Vol. 08, No. 2 (2026) 1051-1062, doi: 10.24874/PES08.02A.018

Proceedings on Engineering Sciences

www.pesjournal.net

PRESCRIPTION DIGITISATION USING BI-LSTM AND CTC-ENHANCED NEURAL NETWORKS

Garima Saroj [1]
Ayushi Gautam
Bhavya Agrawal
Sarthak Gupta

Received 28.09.2025.
Revised 21.01.2026.
Accepted 16.02.2026.

Keywords:

Bidirectional Long-Short Term Memory (Bi-LSTM), Connectionist Temporal Classification (CTC) loss, Sequence modelling, Handwriting recognition, Optical character recognition (OCR), Deep learning

A B S T R A C T

Handwriting is a way to convey an idea or information through written means. Over the decades, due to growing population density, doctors have been known for their often rushed and difficult-to-read handwriting. This readability issue of handwritten medical prescriptions, especially those created by clinicians, has an underlying potential to cause serious problems, ranging from confusion and delays in proper medication, to harmful side-effects. The aim of this research is to explore how deep learning techniques can help to reduce such risks by developing a model capable of reading and converting handwritten medical notes into digitised text. The model is based on a Region-based Convolutional Recurrent Neural Network (R-CRNN) and incorporates Connectionist Temporal Classification (CTC) loss which allows the model to recognise the handwritten characters, even in distorted or cursive writing. A combination of handwritten prescription samples and publicly available handwritten datasets were collected and annotated, preprocessed, and then the model training was performed. The research also aims to contribute towards the improvement of healthcare documentation and enhance the accessibility of medical records for people's well-being and a sustainable future.

© 2026 Published by Faculty of Engineering

1. INTRODUCTION

The art of handwriting articulates thoughts and ideas, with doctors being renowned for their indecipherable handwriting, a phenomenon widely acknowledged. This challenge also extends beyond medical practitioners, affecting pharmacists associated with interpreting prescriptions, resulting in a series of medical errors. A documented case in Texas highlighted the dire consequences, where a cardiologist's prescription for *10 mg Plendil* was misread as *20 mg*, causing a tragic patient fatality, discussed by Brits et al. (2017) and Cerio et al. (2015). The intricacies of recognising human handwriting have spurred technological advancements, particularly in the development of handwriting recognition systems. Another such incident, widely reported in Indian media, involved a young child in Mumbai, who was mistakenly administered *Digene*, which is an antacid, instead of *Diligan*, which is an antihistamine, after a pharmacist misread the doctor's handwritten prescription. The child experienced adverse reactions and side-effects, and the case very well

[1] Corresponding author: Garima Saroj
Email: garimak9@protonmail.com

highlighted the risks associated with illegible prescriptions and the growing need for digitised systems in healthcare practice (Roy et al., 2006). In another reported case, unclear and confusing handwritten prescriptions resulted in patients receiving incorrect medications because of confusion between similarly written drug names, further emphasising the risks associated with manual prescription systems.

Although artificial intelligence and text recognition systems have improved considerably in recent years, recognising cursive and unstructured handwriting remains a difficult task. Despite existing research in this domain, the challenge of recognizing cursive characters persists, owing to factors such as deformations, size variations, diverse styles, incomplete strokes, flow and Ligatures. The problem becomes more and more complicated in medical settings, wherein shorthand notation, abbreviations, and inconsistent writing styles are commonly used by medical practitioners.

In this research, we propose a deep learning based approach for recognising as well as digitising doctor's handwritten prescriptions. The model we propose is built using a Region-based Convolutional Recurrent Neural Network (R-CRNN) and incorporates a Connectionist Temporal Classification (CTC) loss function. This architecture allows the model to process and transcribe handwritten text without requiring manually aligned character-level annotations.

The aim of this research is to address the problems caused by illegible handwritten prescriptions and ultimately contribute towards better healthcare documentation. By automating handwritten text recognition, the proposed model can help medical professionals convert handwritten notes into digital records with reduced chances of interpretation-related errors and in turn help avoid its harmful consequences.

The proposed model uses combined methods from computer vision, CNN-based feature extraction, and sequence modelling to process handwritten prescriptions more effectively, helping in the recognition of difficult handwriting patterns.

2. LITERATURE REVIEW

Handwriting recognition is all about processing handwritten text from images or PDFs by converting it into segments and then transforming it into something a computer can understand, which we generally refer to as a digitised or machine-readable form. This becomes even more difficult when the model has to deal with cursive, overlapping, uneven, and highly unstructured handwritings, especially in the case of medical prescriptions.

Optical Character Recognition (OCR) methods usually struggle with such inputs because they mostly depend on explicit word segmentation and character segmentation, which does not perform well on connected or free-flowing handwriting styles (Bhushan et al., 2001).

In this research, the Convolutional Neural Network (CNN) layers extract significant spatial features directly from the prescription images, whereas the Recurrent Neural Network (RNN) layers, specifically the Bidirectional Long-Short Term Memory (Bi-LSTM), provided a way to understand the characters as part of a sequential pattern rather than, as isolated characters.

Handwritings in the medical field are particularly known to be difficult to process because doctors commonly use shorthand notations, abbreviations, and rapidly written strokes while writing medical prescriptions. Although publicly available datasets such as Interface Analysis and Management (IAM) Handwriting Database provide a really strong foundation for handwriting recognition research, they could not fully represent the domain-specific vocabulary, usual medical terms and writing styles commonly observed in medical documents. To address this limitation, researchers such as Kizilirmak and Yanikoglu (2023) explored various techniques including data augmentation and domain adaptation to improve handwriting recognition performance across more varied handwriting conditions.

