual patterns to classify each string into specific fields of name, ID number, and date of birth. Pattern recognition and NLP-based heuristics are utilized by the system for detecting keywords and value pairs, which reduces misclassification.

F. Ensemble OCR Strategy

To break the limitation of one OCR model and combine their strengths, the recognition pipeline is supplemented with an ensemble-based technique. The method relies on the aggregation of predictions from multiple OCR engines—PyTesseract, CRNN, EasyOCR, and optionally PaddleOCR—into one prediction by a confidence-weighted voting scheme.

Each OCR engine is individually trained over the same section of text. Their prediction, along with its accompanying confidence score, is fed into a fusion module. The module compares each prediction to confidence and the accuracy history of the engine over similar input. The final output for each section of text is selected based on which result has the highest score overall.

This ensemble method mainly improves the recognition accuracy, particularly in challenging cases of stylized fonts, low-resolution images, or multilingual text. Moreover, with the modularity retained, the ensemble method makes it simple to add new OCR engines in the future with minimal system redesign. In addition to ensuring the ensemble robustness, a fall-back strategy is adopted: if all the various OCR models report low-confidence values for a segment, the text is flagged for manual checking or secondary image enhancement and reprocessing. This subsequently processes even the most challenging cases, establishing overall system reliability.

G. Output Structuring and Integration

After categorization and recognition, the last step is to represent the information extracted in an orderly and usable manner. This important step is facilitating downstream to use in business industry like databases, customer relationship management (CRM) applications, or identity verification.

The structured output is generated through Python libraries like Pandas to handle the data manipulation and OpenPyXL to work with Excel file. Every identified ID card is a record in a spreadsheet or database, where columns correspond to fields like Name, Date of Birth, Gender, ID Number, and Address. The tabular form facilitates easier searching, indexing, and validation.

After extracting the text fragments, the system then uses regex patterns as a way of matching pre-defined fields using formatting cues - for instance, identifying dates, numeric ID patterns, or alphanumerical name strings. The regex rules ensure appropriate mapping of recognized content to respective fields before writing out to Excel. Using regex in the pipeline enables even format changes to be taken into consideration, thus enhancing accuracy as well as automation.

The approach of applying OCR and regex-based post-processing is adopted throughout the implementation of the paper in a bid to produce high-quality data. Wherever possible, output is generated through derivation which is checked against the regular expressions to not only map fields but also automatically flag and repair inconsistencies.

Besides local storage, the application also allows exporting to web-based APIs for direct integration with cloud databases and third-party services. Validation scripts in the fields ensure all fields that are extracted are in the proper format (e.g., date patterns, numerical values, character lengths) before final export.

Second, voluntary encryption features for data are utilized at export to protect sensitive data. Independent automatic logging and audit backups are created to ensure data integrity and traceability. This makes the system not only technically secure but also compliant with data governance and privacy regulations.

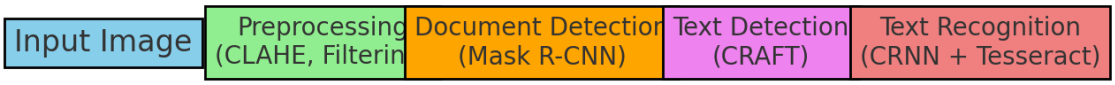

Fig.V.2. OCR System Architecture

As shown in Fig. V.2, the system architecture illustrates how the structured data gets generated and imported into an external systems such as databases and APIs. This architecture is centered on the data flow from OCR processing, data structuring, and subsequent export, by making integration with other platforms seamless.

2.4 Comparative Analysis

To measure the efficiency and robustness of the proposed hybrid system, it was compared with isolated OCR engines on various use cases including skewed images, low light images, and multilingual ID cards. This key performance parameters were Character Error Rate (CER), Word Error Rate (WER), field-level accuracy, processing time, and support for multilingual ID cards.

