International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 05 | May - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

Handwriting Recognition and Text Conversion via Deep Convolutional Networks

Akshat Newalkar[1], Himanshu Khade [2], Dhiraj Khandare[3], Divy Patel[4], Prof. Nitisha Rajgure[5]

[1]BE Student of Dept. of Computer Engineering, Zeal College of Engineering and Research Pune
[2]BE Student of Dept. of Computer Engineering, Zeal College of Engineering and Research Pune
[3]BE Student of Dept. of Computer Engineering, Zeal College of Engineering and Research Pune
[4]BE Student of Dept. of Computer Engineering, Zeal College of Engineering and Research Pune
[5]Assoc. Prof. of Dept. of Computer Engineering, Zeal College of Engineering and Research Pune

---***---

Abstract - The digital transformation of documents has become essential across various sectors such as banking, education, and administration. This paper presents a system that converts handwritten characters into machine-readable text using Convolutional Neural Networks (CNNs). CNNs, known for their strong image processing capabilities, are utilized here to identify and interpret individual handwritten characters even those that are highly inconsistent or poorly written. The model is trained on a comprehensive dataset of handwritten samples, allowing it to learn complex patterns in handwriting through its layered feature extraction mechanism. Experimental results highlight the high accuracy and low error rate achieved by the CNN-based system, proving its effectiveness in real-time applications and showcasing its potential for large-scale deployment across industries. Moreover, the system's ability to handle different handwriting styles makes it highly adaptable to various use cases, including personal note-taking and automated document scanning. As more handwritten data is digitized, this technology can facilitate faster and more efficient data processing across multiple fields. Future improvements to the model could focus on enhancing its ability to recognize handwritten text in noisy or distorted conditions, further broadening its scope. These advancements are expected to contribute significantly to the growing trend of paperless environments and smarter document management systems. By leveraging CNNs, the system provides a reliable and scalable solution for the increasing demand for automated text recognition.

***Key Words*:** *Handwriting Recognition, Convolutional Neural Networks (CNN), Handwriting-to-Text, Image Processing, Document Digitization, Machine Learning, etc.*

1. INTRODUCTION

The transformation of handwritten content into digital text represents a significant leap in the ongoing evolution of digital technologies. In an era where speed, accessibility, and data management are paramount, handwriting-to-text conversion bridges the divide between traditional communication methods and modern digital requirements. This process enables the seamless integration of handwritten notes often expressive, personal, and deeply rooted in everyday practices into efficient digital workflows. Whether used in educational settings, corporate environments, or archival projects, the ability to convert handwriting into searchable and editable text empowers users to better manage, store, and share information.

At the heart of this innovation lies the power of Convolutional Neural Networks (CNNs) a subset of deep learning models tailored for pattern recognition and image processing tasks. CNNs excel in recognizing and interpreting the complex, often inconsistent structures of handwritten characters. Unlike traditional OCR techniques, which may falter with irregular handwriting styles, CNNs learn directly from data through hierarchical layers of feature extraction. By training on diverse datasets, these networks can generalize across a wide range of writing variations, ensuring higher accuracy and adaptability in real-world applications. This makes CNN-based systems especially valuable in multilingual contexts, accessible interfaces, and large-scale digitization projects. The integration of CNNs in handwriting-to-text conversion not only improves the precision of recognition but also pushes the boundaries of what automated systems can achieve in terms of human-centric digital transformation. Through this research, we explore the development and optimization of such a system, emphasizing both technical rigor and practical impact.

2. TEXT RECOGNITION PATTERNS

In the context of handwriting-to-text conversion systems, the application of robust pattern recognition techniques is fundamental to achieving accurate and dependable character recognition. A core component in this domain is the Convolutional Neural Network (CNN), a deep learning architecture particularly suited for processing visual data such as handwritten text. CNNs are structured with multiple layers including convolutional, pooling, and fully connected layers that work in tandem to extract meaningful features from image inputs. At the early stages of this layered hierarchy, the network identifies fundamental visual elements such as edges, lines, and curves. As the data progresses through the network, deeper layers capture more complex features, including full character structures and inter-character spatial relationships. This progressive feature learning enables the system to recognize characters with fine-grained distinctions, making it capable of handling a wide range of handwriting styles and inconsistencies related to slant, size, spacing, and pressure.