Nisa (2023) demonstrated how synthetic handwriting data could help improve model generalisation across different writing styles. Fajardo et al. (2019) and Hassan et al. (2021) have also explored deep learning approaches specifically for recognising cursive medical handwriting. These ideas have positively influenced several preprocessing and augmentation decisions that we made during the development of our pipeline.

2.1. Region-based Convolutional Recurrent Neural Networks (R-CRNN)

Recent work by LeCun et al. (2015) established the foundations of deep learning techniques such as CNNs, which are widely used for spatial feature extraction from images. Similarly, Hochreiter and Schmidhuber (1997) introduced the LSTM architecture for handling sequential data such as handwritten text.

The R-CRNN architecture combines CNN layers for image feature extraction, Bi-LSTM layers for sequential pattern recognition, and a transcription layer for text prediction. We used bidirectional LSTM so the model can look at both past and future context in a word, which really helps when decoding messy or cursive writing (Zhao, 2022). Another similar CRNN-based handwriting recognition frameworks were also explored by Yin et al. (2021).

The CNN block works by sliding filters over the image. A basic operation, adapted from convolutional neural

network formulations (Alzaga et al., 2025; Zhao, 2022), looks like this:

$$F_k = \sigma\left(\sum_m \sum_n W_{mn}^k X_{(i+m)(j+n)} + b_k\right)$$

where W_k are the filter weights, b_k is the bias term, and σ is the activation function (typically ReLU).

The features go into Bi-LSTM layers that operate on sequences over time. The following equations (Graves et al., 2006) show how the gates and states are calculated at each timestep:

$$\begin{gathered} i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \\ f_t = \sigma\left(W_f x_t + U_f h_{t-1} + b_f\right) \\ o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \\ \widetilde{C_t} = \tanh(W_C x_t + U_C h_{t-1} + b_C) \\ C_t = f_t \odot C_{t-1} + i_t \odot \widetilde{C_t} \\ h_t = o_t \odot \tanh(C_t) \end{gathered}$$

Wherein the input gate i_t decides what new information to store, the forget gate f_t decides what previous information to discard, and the output gate o_t controls what part of the memory is output at each step. The cell state C_t acts as long-term memory that stores important information across time steps, and the hidden state h_t is the final output of the LSTM block at each time step passed forward in the sequence. This structure is particularly useful for capturing sequential dependencies across handwritten character patterns.

2.2 Connectionist Temporal Classification (CTC) Loss

CTC loss has always been particularly useful in handling unsegmented data with variable length. It was formally introduced by Graves et al. (2006) and has since been widely applied in most end-to-end recognition systems (Bluche, 2016). We used CTC loss, so the model does not require perfectly labelled character boundaries during the training process. Instead, it learns the most probable alignment between the input sequence and the target text automatically. The loss, as defined by Graves et al. (2006), is calculated as:



Figure 1. Illustration of Sequence modelling using CTC

$$\mathcal{L}_{\mathcal{CTC}}(X, Y) = -\log P\,(Y \mid X)$$

where X is the input feature sequence that we extracted from the handwritten image, Y is the target text, and $P(Y \mid X)$ is the the probability of predicting the correct text sequence Y from the given input X. The CTC loss function thus works by minimising the difference between the predicted sequence and the actual handwritten text during the training phase.

CTC loss operates with the ability to operate without the need for pre-segmentation of the input data into identifiable units (e.g., individual letters or words in handwriting recognition). Because it directly learns the mapping between the input sequences and their corresponding labels from the neural network's training process. This direct learning approach, along with the backpropagation algorithm, makes the model automatically and efficiently learn the complex patterns and dependencies of the training dataset.

In context to handwriting recognition, CTC's approach to sequence learning has always been invaluable. It enables the neural network model to handle the intrinsic variability and idiosyncrasies of human handwriting, including variations in character size, style, and spacing. Figure 1 illustrates how CTC loss is used for sequence modelling in reading and recognising handwritten texts. By learning to recognize patterns across the entire sequence of input data rather than relying on pre-defined segmentation, CTC-equipped models can achieve higher levels of accuracy and generalisation in translating handwritten text into digital format.

2.3 Broader applications and trends

Recent case studies, including lightweight Transformer-based architectures such as MSdocTr-Lite (Dhiaf et al., 2023), have performed well on handwriting recognition tasks within healthcare applications. However, in resource-limited environments and at comparatively smaller research scales, R-CRNN model architecture still perform well because it combines CNN's spatial feature extraction with LSTM's sequential modelling capabilities.

Artificial intelligence and machine learning for text and handwriting recognition is a field that is continuously evolving, but several challenges still remain, such as limited annotated datasets and extreme variations in handwriting styles across different regions and individuals. Some of these challenges were also discussed in the review presented by Singh et al. (2024). This study therefore combines the R-CRNN architecture with CTC loss to address one of the challenges in here, specifically, the difficulty of recognising handwritten prescription text without requiring character-by-character segmentation. The model is trained to process handwritten prescription patterns by combining image preprocessing, sequential modelling, and medical vocabulary handling to improve the overall recognition consistency.