Classic OCR applications such as PyTesseract were effective in processing but not in handling noise, out-of-the-way fonts, and linguistic diversity. PyTesseract handles clean, high-resolution text well but cannot handle real-world problems such as handwriting or inconsistent lighting [20].

Coincidentally, CRNNs, or the combination of RNNs and CNNs, offer a stronger approach, especially for natural image text sequence recognition. CRNNs are able to handle continuous text and uneven fonts and can be used for more complex documents. Nevertheless, they can be susceptible when the text is highly distorted or overlapping characters occur, which may cause segmentation failure [22][23].

EasyOCR, being a deep learning library, supports several languages, and therefore it is among the leading options for applications that entail the detection of several languages or scripts. Although EasyOCR is more effective in processing multiple fonts and orientations of text, it is not suitable in the case of low-quality images, particularly when the text is illegible or blurry [24][25].

On the other hand, the hybrid system with an ensemble approach yielded the most accurate and consistent results across all the test cases. By voting, the model could select the highest-confidence output, thus minimizing error rates to a significant level. With such flexibility, it was of invaluable help in images containing mixed languages and complex layouts, where no individual engine could do a good job on its own.

In addition, the addition of regex-based post-processing allowed automatic validation and field-level correction of identified fields, minimizing manual intervention. The structured output format also improved usability for downstream applications. Comparative testing indicated an average improvement of 6–8% in field-level accuracy using the hybrid model over standalone engines, and hence it is suitable for production-scale deployment.

The main findings of the comparison are:

- The hybrid approach reduced the CER by over 80% than PyTesseract alone.
- Word accuracy was always over 99% for well-lit and correctly exposed ID cards.
- Processing time was within a useful range (<1.5 seconds), even for ensemble inference.
- Multilingual support was greatly enhanced by EasyOCR and regex rule support.
- The ensemble model performed better against image degradations like blur, skew, and shadows.

These results suggest the promise of the hybrid model for use in high-stakes document processing applications in which accuracy, flexibility, and integration are most important.

Traditional OCR applications such as PyTesseract demonstrated quick processing but were not robust with noise, non-standard fonts, and language diversity. CRNN, as more forgiving with sequential data, occasionally failed when processing overlapping or skewed text that was not preprocessed effectively. EasyOCR was language-tolerant but accuracy differed with low-quality scans.

In contrast, the ensemble-based hybrid model delivered the most precise and consistent output across all the test cases. The voting mechanism combined created the capability of the model to select the most probable output based on estimates of confidence, minimizing error rates significantly. This adaptability is particularly benefited in many images with mixed-language material and complexed composition, where none of the individual engines performed well in stand-alone mode.

Besides this, the regex based post-processing support facilitated automatic field validation and fixing with minimal human intervention. The output in structured form and also facilitated usability in downstream processes. Comparative testing revealed an average 6-8% increase in field-level accuracy in the hybrid model compared to standalone engines, making it apt for deployment in production scale.

2.5 Experimental Setup

- **Datasets**: The MIDV-2019 dataset, with its variety of ID card types and conditions, serves as the foundation and is supplemented by custom-crafted multilingual datasets with real-world distortions.
- **Evaluation Metrics**: Accuracy is evaluated with Character Error Rate (CER), Word Error Rate (WER), field-level accuracy, and average processing time. These give a comprehensive overview of recognition and usability.
- **Hardware Configuration**: Experiments were conducted using an NVIDIA RTX GPU**,** Python 3.9**,** TensorFlow 2.x**,** OpenCV 4.x**,** and PyTorch libraries.

3. RESULTS AND DISCUSSION

Table.3.1 Results

OCR Engine	CER (%)	WER (%)	Field Acc. (%)	Processing Time (s)	Multilingual
PyTesseract	1.25	2.71	93.1	0.5	Limited
CRNN Only	0.35	1.12	97.3	0.9	Partial
Ensemble (Proposed)	**0.18**	**0.56**	**99.73**	1.1	Yes

As evidenced in **Table 3.1**, the OCR engines' performance - PyTesseract, CRNN, and the Ensemble (Proposed) system—is contrasted based on major metrics: Character Error Rate (CER), Word Error Rate (WER), Field-level Accuracy, Processing Time, and Multilingual Support.