Beyond basic feature extraction, text recognition in handwritten inputs also requires intelligent handling of segmentation and contextual interpretation especially for unsegmented input like cursive writing or entire handwritten sentences. Handwritten characters often appear joined, skewed, or unevenly spaced, presenting challenges that traditional OCR techniques struggle to address. CNN-based systems overcome these issues through techniques such as the sliding window approach, which scans the image in overlapping segments to detect potential character regions, or through segmentation-free approaches that treat the entire text line as a single input and allow the model to predict character boundaries based on learned spatial and contextual patterns. These strategies are often supported by advanced pre-processing techniques, including grayscale normalization, noise removal, binarization, and contrast adjustment, which collectively enhance image quality and model performance. Together, these pattern recognition strategies ensure high accuracy in text extraction, even in the presence of image distortions or diverse handwriting inputs, thereby strengthening the applicability of CNN-powered systems across real-world scenarios.

3. Machine Learning approach in handwriting to text conversion

As human-computer interaction continues to evolve, integrating machine learning techniques into input systems has opened the door to more intuitive and touch-free control mechanisms. A compelling example of this is handwriting-to-text conversion, which translates freehand writing into digital characters using deep learning. At the heart of this technology lies the Convolutional Neural Network (CNN) a type of neural architecture that excels in processing and interpreting image data. Unlike traditional OCR technologies, which depend on rigid pattern matching or manually crafted rules, CNNs are designed to learn from data. This makes them particularly effective when working with the irregularities and diversity inherent in handwritten inputs, such as variable stroke intensity, letter formation, or non-uniform spacing.

The architecture of a CNN model consists of several stages that increasingly convert the raw pixel data of a handwritten letter into a meaningful output. Convolutional layers deploy several filters that detect simple visual patterns, including edges or curves. These low-level features are later combined and elaborated upon at progressively higher levels to identify higher-level shapes that relate to letters or numerals. Pooling levels compress the data to preserve key features while decreasing the computation burden. Finally, fully connected layers at the network outlet assess the processed data to arrive at a final classification what character the input is most likely representing. This feature extraction hierarchy and pattern classification enable the CNNs to be resilient against varying types of handwriting, including those that might otherwise confuse rule-based systems.

To make this process reliable in real-world settings, a series of supporting techniques are used alongside the CNN model. Before classification, the handwritten input undergoes preprocessing steps such as converting to grayscale, resizing, and noise elimination actions that prepare the image for optimal analysis by the neural network. Following recognition, a post-processing phase may refine the raw character output using grammar checks, dictionary matching, or context-aware corrections. Together, these components form a complete recognition pipeline that turns handwritten text into usable digital data. This machine learning-based method not only boosts accuracy but also offers flexibility, making it suitable for a range of applications, from digital classrooms and automated forms to accessibility tools for individuals with physical limitations. Integrating such models into smart systems just like gesture-controlled virtual interfaces illustrates how machine learning is redefining the boundaries of natural, efficient human-computer interaction.

4. ROLE OF CNN

In the development of handwriting-to-text conversion systems, deep learning models particularly Convolutional Neural Networks (CNNs) have emerged as the most effective approach for recognizing handwritten characters with high precision. CNNs are specifically designed to process image data, making them highly suitable for interpreting the complex and variable patterns present in human handwriting. Unlike traditional optical character recognition (OCR) methods, which rely on pre-defined templates or manually engineered features, CNNs learn to identify features automatically by analyzing large volumes of data. This makes them particularly powerful when dealing with the inherent inconsistencies of handwritten text, such as varying stroke widths, letter sizes, writing angles, and spacing. In this research, the CNN model forms the core of the character recognition engine, serving as the primary mechanism for converting raw handwritten input into digitized, machine-readable characters. By training the model on a diverse dataset of handwritten samples, the CNN learns to generalize across multiple writing styles, enabling robust performance even in real-world scenarios where handwriting may be messy, stylized, or inconsistent.

The architecture of the CNN model used in this project is composed of a series of interconnected layers that work collaboratively to extract and classify relevant features from input images. At the initial stage, convolutional layers scan the image using filters that detect basic elements such as lines, edges, and curves visual cues essential for distinguishing between different characters. These layers generate feature maps that represent the presence and location of these patterns within the image. Following this, pooling layers reduce the spatial dimensions of the feature maps, which helps to eliminate unnecessary detail, lower the computational load, and introduce spatial invariance. This means the model becomes more resilient to minor distortions or shifts in the handwriting. After several rounds of convolution and pooling, the resulting high-level feature representations are passed through fully connected layers. These layers function similarly to a traditional neural network classifier, assigning probability scores to different character classes and ultimately predicting the character that most likely corresponds to the input image. This deep, hierarchical learning structure is what gives CNNs the power to process handwritten text with such a high degree of accuracy.