3. METHODOLOGY

This section describes the whole workflow of the research, starting with the collection and annotation of datasets, followed by the preprocessing pipeline, the model architecture, training setup, and finally, the evaluation methods used for developing the prescription digitisation model (as displayed in Figure 2). The handwritten prescription samples were manually collected and combined with the IAM Handwriting dataset. Each image was manually annotated using the VGG Image Annotator (VIA) tool. Data preprocessing included grayscale conversion, resizing, noise removal, thresholding, and normalisation. Data augmentation techniques were also applied to introduce greater variation within the training data. The model was developed using TensorFlow and Keras to build the R-CRNN network along with Bi-LSTM layers, and was trained using CTC loss, this helped the model process the handwritten text without requiring explicit character-by-character segmentation. The model performance was evaluated using accuracy, precision, recall, F1-score, Character Error Rate (CER), and Word Error Rate (WER).

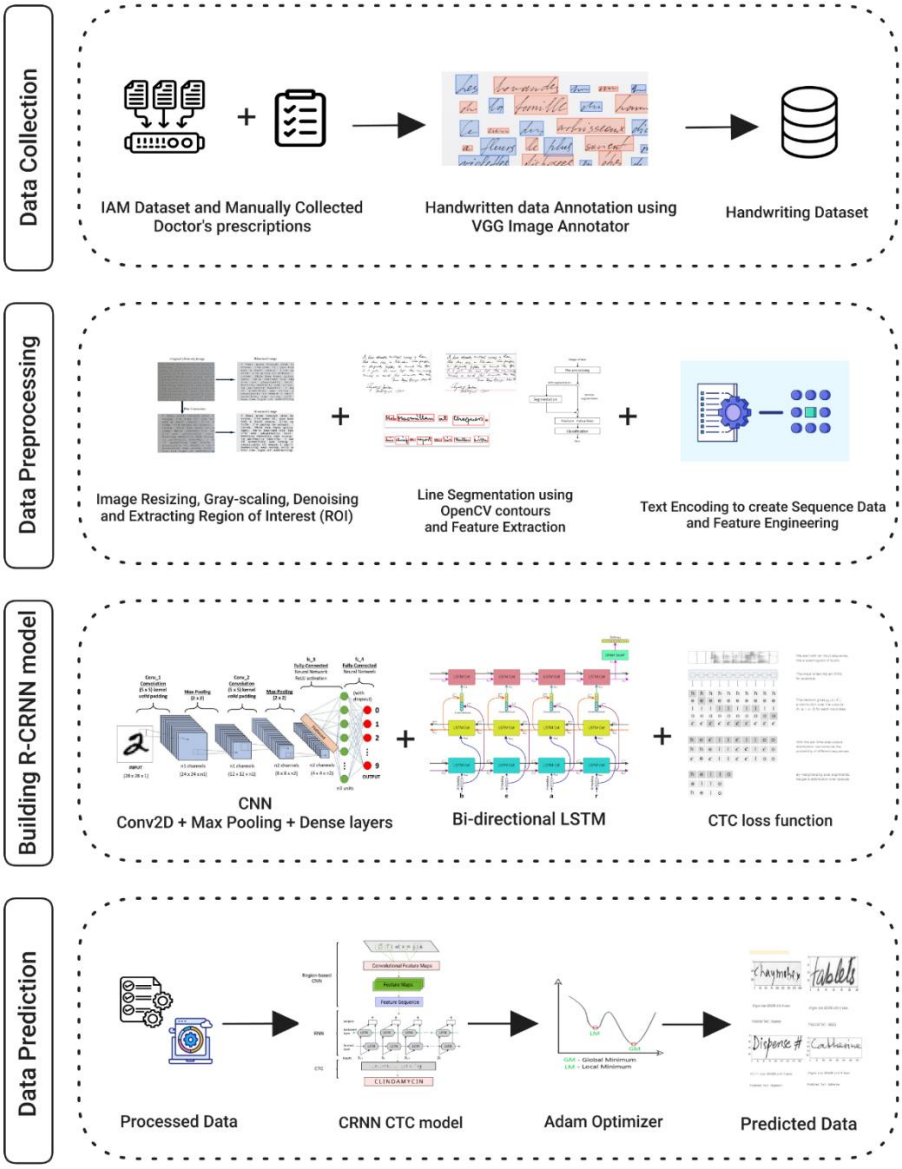


Figure 2. Systematic model workflow comprising of the Data Collection, Preprocessing, model building, and prediction stages

3.1 Dataset Collection and Annotation

During the early stages of data collection, hospitals and healthcare institutions were approached for obtaining the handwritten prescription samples for training the model. However, due to privacy concerns and strict data protection policies, these requests were declined. The prescription samples were therefore collected manually from local pharmacies and individual medical practitioners, while patient-related information was

carefully removed to maintain patients' data confidentiality.

The quality and orientation of the prescription images had considerable variation because they were collected from different sources. A fraction of dataset contained images with shadows, uneven lighting, overlapping text, or partially blurred prescription content. To address these inconsistencies, preprocessing steps such as grayscale conversion, resizing, thresholding, and noise filtering were included before starting the data training. These preprocessing steps also helped in improving text visibility and reducing any unwanted background artefacts that could negatively affect feature extraction during the training.

The IAM Handwriting Forms Dataset was chosen to increase the variation in handwriting styles within the training dataset, the source of this database was developed by the University of Fribourg in Switzerland (Marti & Bunke, 2002). The dataset, as described by IAM, contains unconstrained handwritten English texts scanned at a 300 dpi and stored as grayscale images in PNG file formats. The forms were grouped according to individual writers, which made the dataset useful for handwriting recognition and writer-based analysis tasks. The inclusion of IAM database samples added more variation in the writing styles and also helped the model generalise better according to different handwriting patterns (Nisa, 2023).