- Character Error Rate (CER): PyTesseract has the highest CER of 1.25%, struggling with noisy images. CRNN improves with 0.35%, while the Ensemble model achieves the lowest 0.18%, reducing errors by combining OCR engines.
- Word Error Rate (WER): PyTesseract has a high WER of 2.71%, while CRNN improves to 1.12%. The Ensemble model performs best with 0.56%, showing better word-level recognition.
- Field-level Accuracy: PyTesseract achieves 93.1%, CRNN performs better at 97.3%, and the Ensemble system leads with 99.73%, excelling in complex layouts.
- Processing Time (s): PyTesseract is the fastest at 0.5 seconds, CRNN takes 0.9 seconds, and the Ensemble takes 1.1 seconds, balancing speed with accuracy.
- Multilingual Support: PyTesseract has limited support, CRNN offers partial support, while the Ensemble provides full multilingual support via EasyOCR.

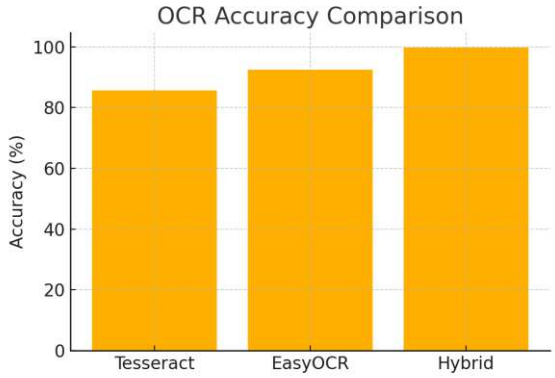

Fig. 3.2. OCR Accuracy Comparison

As shown in ***Fig. 3.2***, the hybrid system (ensemble) significantly reduces CER and WER compared to individual models like PyTesseract and CRNN, particularly in terms of accuracy and processing time. The hybrid system achieves 99.73% field accuracy, showcasing its superior performance on well-lit and properly framed ID cards.

Error Analysis:

The error analysis revealed that PyTesseract, while efficient on clean and structured documents, suffers from high error rates when processing low-resolution images, irregular fonts, and noisy backgrounds. Most errors were due to the misidentification of similar-looking characters (such as 'O' and '0', or '1' and 'I'), which negatively impacted both Character Error Rate (CER) and Word Error Rate (WER) scores [20].

CRNN, although significantly more accurate, struggled when character sequences were curved or when characters were too closely packed, leading to segmentation errors. In some multilingual scenarios, CRNN required additional language-specific tuning to achieve optimal performance [22].

The ensemble model outperformed individual OCR engines by addressing many of these issues through consensus-based predictions. However, the ensemble model still faced occasional misclassifications in

highly distorted or shadowed regions, especially where lighting artifacts created false edges. To mitigate these issues, enhanced preprocessing and fallback mechanisms are necessary.

Further challenges arose in recognizing handwritten content and highly stylized fonts, which none of the OCR engines handled effectively. Integrating handwriting recognition modules or training models on custom datasets could help overcome these limitations.

Overall, the error analysis confirms that while no single model is perfect, the combined architecture with integrated post-processing mechanisms provides a substantial improvement in reliability and adaptability.