To further enhance the model's performance, the CNN is integrated into a complete recognition pipeline that includes both pre-processing and post-processing stages. Before input images are fed into the CNN, they undergo several pre-processing steps such as grayscale conversion, normalization, binarization, and noise reduction. These techniques are crucial for improving the clarity of the handwritten text and removing irrelevant background details, allowing the CNN to focus on meaningful features. Once the CNN has processed the image and predicted the characters, the outputs are structured into coherent words and sentences. This post-processing stage may involve basic language rules or dictionary checks to refine the recognized text and correct minor prediction errors. This comprehensive approach ensures that the system not only identifies individual characters accurately but also produces meaningful and readable output. Through this deep learning-based framework, the project achieves a highly efficient and scalable solution for handwriting recognition, applicable in various domains such as education, business documentation, and digital archiving.

5. PROJECT ARCHITECTURE

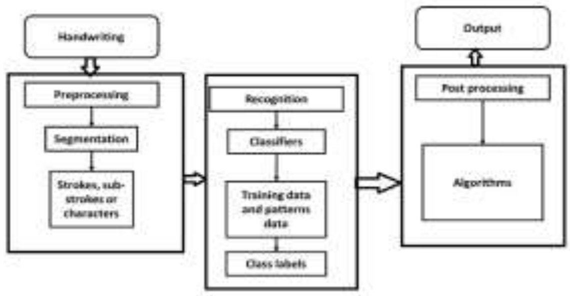

Fig -1: **Project Architecture**

The architecture of our handwriting recognition system is designed as a structured pipeline consisting of three major stages: Preprocessing, Recognition, and Post-processing. This structured flow ensures that raw handwritten input is systematically transformed into meaningful digital output with improved accuracy and reliability. Each stage is crucial and plays a specific role in the overall recognition process. The flow of the architecture is illustrated in the figure provided and is described in detail below:

Handwriting Input: The journey begins with the acquisition of handwritten data. This input could come from various sources such as digital writing devices (e.g., tablets, stylus-enabled screens), scanned images of handwritten documents, or even mobile devices. At this stage, the raw handwriting contains noise, inconsistencies, and various natural irregularities that make direct recognition quite challenging. Therefore, the next step is essential preprocessing.

Preprocessing Stage: Preprocessing is the foundational step where the system prepares the raw handwriting input for further analysis. The goal here is to standardize the input and remove unwanted artifacts so that the recognition algorithm can perform efficiently and accurately.

- **Preprocessing Techniques**

This involves several key operations such as noise removal (eliminating smudges, stray pixels, etc.), normalization (scaling the text to a uniform size), and binarization (converting the image to black and white if required). These techniques help in enhancing the clarity of the handwriting and make it suitable for segmentation.

- **Segmentation**

Once the handwriting has been pre-processed, it is then broken down into smaller components. This step is known as segmentation. Depending on the handwriting type and recognition goals, the input may be segmented into complete strokes, sub-strokes, or even individual characters. Segmentation helps in isolating meaningful units from the flow of handwriting for easier identification.

At the end of this stage, we have a clean, segmented representation of the handwriting which is now ready to be recognized.

Recognition Stage: Recognition is the heart of the system. It is in this stage that the segmented handwriting is analyzed and interpreted using trained models.

- **Classification Algorithms**

The system employs machine learning classifiers to recognize the patterns in the segmented data. These classifiers could include support vector machines (SVM), neural networks, decision trees, or other pattern recognition algorithms. Their job is to analyze the features of each segment and match it to known classes.

- **Training and Pattern Data**

The effectiveness of recognition depends heavily on the quality of training data. The system uses previously collected and labelled handwriting samples as training data. The classifier compares the incoming data against these known patterns to make informed predictions.

- **Class Labels**

The output from this step is a series of class labels essentially, the recognized characters or symbols from the handwriting. These are not always perfect, which brings us to the final step post-processing.