A custom medical corpus of common medicine names, dosage units, and frequently used abbreviations found in handwritten prescriptions was created to improve the model's handling of domain-specific medical terminology. All the collected images were manually annotated using the VGG Image Annotator (Visual Geometry Group, n.d.), which is an open-source annotation tool developed by the Visual Geometry Group at the University of Oxford. VIA allowed accurate marking of text regions and corresponding labels, resulting in a manually annotated dataset containing over 2,000 prescription samples.

During data annotation, we were very careful with medicine names, dosage instructions, shorthand notations, and commonly repeated prescription patterns. We also consulted local pharmacies and a few doctors to make sure that the annotations assigned to the handwritten text were appropriate and medically relevant. In some cases, handwritten words were too difficult to interpret even manually because of highly cursive writing or overlapping characters. To address this issue, we incorporated a small medical corpus containing medicine names, abbreviations, and prescription-related terminology associated with the collected samples. And the samples with completely unreadable content or with severe image distortion were completely excluded from the final dataset to maintain annotation consistency and reduce incorrect labelling during training.

3.2 Image Preprocessing and Augmentation

To prepare the collected and well annotated prescriptions for training phase, it was decided to follow a structured data preprocessing pipeline, the pipeline included image segmentation, transformation, and augmentation. The data preprocessing pipeline was designed to standardise the input images and help the model deal with different handwriting styles during the training.

After the preprocessing, image segmentation was done to isolate the handwritten text regions from the background using bounding boxes marked during the annotation phase. Then image segmentation was performed on three levels, line segmentation, word segmentation, and finally character segmentation with shrinking bounding boxes at each transition level as shown in Figure 3. This helped the model focus on the relevant handwritten regions while improving the overall learning process during training.

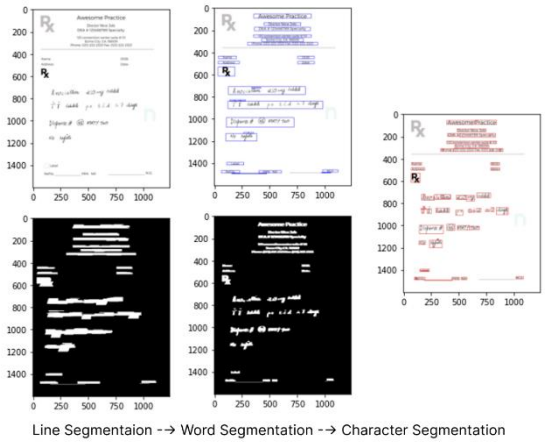


Figure 3. Image preprocessing and Segmentation

Once the image segmentation was achieved, the images were resized to a fixed dimension of 256×64 pixels to maintain consistency in the input format throughout the dataset. Gaussian blurring and median filtering were also applied to reduce background noise, smudges, and ink artefacts that were present in handwritten prescription samples. Adaptive thresholding was additionally used in a specific fraction of data samples to make the faint or broken handwritten strokes clearly visible before training. And finally, the pixel values of the images were normalised to a value between 0 and 1, before being passed into the model.

Data augmentation was performed using a custom DataGenerator class so that the model could train on different handwriting styles. Image augmentation techniques such as small random rotations, zooming, horizontal and vertical shifts, and slight distortions were applied to selected parts of the dataset wherever required. These changes were added to simulate the natural variations present in handwritten prescription samples.

After preprocessing and augmentation, the dataset was divided into training, validation, and testing sets using an 80:10:10 split ratio.

3.3 Model Architecture

Prior to finalising the proposed R-CRNN architecture for the research, a few more deep learning architectures for the handwritten prescription recognition were tested, as shown in Table 1:

1. CNN with Dense Classifiers:This architecture used CNN layers followed by dense classification layers and extracted basic handwriting features from the prescription images. However, the model gave more incorrect outputs while processing connected and cursive handwritten text because it did not capture the character sequence properly. Thus, the CER and WER values remained much higher during testing.
2. Vanilla RNN / LSTM Model:The vanilla RNN/LSTM model performed slightly better than the CNN + dense layers model because it could process sequential information within handwritten text. The model showed improved recognition of character sequences, although longer handwritten samples, inconsistent spacing, and variations in writing style still affected the overall performance during testing.
3. Standard CRNN Model: The standard CRNN model combined convolutional feature extraction with Bi-LSTM based sequence modelling. Compared to the CNN and vanilla RNN architectures, it was better suited for handwritten text recognition as it could learn both visual features and character sequence information simultaneously. This was reflected by the reduction in both CER and WER values.
4. Proposed CRNN + CTC Model: The proposed architecture incorporated CTC loss in addition to the CNN and Bi-LSTM layers. Unlike the previous models, explicit character level alignment was not required during training, making it more suitable for handwritten prescription samples where character boundaries were often unclear. Among all the architectures tested, this model produced the most consistent recognition results and achieved the lowest CER and WER values.

Table 1. Comparative Analysis of Model Architectures

Approach	Components	CER (%)	WER (%)	Training Time (per epoch)	Cursive Support
CNN + Dense	CNN + Fully Connected	22.8	35.4	~12s	Poor
Vanilla RNN	Basic RNN / LSTM	18.3	29.1	~10s	Weak
CRNN	CNN + Bi-LSTM	13.6	20.9	~18s	Moderate
CRNN + CTC (Proposed)	CNN + Bi-LSTM + CTC Loss	9.7	15.2	~22s	Strong

The proposed model comprises of three main blocks, firstly, the CNN layers for extracting spatial features from the input images. Two convolutional layers were created, each with 256 filters of size 3×3 and ReLU as activation function. Followed by MaxPooling2D layers with a 2×2 window, which help to reduce spatial dimensions while preserving critical visual information. The output of these layers formed a condensed feature map, which was then reshaped into a 1D sequence for further processing. A dense layer with 128 units was included for temporal analysis to project the spatial features into a formatted sequence by the Bidirectional LSTM layers.