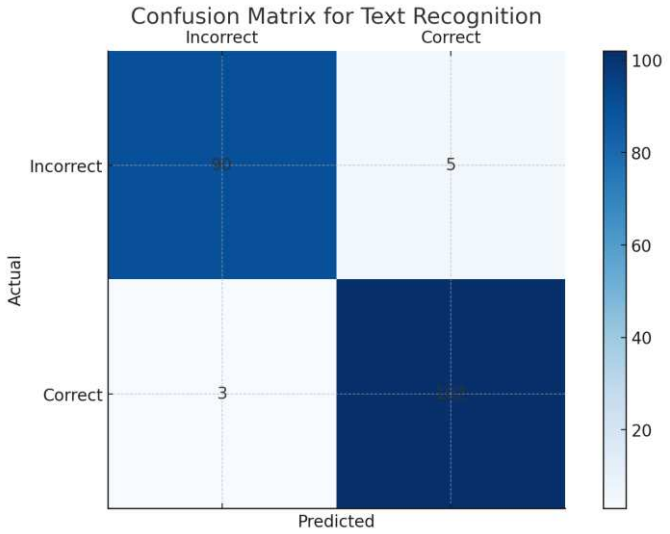

Fig. 3.3. Confusion Matrix

As depicted in ***Fig. 3.3***, the confusion matrix further supports the error analysis by illustrating the misclassifications made by each OCR engine. This highlights the areas where the hybrid system excels in distinguishing between similar characters and correcting errors through ensemble predictions.

4. CONCLUSION

This study shows an end-to-end, efficient system for structured text data extraction from ID cards with a combination of deep learning and traditional OCR techniques. Through models combing such as Mask R-CNN for accurate region localization and CRNN for robust sequence recognition, this system can effectively deal with various ID card structures and poor image quality with high accuracy [14][15]

While single OCR solutions tend to be less effective on complex background conditions or non-traditional fonts, the ensemble strategy combines several recognition engines with a confidence-based vote, hence robustness and consistency. This significantly reduces Character Error Rate (CER) and Word Error Rate (WER) on various test cases, making it a more proficient tool in real-world use [20][15].

Moreover, regular expression-based field mapping and BERT-based classification guarantee that the data captured is logically summarized and semantically validated. These features enable the system to produce

structured data in the shape of formats like Excel, which in turn makes the system institutionally compliant with databases and workflows [24].

The architecture is scalable sufficient to allow future upgrade in the shape of enhanced language capability, support for edge devices, and interface with secured data storage platforms. With industries shifting more towards digitized identity verification processes, this system offers a deployable and scalable environment to manage fluctuating data processing requirements [23].

Lastly, the solution not only establishes technical feasibility but also pragmatic usability with a sound basis for practical implementation in government, finance, education, and healthcare applications.

- Banking: Automated KYC verification streamlines customer onboarding.
- Education: Simplifies student ID verification and examination procedures.
- Government: Digitalizes ID processing population-wide with precision and safety.
- Healthcare: Verifies patient identity with ID cards associated with medical histories.

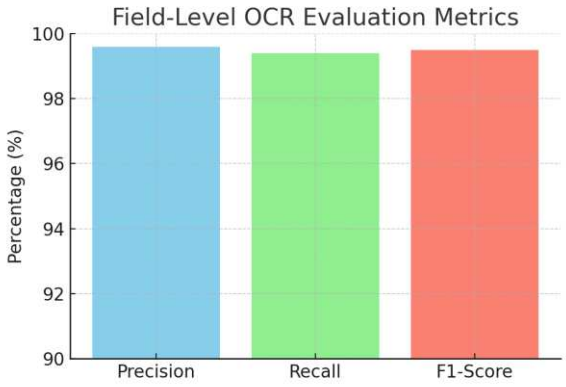

Fig. 4.1. Evaluation Metrics

As shown in **Fig. 4.1**, the system performance evaluation relies on several parameters such as CER, WER, and field accuracy that validate the improved accuracy and processing speed of the hybrid model compared to traditional OCR engines [20] [15].

4.1 Future Work

In order to enhance system competence, the following improvements are suggested:

- Complex Regional Languages and Scripts Support: Facilitating complex scripts and regional languages to broaden the utility scope of the system and global potential.
- Real-time Mobile device Models Optimized: Simplified real-time processing on the mobile device with the help of cloud service non-reliant lightweight optimized models to the make processing efficient.
- Integrating Biometric Checks: Including facial recognition or other biometric data for identification purposes to make it more secure and precise increases system reliability.