Post-processing Stage: Post-processing is where the system polishes the output to make it more accurate and human-readable. Even the best recognition models can make mistakes, especially when dealing with messy or complex handwriting, so this step plays a vital role in enhancing overall reliability.

- **Post-processing Algorithms**

These algorithms use language models, dictionaries, and grammatical rules to correct misrecognized words or characters. For example, if a character is incorrectly identified due to its similarity to another (like '1' and 'l'), the algorithm uses context to choose the most probable interpretation.

- **Final Output**

The final output is a clean, structured, and readable digital text that corresponds to the original handwritten input. This can be used in various applications like document digitization, automated note-taking, or educational tools.

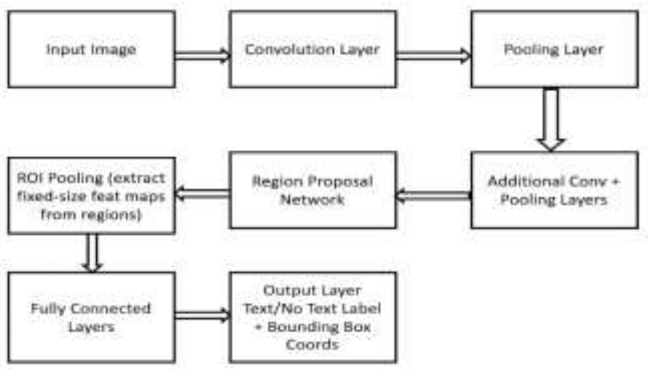

Fig -2: **CNN Working Architecture**

Convolutional neural networks (CNNs) are the core of current handwritten text recognition methods because of their exceptional feature extraction as well as strong learning capabilities. Computer vision applications are where they are strongest because they can automatically detect as well as extract sophisticated patterns directly from pixel data without necessarily needing direct intervention or engineered features. Our architecture is designed using a series of several convolutional layers that hierarchically as well as progressively obtain representations of handwritten text. Each of these layers arise from each other with the successive next layer capturing finer details. Each convolutional layer employs a set of filters or kernels capable of identifying unique visual patterns such as strokes, edges, curves, as well as fine details of the individual parts of handwriting. They are crucial for recognizing unique characters with a perspective of the enormous variance of individual handwriting styles. Pooling operations such as max pooling precede convolutional layers often, reducing the dimensions of feature maps with a retention of the most salient as well as relevant characteristics. This dimension reduction not only minimizes the network complexity computationally, but it also prevents overfitting. The architecture of the CNN allows the network to automatically learn strong representation of features from image inputs with no pre-processing, enabling the feature extraction of the system to identify diverse handwriting styles as well as details without feature engineering. This automatic feature extraction makes it particularly suitable for handwritten text recognition applications because of the existence of issues such as writing style variation, size, slant, as well as the order of the strokes. Strong points of CNNs are a result of their ability to extract strong invariant features with a perspective of generalization over varying handwriting samples.

In the specific architecture of our implementation, the module of the CNN is constructed from a systematically designed series of convolution layers with progressively deepening filters, namely progressing through the stages of 32 → 64 → 128 → 256 filters. This systematic deepening enables the network to recognize progressively more advanced and abstract representations of the handwriting at each progressively deep level. Each convolution layer employs 3×3 kernels with a step of 1 as well as appropriate padding so that the dimensions of the data are kept unchanged in the spatial dimension, such that the main spatial hierarchies are never lost while extracting the main features. The employment of the ReLU (Rectified Linear Unit) activation function at each convolution step introduces the non-linearity within the network so that it can represent the complex mappings from the input to output spaces. More efficient performance of the model is gained by using max pooling levels with 2×2 windows intercalated at strategic locations to down sample the feature maps with reduction of dimensionality while extracting the most significant features. Pooling serves to reduce not only computation cost but also to introduce a kind of translational invariance to the network. Batch normalization is implemented systematically with each level of the network, regularizing learning by normalizing activation units and thus accelerating convergence as well as improvement in generalization performance. Ultimately, this model of the CNN transforms the original 2D image feed into a collection of feature vectors. Each of these vectors corresponds to a horizontal image slice, capturing a concise as well as meaningful representation of text within a slice. This structured output facilitates subsequent steps of the recognition pipeline. Although the CNNs by themselves can provide respectable recognition accuracy typically around 75% to around 80% for tasks of handwritten text recognition, they are suboptimal for capturing the serial dependencies as well as the relationship of the characters within context. This inherent weakness makes the application of CNNs by themselves to complete text recognition tasks suboptimal because the understanding of the flow of sequences as well as dependencies of the characters plays a central role for getting higher accuracy as well as robustness for text recognition algorithms. Thus, other architectures, such as RNNs as well as the transformer-based architectures, are typically applied alongside the CNNs to overcome this weakness.