Following the CNN layer, two Bidirectional LSTM layers were stacked, with the first layer containing 256 units and the second layer containing 128 units. Due to them being bidirectional, these layers processed the extracted feature sequence in both forward and backward directions, so the model could capture contextual dependencies across words within sentences. This capturing of contextual dependencies becomes particularly useful while dealing with cursive, overlapping, or connected handwriting patterns.

To avoid performing explicit character-level segmentation during recognition, a CTC loss layer was added (Shah et al., 2019). This layer aligned the input feature sequences with the target labels automatically, based on the temporal probabilities of the dataset, instead of exact spatial alignment between characters and image regions.

The model was compiled using the Adam (Adaptive Moment Estimation) optimiser during the training, because of its ability to dynamically adapt to learning rates, while also handling larger and complex datasets. Adam was selected as the optimiser because it combines momentum-based optimisation with adaptive learning rates. It incorporates concepts used in both Adaptive Gradient Algorithm (AdaGrad) and Root Mean Square Propagation (RMSProp), allowing the learning rate to be adjusted separately for different parameters during training. Before finalising the optimiser, the model was also experimented with Stochastic Gradient Descent (SGD), but the training process was slower and showed less stable convergence. The overall architecture of the proposed model is shown in Figure 4, including the convolutional layers, reshaping stage, Bi-LSTM layers, and the CTC component through which the input passes before generating the final text output.

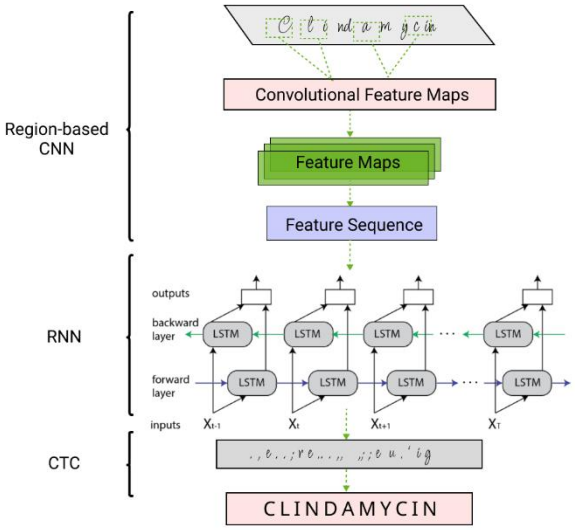


Figure 4. Model Architecture

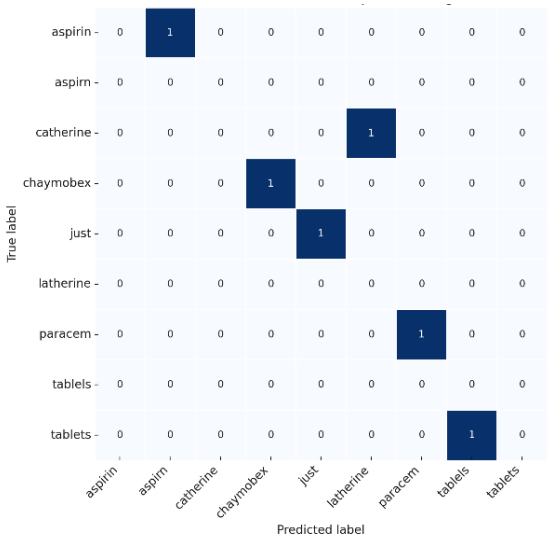


Figure 5. Confusion Matrix visualization

3.4 Training Configuration

The training configuration used during experimentation was based primarily on TensorFlow and Keras. A batch size of 32 was used and the model was trained for a maximum of 50 epochs. This value was selected after preliminary experimentation, while early stopping was also enabled to terminate the training whenever the validation loss stopped showing improvements, helping reduce overfitting and unnecessary computation loss. The Adam optimiser was chosen after comparison with SGD, as it showed more stable convergence while training. All the training and validation tasks were executed on a GPU enabled system powered by NVIDIA hardware, which reduced the overall training time and allowed multiple model configurations to be evaluated efficiently.

3.5 Evaluation Criteria

To evaluate the performance of the proposed R-CRNN model for recognising the handwritten prescriptions, classification metrics were used: Accuracy, Precision, Recall, and F1-score. Accuracy, to measure the proportion of correct predictions generated by the model and to get an overall indication of performance across the dataset. Precision was used to measure how many predicted outputs were actually correct, while Recall measured how many handwritten patterns were correctly identified. Because of connected or inconsistently written characters across the handwritten dataset, both Precision and Recall were important for the proper analysis of prediction quality. And then the F1-score combined the Precision and Recall metrics into a single classification metric, to provide a more balanced analysis of the model's performance.