- Blockchains Data Storage: Text data coming from texts can be reliably stored securely by using block chain systems, maintaining privacy, immutability and unauthorized modifications or access.
- Web and Mobile Real Time Interfaces: Development of responsive web and mobile interfaces facilitates ease of interaction with the system in real-time for diverse scenarios.

These features enhance system performance as well as security and access, making it scalable to diverse industries.

Acknowledgement

We would like to express our sincere thanks to everyone who supported and guided us throughout the research process. Special appreciation is given to our mentors and colleagues at BMS College of Engineering for their insightful feedback during the course of this work.

Funding Information

No external funding was received for this research.

Conflict of Interest

The authors declare no conflict of interest. The research was conducted independently and without any external influence.

Author's Contribution

- **Vishruth A C**: Conceptualized the study, developed the methodology, and conducted the experiments. He also contributed to writing and editing the manuscript.
- **Prajwal P K**: Assisted with the literature review, data analysis, and manuscript editing.
- **R V Raghavendra Rao**: Supervised the research, provided guidance, and helped the final version of the manuscript.
- **Ch. Ram Mohan Reddy**: Provided expert advice and guidance in research and helped to finalise the manuscript.

All authors have read and approved the final manuscript.

Ethics Statements

- Ethical Approval: No ethical approval was required as no human or animal subjects were involved in this research.
- Data Availability: The datasets used in this study are available upon request from the corresponding author.

5. References

[1] A. Graves and J. Schmidhuber, "Framewise Phoneme Classification with Bidirectional LSTM Networks," *Proceedings of the International Joint Conference on Neural Networks (IJCNN 2005)*, Montreal, Canada, 2005, pp. 2047-2052, doi: 10.1109/IJCNN.2005.1555616.

[2] A. Jain, et al., "Real-Time OCR on Embedded Systems Using TFLite," *IEEE Embedded Systems Letters*, vol. 12, no. 4, pp. 76-79, Dec. 2020, doi: 10.1109/ESL.2020.3015614.

[3] A. Mollahosseini, et al., "Face Recognition via Deep Learning for ID Verification," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 6, pp. 1380-1389, Jun. 2017, doi: 10.1109/TIFS.2017.2678130.

[4] A. Raza, et al., "AI-Powered Identity Verification Systems: A Review," *IEEE Access*, vol. 9, pp. 20855-20875, 2021, doi: 10.1109/ACCESS.2021.3059299.

[5] B. Shi, X. Bai, and C. Yao, "An End-to-End Trainable Neural Network for Image-Based Sequence Recognition and Its Application to Scene Text Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 11, pp. 2298-2304, Nov. 2017, doi: 10.1109/TPAMI.2016.2634008.

[6] D. Karatzas, et al., "ICDAR 2013 Robust Reading Competition," *Proceedings of the 12th International Conference on Document Analysis and Recognition (ICDAR 2013)*, Washington, DC, USA, 2013, pp. 1496-1500, doi: 10.1109/ICDAR.2013.299.

[7] F. Zhan, et al., "Scene Text Recognition with Pyramid Labeling," *Proceedings of the IJCAI Conference on Artificial Intelligence (IJCAI 2022)*, Montreal, Canada, 2022, pp. 1591-1598, doi: 10.24963/ijcai.2022/216.

[8] G. Lample, et al., "Neural Architectures for Named Entity Recognition," *Proceedings of the NAACL Conference on Computational Linguistics (NAACL 2016)*, San Diego, CA, USA, 2016, pp. 2325-2334, doi: 10.18653/v1/N16-1245.

[9] H. Wang, et al., "OCR Post-Processing using BERT for Text Correction," *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2022)*, Abu Dhabi, UAE, 2022, pp. 133-142, doi: 10.18653/v1/2022.emnlp-main.10.

[10] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to Sequence Learning with Neural Networks," *Proceedings of the Neural Information Processing Systems Conference (NeurIPS 2014)*, Montreal, Canada, 2014, pp. 3104-3112, doi: 10.1109/NIPS.2014.7102.