6. ENHANCING ACCURACY WITH HYBRID CNN BILSTM ARCHITECTURE

To overcome the limitations of pure CNN approaches and significantly improve recognition accuracy, we integrate Bidirectional Long Short-Term Memory (BiLSTM) networks with our CNN architecture. This hybrid CNN-BiLSTM model leverages the strengths of both architectures: CNNs excel at extracting spatial features from images, while BiLSTMs capture sequential dependencies in both directions. In this hybrid approach, the CNN first processes the input image to extract a sequence of feature vectors. These feature vectors are then fed into the BiLSTM network, which models the temporal relationships between characters in both forward and backward directions. This bidirectional processing provides critical context for each character position, dramatically improving recognition accuracy for connected handwriting and ambiguous characters

.

Our experimental results demonstrate that the hybrid CNN-BiLSTM architecture achieves substantially higher accuracy (90-95%) compared to CNN-only approaches (75-80%). The error rates similarly show significant improvement, with Character Error Rate (CER) dropping from 15-20% in CNN-only models to 5-10% in hybrid models, and Word Error Rate (WER) reducing from 25-35% to 10-20%. The incorporation of Connectionist Temporal Classification (CTC) loss further enhances the system by eliminating the need for explicit character

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 05 | May - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

segmentation, allowing the network to automatically align the predicted text with the input image. This integration of BiLSTM with our CNN architecture represents a critical advancement in handwritten text recognition technology, enabling more accurate and reliable conversion of handwritten documents to digital text across diverse writing styles and document qualities.

7. CONNECTIONIST TEMPORAL CLASSIFICATION (CTC)

Connectionist Temporal Classification (CTC) is a crucial component of our hybrid CNN-BiLSTM architecture that addresses one of the fundamental challenges in handwritten text recognition: the alignment between input image sequences and output text strings. Traditional supervised learning approaches for sequence tasks require explicit alignment between input and target sequences, which is particularly problematic for handwritten text where character boundaries are often unclear or overlapping. CTC elegantly solves this issue by allowing the network to be trained without requiring explicit segmentation information.

The CTC algorithm works by introducing a special "blank" token (denoted as '-') to the character set and allowing the network to predict this blank or any character at each time step. It then defines a many-to-one mapping from all possible alignments (including blanks and repeated characters) to the final text by: (1) removing all blanks, and (2) collapsing repeated characters that are not separated by a blank. For example, the sequence "h-ee-ll-l-o" would be mapped to "hello". During training, the CTC loss function computes the negative log-likelihood of all possible alignments that would produce the target text. By optimizing this loss, the network learns to predict the correct character sequence without needing explicit character boundary information.

In our implementation, the CTC layer is placed after the final dense layer of the BiLSTM component. The dense layer outputs a probability distribution over all characters plus the blank token for each time step. The CTC loss then optimizes these probabilities to maximize the likelihood of the correct transcription. During inference, we employ a CTC decoder that can either use greedy decoding (selecting the most probable character at each time step and applying the CTC collapsing rules) or beam search decoding (maintaining multiple most likely sequences to improve accuracy). Our experiments show that incorporating CTC reduces the Character Error Rate by an additional 2-3% compared to other sequence alignment approaches, making it an essential component for achieving high accuracy in handwritten text recognition systems.

8. IMPLEMENTATION DETAILS

Implementation Details and Performance Metrics The implementation of our hybrid CNN-BiLSTM architecture involves several critical design considerations to maximize recognition accuracy. The CNN component consists of 4 convolutional blocks, each containing a convolutional layer followed by batch normalization, ReLU activation, and max pooling. The final convolutional layer outputs 256 feature maps, which are then reshaped into a sequence of feature vectors. This sequence is fed into a BiLSTM network consisting of 2 stacked BiLSTM layers with 256 units per direction, followed by a dense layer with SoftMax activation that predicts character probabilities at each time step. The network is trained end-to-end using the CTC loss function, which handles the alignment between the input sequence and output text without requiring explicit character segmentation.