Distance-based evaluation methods such as Levenshtein Distance (also known as edit distance) were also explored, that measured the number of character-level changes required to convert the predicted text into the actual ground truth text. This included string operations such as insertion, deletion, and substitution while converting one text sequence into another. Although this method was useful for analysing partial matches and minor recognition errors, it was comparatively less suitable for our sequence-to-sequence handwritten text recognition setup, wherein predicted and target sequences could vary considerably a lot in length and alignment. It also became difficult to interpret the overall classification behaviour directly using edit distance thresholds alone. And that is why, we focused primarily on accuracy, precision, recall, and F1-score metrics, computed at both character and word levels, to evaluate the transcription performance of the model more consistently and compare the results more reliably with existing handwriting recognition studies.

The Character Error Rate (CER) and Word Error Rate (WER) were also calculated to evaluate how accurately the model recognised individual characters and complete words, respectively, as shown in *Table* ***1***. CER and WER were used because they compare the model's predictions against a ground-truth reference, the mathematical calculation for both the error rate metrics is the same, just applied to different units i.e. characters v/s words:

$$Error\ Rate = \frac{S + D + I}{N}$$

These metrics, along with accuracy, precision, recall, and F1-score, helped to provide a clearer understanding of the model's performance and also highlighted the areas where further improvement was still needed.

To analyse the model's prediction and behaviour more closely, we also constructed a confusion matrix using a

sample of predicted and actual labels. The visual representation shown in Figure 5 helps to identify not only the correctly recognised terms but also the types of errors that are commonly made by the model during the transcription. For instance, predictions such as *aspirin* being recognised as *aspirn*, or *chaymobex* being interpreted as *chaymoney*, reflect the difficulties involved in distinguishing visually similar handwritten characters in cursive handwriting.

The confusion matrix also helped us in understanding where the recognition model was making repeated mistakes, and which character patterns required further improvements through fine-tuning or additional training data (Institute for Safe Medication Practices, 2011). To reduce such errors during recognition, the predictions were also associated with a small medical corpus containing commonly used medicine names, abbreviations, and dosage-related terminology collected during annotation. This helped the model produce more reliable predictions in cases where handwritten characters appeared similar, overlapping, or partially connected.

4. RESULTS & DISCUSSION

The experimental setup was designed using TensorFlow and Keras for the training and evaluation of performance of the proposed R-CRNN model for the recognition of handwritten prescriptions. TensorFlow and Keras were preferred over other frameworks for the flexible support they provide for deep learning architectures and sequence-based models. To handle the computational load associated with the simultaneous CNN and recurrent RNN integrated training, all the training and evaluations were performed on a GPU-enabled NVIDIA system.

Adam optimizer was incorporated for model training as it showed more stable convergence and better performance when compared to Stochastic Gradient Descent (SGD). To ensure a well-balanced model stability and training speed, the learning rate (α) was chosen to be 0.001 and the beta parameters were chosen as: first moment decay (β_1) = 0.9 and second moment decay (β_2) = 0.999 respectively, along with an epsilon value (ε) of 1e-07. This was done because the gradients of the dataset were highly sparse.

Model: "model_6"

Layer (type)	Output Shape	Param #	Connected to
input_data (InputLayer)	[(None, 256, 64, 1)]	0	
conv2d_18 (Conv2D)	(None, 256, 64, 256)	2560	input_data[0][0]
max_pooling2d_18 (MaxPooling2D)	(None, 128, 32, 256)	0	conv2d_18[0][0]
conv2d_19 (Conv2D)	(None, 128, 32, 128)	295040	max_pooling2d_18[0][0]
max_pooling2d_19 (MaxPooling2D)	(None, 64, 16, 128)	0	conv2d_19[0][0]
reshape_9 (Reshape)	(None, 64, 2048)	0	max_pooling2d_19[0][0]
dense_4 (Dense)	(None, 64, 128)	262272	reshape_9[0][0]
bidirectional_8 (Bidirectional)	(None, 64, 256)	263168	dense_4[0][0]
bidirectional_9 (Bidirectional)	(None, 64, 128)	164352	bidirectional_8[0][0]
input_label (InputLayer)	[(None, 10)]	0	
Dense_output (Dense)	(None, 64, 77)	9933	bidirectional_9[0][0]
input_length (InputLayer)	[(None, 1)]	0	
label_length (InputLayer)	[(None, 1)]	0	
outputs (CTCLayer)	(None, 1)	0	input_label[0][0] Dense_output[0][0] input_length[0][0] label_length[0][0]

Total params: 997,325
Trainable params: 997,325
Non-trainable params: 0

Figure 6. Model Architecture and Parameter Summary

The structure and parameter configuration were examined before evaluating the model's recognition performance. Figure 6 presents the detailed overview of the R-CRNN architecture, including the output shape of each layer, the number of trainable parameters, and the connectivity between different components. The complete architecture consisted of approximately 997325

trainable parameters, was trained using the Adam optimizer.