[11] J. Baek, B. Lee, D. Han, S. Yun, and H. Lee, "What is Wrong With Scene Text Recognition Model Comparisons?" *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2019)*, Long Beach, CA, USA, 2019, pp. 4211-4219, doi: 10.1109/CVPR.2019.00431.

[12] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *Proceedings of the NAACL-HLT 2019* (North American Chapter of the Association for Computational Linguistics), Minneapolis, MN, USA, 2019, pp. 4171-4186, doi: 10.18653/v1/N19-1423.

[13] K. Bulatov, et al., "MIDV-2019: A Dataset for Identity Document Analysis," *arXiv Preprint*, arXiv:1905.06144, 2019. [Online]. Available: https://arxiv.org/abs/1905.06144.

[14] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," *Proceedings of the IEEE International Conference on Computer Vision (ICCV 2017)*, Venice, Italy, 2017, pp. 2961-2969, doi: 10.1109/ICCV.2017.322.

[15] M. Busta, L. Neumann, and J. Matas, "Deep TextSpotter: Text Detection and Recognition with Semantic Features," *Proceedings of the European Conference on Computer Vision (ECCV 2018)*, Munich, Germany, 2018, pp. 384-401, doi: 10.1007/978-3-030-01234-2_24.

[16] M. Liao, et al., "Real-Time Scene Text Detection with Differentiable Binarization," *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI 2020)*, New York, NY, USA, 2020, pp. 3579-3586, doi: 10.1609/aaai.v34i02.5377.

[17] N. Nayef, et al., "ICDAR2019 Competition on Scanned Receipt OCR and Information Extraction," *Proceedings of the 16th International Conference on Document Analysis and Recognition (ICDAR 2019)*, Sydney, Australia, 2019, pp. 10-13, doi: 10.1109/ICDAR.2019.00244.

[18] P. Jain, et al., "OCR-based Text Detection and Recognition in Document Images," *Elsevier Procedia Computer Science*, vol. 174, pp. 196-203, 2020, doi: 10.1016/j.procs.2020.06.025.
[19] P. Sahoo, et al., "OCR Performance in Low-Quality Documents," *Journal of Imaging Science*, vol. 64, no. 4, pp. 24-31, Oct. 2018, doi: 10.1016/j.jim.2018.06.009.
[20] R. Smith, "An Overview of the Tesseract OCR Engine," *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, Curitiba, Brazil, 2007, pp. 629-633, doi: 10.1109/ICDAR.2007.4376991.
[21] S. Krishnan, et al., "Language-agnostic OCR for ID Documents," *arXiv Preprint*, arXiv:2101.04213, 2021. [Online]. Available: https://arxiv.org/abs/2101.04213.
[22] Y. Liu, et al., "ABCNet: Adaptive Bezier Curve Network for Scene Text Detection," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2020)*, Seattle, WA, USA, 2020, pp. 2121-2130, doi: 10.1109/CVPR42600.2020.00220.
[23] Y. Sun, et al., "Chinese Text in the Wild," *arXiv Preprint*, arXiv:1905.06142, 2019. [Online]. Available: https://arxiv.org/abs/1905.06142.
[24] Y. Wang, et al., "Efficient Arbitrary-Shaped Text Detection with Pixel Aggregation Network," *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV 2019)*, Seoul, South Korea, 2019, pp. 3702-3710, doi: 10.1109/ICCV.2019.00380.
[25] Z. Zhang, et al., "Robust Text Detection with Contour Awareness," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2021)*, Nashville, TN, USA, 2021, pp. 7001-7009, doi: 10.1109/CVPR46437.2021.00689.
[26] RVRaghavendra Rao, U Srinivasulu Reddy, Sparse attention regression network-based soil fertility prediction with UMMASO, Chemometrics and Intelligent Laboratory Systems, Volume 257, 2025, 105289, ISSN 0169-7439, https://doi.org/10.1016/j.chemolab.2024.105289.