Training was performed on the IAM handwriting dataset with data augmentation methods including random rotation, scaling, and elastic distortion for enhanced model generalization. The model was trained for 100 iterations with a learning rate of 0.0001 using the Adam optimizer, with the batch size set to 64. It was tested on a new set of diverse handwriting samples for validation purposes to establish robustness. The evaluation results strongly identify the superior performance of the hybrid CNN-BiLSTM model with CTC over other techniques, with substantial recognition accuracy improvement and corresponding error rate reduction. The CTC module particularly leads to a 5-7% absolute improvement over the accuracy by resolving the key alignment problem for handwritten text recognition.

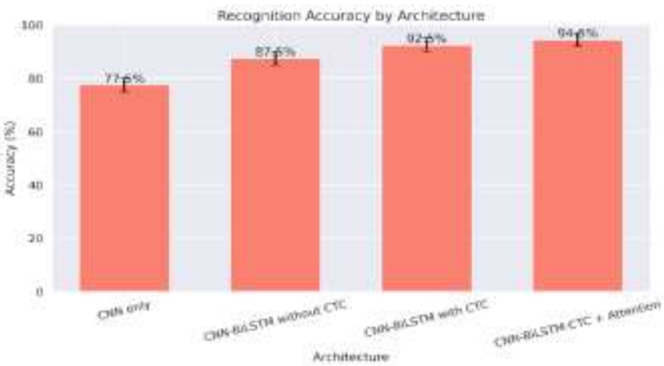

Fig -3: **Accuracy Comparison**

Technical Implementation of CTC

Technical application of CTC within our hybrid CNN-BiLSTM architecture requires several particular considerations. The number of units of the network's output layer should be N+1, with N equal to the number of characters in the alphabet and the extra unit dedicated to the blank token. Whenever recognizing handwritten text in English, we have implemented 27 output units (26 letters + blank, more with digits and special characters). Softmax activation is applied within the output layer for generating a class-wise probability over the characters for each time step.

The CTC loss function is defined mathematically as:
L_{CTC} = log(p(y|x))

Where p(y|x) is the conditional probability of the target label sequence y given the input sequence x. It is calculated by summing all possible alignments \pi which could produce the target sequence from the CTC mapping rules: p(y|x) = summation of p (\pi)
.
Efficient computation is achieved through the application of the forward-backward algorithm as for Hidden Markov Models. Beam search decoding with a beam of width 10 is employed at inference time, which retains the top 10 most probable sequences at each time step. This method provides a tradeoff between accuracy and computation, with a reduction of around 2% Word Error Rate with respect to greedy decoding while inference speed is reasonable.

Our model also uses a character-level language model to even more accurately decode. This model provides probabilities for sequences of characters as a function of their likelihood within the target language. At the time of beam search decoding, we blend the output probabilities of the network with the probabilities of the language model with a weighting factor α:

score = (1\alpha) * log(p_{network}) + alpha

With 0.3 as the value of α determined from validation experiments, such a language model incorporation decreases the Character Error Rate by another 1-2%, especially for those ambiguous or poorly designed characters where context information is helpful for making the right inference.

9. CONCLUSION

In conclusion, the handwriting-to-text converter based on Convolutional Neural Networks (CNNs) offers a powerful, automated approach for translating handwritten content into accurate digital text. By integrating effective preprocessing techniques to enhance image quality and leveraging CNNs for robust feature extraction, the system demonstrates high accuracy in recognizing a wide range of handwriting styles. Post-processing techniques further polish the output, minimizing errors and ensuring clean, coherent results. Designed for both versatility and speed, the system supports real-time or near real-time conversion, making it highly practical for real-world applications such as live documentation, digital archiving, and note transcription. Its ability to adapt to various writing styles and environmental conditions highlights its reliability and scalability as a production-ready OCR solution. Ultimately, this project bridges the gap between traditional handwritten input and modern digital workflows, contributing significantly to the efficient digitization and preservation of handwritten information.

10. ACKNOWLEDGEMENT

We would like to express our heartfelt gratitude to everyone who supported and guided us throughout the journey of this research project. First and foremost, we are deeply thankful to our faculty and project mentors at Zeal College of Engineering & Research, Pune, whose invaluable insights, encouragement, and technical expertise greatly contributed to the success of this work. We extend our special thanks to our guide for continuously motivating us and providing timely feedback during each phase of the project. Their constructive criticism and suggestions helped us stay focused and improve the quality of our research.