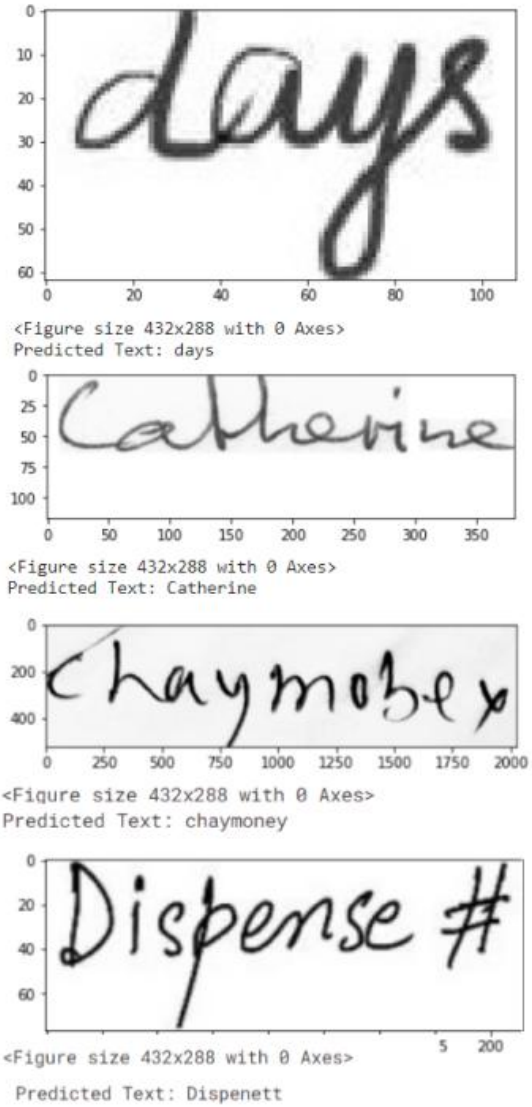


Figure 7. Sample output Images obtained from the model

The training dataset combined 11982 samples. Out of which, 9021 were handwritten line images from the IAM Handwriting Database and 2961 were the prescription lines that were manually collected and annotated using VIA. During data preprocessing, image augmentation increased the data variability using random rotations (±5°), horizontal shifts (±10%), zoom (90-110%), and elastic distortions. To get a consistent fit, all the images were resized to 256x64 pixels and normalised to grayscale frames (with a pixel range [0,1]). The dataset was split using 80:10:10 ratio, resulting in 9586 training samples, 1198 validation samples, and 1198 testing samples respectively, while also maintaining variation in handwriting style and character density across all the three different subsets.

The model achieved an accuracy of 73%, with a precision of 79%, a recall of 85%, and an F1-score of 82%, indicating that the model was capable of recognising different writing styles and readability levels of handwritten prescriptions. While training, the loss reduced steadily across progressing epochs, showing the model had gradual convergence and that the learning behaviour was stable. The model was able to process several handwriting challenges that are frequent in medical prescriptions, including connected characters in cursive handwriting, overlapping strokes and inconsistencies in writing patterns, and performed better as compared to standard CRNN and deep learning models, a tabular comparison is shown in

*Table **1*** and *Table 2*.

Table 2. Model Performance Metrics

Model	**Accuracy**	**Precision**	**Recall**	**F1-score**
Standard CRNN	64	67	68	62
R-CRNN+ CTC loss	73	79	85	82

The proposed R-CRNN model handles cursive and unsegmented handwriting much more effectively, where the traditional machine learning approaches and previously existing neural network-based systems used to struggle. The model captured both the visual and the contextual handwriting patterns more consistently, owing to the combination of convolutional layers for spatial feature extraction, and bidirectional LSTM layers for sequential learning. (Some of the outputs are shown in the Figure 7). The CTC loss layer further helped the model to process variable length handwritten sequences without requiring manually segmented characters or words during the training phase.

A model-based normalisation scheme was also integrated in the model architecture, as explored in recent studies such as Kizilirmak and Yanikoglu (2023), so that the model could handle the different handwriting styles better.

Overall, the experimental findings confirmed not only the technical viability of the R-CRNN model for decoding complicated and distorted handwritten texts but also its usability in streamlining healthcare documentation.

5. CONCLUSION

This research presents a practical approach for recognising doctors' handwriting using an R-CRNN-based deep learning model. The model was able to process a wide range of handwritten prescription styles and convert them into digitised text with reasonable recognition accuracy (up to 73%). This could help reduce errors caused by misreading of handwritten medical prescriptions and help avoid harmful side effects resulting from incorrect interpretations.

The proposed architecture combines the CNN layers, Bi-LSTM layers, and CTC loss to process distorted and unsegmented handwritten text without requiring any explicit character-level segmentation. The experimental results prove that the model could handle the challenges associated with handwritten prescriptions, including connected characters in cursive handwriting, overlapping strokes, as well as inconsistencies in handwriting styles. The work demonstrates how deep learning methods can support more reliable digitisation of handwritten medical records and improve healthcare documentation workflows.

There are still some limitations remaining to be addressed even though the proposed model shows promising results. While the current model focuses mainly on recognising alphabetical characters, whereas handwritten prescriptions in real medical settings also contain several special symbols, dosage markings, shorthand notations, and prescription-related characters such as "+", "-", "~", "#", "/", and "*". These symbols can carry important meaning and therefore also need to be considered during the recognition process. The performance of the model could further improve by training on larger datasets and including more domain-specific medical terminology and symbols during training. Future research should also explore more advanced architectures such as transformers and encoders, for handling multilingual prescriptions and complicated handwriting patterns.

Acknowledgement: The authors acknowledge the academic guidance provided by the faculty mentors and the support provided by local pharmacies during data collection and coordination. The authors also thank our family members and close contacts who voluntarily contributed with handwritten prescription samples, which were important for this research work.

REFERENCES

Alzaga, M. E. D., Rebua, S. M. P., Teston, J. A. A., & Viray, M. R. (2025). Diagnostic tool for avian influenza using image processing. *Proceedings on Engineering Sciences, 7*(1), 59–68. https://doi.org/10.24874/PES07.01.008

Bhushan, A., Patil, K., & Jolapure, S. (2001). *Handwritten script recognition*. *Journal of Computer Engineering*, 30–33. Retrieved from http://www.iosrjournals.org/iosr-jce/papers/sicete-volume2/18.pdf

Bluche, T. (2016). *Joint line segmentation and transcription for end-to-end handwritten paragraph recognition*. *Advances in Neural Information Processing Systems, 29*, 838–846. https://doi.org/10.48550/arXiv.1604.08352

Brits, H., Botha, A., Niksch, L., Terblanché, R., Venter, K., & Joubert, G. (2017). *Illegible handwriting and other prescription errors on prescriptions at National District Hospital, Bloemfontein*. *South African Family Practice, 59*(1), 52–55. https://doi.org/10.1080/20786190.2016.1254932

Cerio, A. P., Mallare, N. A. L., & Tolentino, R. M. S. (2015). *Assessment of the legibility of the handwriting in medical prescriptions of doctors from public and private hospitals in Quezon City, Philippines*. *Procedia Manufacturing, 3*, 90–97. https://doi.org/10.1016/j.promfg.2015.07.112

Dhiaf, M., Rouhou, A. C., Kessentini, Y., & Ben Salem, S. (2023). *MSdocTr-Lite: A Lite Transformer for full page multi-script handwriting recognition*. arXiv. https://doi.org/10.48550/arXiv.2303.13931

Fajardo, L. J., Sorillo, N. J., Garlit, J., Tomines, C. D., Abisado, M., Imperial, J. M., Rodríguez, R., & Fabito, B. S. (2019). *Doctor's cursive handwriting recognition system using deep learning*. *2019 IEEE 11th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM)*, 1–6. https://doi.org/10.1109/HNICEM48295.2019.9073521

Graves, A., Fernández, S., Gomez, F., & Schmidhuber, J. (2006). *Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks*. *Proceedings of the 23rd International Conference on Machine Learning (ICML)*, 369–376. https://doi.org/10.1145/1143844.1143891

Hassan, E., Tarek, H., Hazem, M., Bahnacy, S., Shaheen, L., & Elashmwai, W. H. (2021). *Medical prescription recognition using machine learning*. *2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC)*, 973–979. https://doi.org/10.1109/CCWC51732.2021.9376141

Hochreiter, S., & Schmidhuber, J. (1997). *Long short-term memory*. *Neural Computation, 9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Institute for Safe Medication Practices. (2011). *FDA and ISMP lists of lookalike drug names with recommended tall man letters*. Retrieved from http://www.ismp.org/tools/tallmanletters.pdf

Kizilirmak, R. C., & Yanikoglu, B. (2023). *English handwriting recognition using CNN-BiLSTM with test-time augmentation*. *Pattern Recognition Letters, 166*, 128–135. https://doi.org/10.1016/j.patrec.2023.01.007

LeCun, Y., Bengio, Y., & Hinton, G. (2015). *Deep learning*. *Nature, 521*(7553), 436–444. https://doi.org/10.1038/nature14539

Marti, U. V., & Bunke, H. (2002). *The IAM-database: An English sentence database for offline handwriting recognition*. *International Journal on Document Analysis and Recognition, 5*(1), 39–46. Retrieved from https://fki.tic.heia-fr.ch/databases/download-the-iam-handwriting-database

Nisa, H. (2023). *Handwritten synthetic dataset from the IAM* [Data set]. figshare. https://doi.org/10.25439/rmt.24309730.v1

Roy, V., Gupta, P., & Srivastava, S. (2006). *Medication errors: Causes & prevention*. *Health Administrator, 19*(1), 60–64. Retrieved from http://medind.nic.in/haa/t06/i1/haat07i1p60.pdf

Shah, J. K., Padte, A. S., & Ahirao, P. (2019). *SangCTC – Improving traditional CTC loss function for optical music recognition*. *2019 IEEE Pune Section International Conference (PuneCon)*. https://doi.org/10.1109/PuneCon46936.2019.9105720

Singh, D., Verma, G. K., & Yadav, S. (2024). *Deep learning techniques in handwriting recognition: A review*. *Journal of Intelligent Systems, 33*(1), 1–12. https://doi.org/10.1515/jisys-2023-0001

Visual Geometry Group, University of Oxford. (n.d.). *VGG Image Annotator (VIA)*. Retrieved from https://www.robots.ox.ac.uk/~vgg/software/via/

Yin, X.-W., Ma, Z., Chen, X., & Zhou, Q. (2021). *Handwritten string recognition based on YOLOv4 and CRNN on different deep learning frameworks*. *2021 5th Asian Conference on Artificial Intelligence Technology (ACAIT)*. https://doi.org/10.1109/ACAIT53529.2021.9731149

Zhao, Z.-X. (2022). *Research on Tibetan character recognition based on model CRNN*. *Second International Conference on Optics and Image Processing (ICOIP 2022)*. Retrieved from https://ui.adsabs.harvard.edu/abs/2022SPIE12328E..1HZ

Garima Saroj
Ajay Kumar Garg Engineering College, Ghaziabad,
India
garimak9@protonmail.com
ORCID 0009-0005-0030-4236

Ayushi Gautam
Ajay Kumar Garg Engineering College, Ghaziabad,
India
ayushigtm4@gmail.com
ORCID 0009-0003-1071-4445

Bhavya Agrawal
Ajay Kumar Garg Engineering College, Ghaziabad,
India
bhagrawal124@gmail.com
ORCID 0009-0002-1170-9501

Sarthak Gupta
Ajay Kumar Garg Engineering College, Ghaziabad,
India
sarthakgupta0212@gmail.com
ORCID 0009-0002-2538-2837