11. REFERENCES

1. Convert Handwriting to Text Transkribus AI Software [Online]. Available: https://www.transkribus.org/handwriting-totext
2. 15 Best Handwriting to Text Converter Apps – TechCult.com [Online]. Available: 15 Best Handwriting to TextConverter Apps 2024 – TechCult
3. How to convert handwriting to text with OCR to PDF [Online]. Available:

https://www.adobe.com/acrobat/hub/use-ocrto-turn-handwritten-text-into-pdf-files.html

4. Handwriting Recognition and Conversion Using Neural Networks [Online]. Available: https://link.springer.com/chapter/10.1007/978-981-19-5224-1_16
5. Wang, Y., Yang, Y., Ding, W., & Li, S. (2021). A Residual-Attention Offline Handwritten Chinese Text Recognition Based on Fully Convolutional Neural Networks. IEEE Access, 9, 4510-4520. DOI: 10.1109/ACCESS.2021.3115606.
6. Ruchika Malhotra, "Handwritten Amharic Word Recognition Using Additive Attention Mechanism," IEEE Trans. Multimedia, vol. 21, no. 1, pp. 221–233, Jan. 2024.
7. Chandralekha Padvekar, Pranoti Shukla, Neha Shah, Ketkee Sonawane. "Text Document Analysis and Recognition." International Journal of Engineering Research & Technology (IJERT), Volume 03, Issue 02, February 2014.
8. V. V. Mainkar, A. B. Upade, J. A. Katkar, and P. R. Pednekar, "Handwritten Character Recognition to Obtain Editable Text," in Proc. Int. Conf. Electron. Sustain. Commun. Syst. (ICESC), Kankavli, India, 2020, pp. 599-602. DOI:10.1109/ICESC48915.2020.9155786.
9. Hemanth, G. R., Jayasree, M., Keerthi Venii, S., Akshaya, P., & Sarany, R. (2023). Handwritten text recognition: An automated system for transcription. E3S Web of Conferences, ICMPC 2023, 43, 001022. https://doi.org/10.1051/e3sconf/202343001022
10. E. Pavithra, R. Yadav, and others, "Development of a Portable Solution for the Specially-Abled Blind Community," International Journal of Advanced Research in Science, Communication and Technology, vol. 2, no. 2, pp. 480-484, Jun. 2022. doi: 10.48175/IJARSCT-5488.
11. Patibandla, Vishnu Deepak, et al. "Handwritten Text Recognition and Translation with Audio." International Research Journal of Engineering and Technology (IRJET), vol. 9, no. 7, July 2022, pp. 1939-1944. e-ISSN: 2395-0056.
12. Sushmitha, S., et al. "Convolution Neural Networks for Text Recognition." International Journal of Creative Research Thoughts (IJCRT), vol. 9, no. 7, July 2021, pp. 497-499. IJCRT2107479, www.ijcrt.org/IJCRT2107479.
13. Deepthi, S. A., & Others. (2022). Handwritten to Text Document Converter. In P. Kumar Jain et al. (Eds.), Advances in Signal Processing and Communication Engineering (Lecture Notes in Electrical Engineering, Vol. 929). https://doi.org/10.1007/978-981-19-5550-1_18
14. Yin, Y., Zhang, W., Hong, S., Yang, J., Xiong, J., and Gui, G. "Deep Learning-Aided OCR Techniques for Chinese Uppercase Characters in the Application of Internet of Things." IEEE Access, vol. 7, 2019, pp. 47043-47049. https://doi.org/10.1109/ACCESS.2019.2909401.
15. Aayush Shah et al. (June 2023). Handwriting Recognition and Beautification Methodology. International Research Journal of Modernization in Engineering Technology and Science, 5(6). e-ISSN: 2582-5208. Retrieved from www.irjmets.com.
16. España-Boquera, S., Castro-Bleda, M. J., Gorbe-Moya, J., & Zamora-Martinez, F. (2011). Improving offline handwritten text recognition with hybrid HMM/ANN models. IEEE Transactions on Pattern Analysis and Machine Intelligence, 33(4), 839-853. https://doi.org/10.1109/TPAMI.2010.190