Bayesian-Driven Deep Learning and Machine Learning Approaches for Handwritten Word Recognition

Lingamaneni Sagar
Computer Science Engineering
Amrita Vishwa Vidyapeetham
Amaravati, India
sagarlingamaneni@gmail.com

Medisetty Dinesh
Computer Science Engineering
Amrita Vishwa Vidyapeetham
Amaravati, India
dineshmedisetty21@gmail.com

Pappula Baladhitya
Computer Science Engineering
Amrita Vishwa Vidyapeetham
Amaravati, India
baladhityapappula9999@gmail.com

M. Srinivas
Dept. of Computer Science
Amrita Vishwa Vidyapeetham
Amaravati, India
m_srinivas@av.amrita.edu

***Abstract*—Identifying handwritten words poses a complicated problem owing to the variances in handwriting styles and the possible noise and distortions present in the data. The performance of used deep learning and traditional machine learning handwriting recognition methods for recognizing handwritten words is studied in this paper and the advantages of using deep learning are discussed. In particular, neural network models such as CNN and 2DLSTM networks had a great work in this area. In contrast, we detect whole handwritten words without the need of segments into standalone characters. The CNNs are used for feature extraction, the Bidirectional LSTMs for sequence prediction and output is decoded via a CTC layer. The model was evaluated over the IAM words dataset from the IAM handwriting database, achieving 87% accuracy and a character error rate of 7.77%. In addition, we experimented with ML techniques, including SVM, RF has got an accuracy of 72% and 80% accuracies, and a Bayesian network of 61% accuracy, to provide a comparative analysis.**

Keywords—2DLSTM, Deep learning, Convolutional Neural Network, Handwriting Word Recognition, Support Vector Machine, Random Forest, Bayesian Network.

I. Introduction

Handwriting recognition has a key role in changing handwritten documents into digital format, which is especially valuable for tasks like archiving historical manuscripts and streamlining data entry. This technology enhances how accessible and usable written information can be. The system takes a 2D image of handwritten text as input and outputs the predicted word, enabling efficient interpretation of handwritten content.

This paper presents a model for offline handwriting recognition, highlighting its critical role in applications like verifying signatures on bank checks, recognizing postal codes on mail, and supporting criminal investigations, translations, and keyword searches [17]. In today's data-driven environment, achieving accurate handwriting recognition is in high demand, though it remains challenging [18]. Research has demonstrated that handwriting is highly individual, with each person displaying a unique style.

Handwritten characters can vary widely in style, size, and shape, even when written by the same person at different times. At the word level, the complexity increases, as factors like continuous handwriting, irregular spacing, and non-linear alignment make recognition more challenging. Conversion of images into text can be applied to scanned documents, photos of text, signs or billboards captured in images, and even subtitle text overlaid on scenes [19]. When dealing specifically with handwritten text, this process is known as Handwriting Recognition (HWR).

In this paper every phase is handled by model. So, in the decoding, the prediction given during the classifications handled by decoded that decodes the word in the image. The usage of Deep learning techniques got better compared to ML techniques. We also constructed Bayesian Network to represent the relationship between the nodes or features of the images. Also, Machine Learning techniques are also used to compared with the existing techniques.

II. Literature Survey

In their study [3], the authors introduced a technique that combines Support Vector Machines (SVM) with a nearest neighbor approach. This method represents word images as Fisher Vectors (FV), which capture gradient information from a GMM. A series of linear SVM classifiers is then trained to detect each attribute from a predefined set of word characteristics. To enhance comparison, Canonical Correlation Analysis (CCA) maps the predicted attribute vector and the binary attribute vector from the true word into a shared subspace. Using the IAM database, this approach achieved an accuracy of 79.99% with a CER of 11.27%.

In this study [4], the authors present a Convolutional Recurrent Neural Network (CRNN) model featuring a Connectionist Temporal Classification (CTC) layer at its output. The convolutional part of the network, used for feature extraction, consists of seven layers with max-pooling applied after the 1st, 2nd, 4th, and 6th layers for down sampling. To mitigate internal covariate shift, batch normalization is used on the third and fifth layers. The Recurrent Neural Network (RNN) component includes two LSTM layers, which form a BLSTM. The output from the RNN is then fed into the CTC layer, where decoding is performed using the beam search algorithm.

In this method, the authors [5] introduce dropout layers within RNN. The input separated is divided into 2 × 2 blocks and processed by four LSTM layers, each scanning in a distinct direction. The output from each LSTM layer is then passed to separate convolutional layers with six feature maps, using filters sized at 2 × 4. These convolutional layers are applied without overlap or biases, acting as a learnable subsampling step rather than a fixed subsampling approach.

The authors [15] introduce a system with three core components: convolutional feature extraction, recurrent layers, and a transcription layer. For feature extraction, they use CNN taking an input image of Wx66x1 dimensions, where W is the variable width. The network is organized into six units, with each unit incorporating two residual modules that include convolutional layers, residual connections, and layer normalization. Max-pooling layers are applied between these units to downscale spatial dimensions to reduce spatial dimensions. To capture time dependency, three bidirectional LSTM layers with 512 units per direction are applied to the feature vector sequence. The final LSTM output is fed into a transcription layer with a CTC layer for decoding. This system achieved 79.51% accuracy on the IAM dataset and 88.05% on the RIMES dataset.

In [16], the authors introduce a model called Scrabble GAN, which is a semi-supervised approach for generating versatile handwritten images, both in style and lexicon. Scrabble GAN can produce word images of any length. It can generate arbitrarily long word images, and it is trained in a semi-supervised manner. The authors evaluated the model using the RIMES, IAM, and CVL datasets, achieving a Word Error Rate (WER) of 25.10% on IAM and 12.29% on RIMES.

In their work [1], THE authors propose an architecture that combines CNN with an encoder-decoder framework. The CNN extracts FEATURE from image patches, which are then fed into a sequence-to-sequence network using LSTMs. The encoder LSTM processes the feature sequence, while the decoder LSTM performs character recognition and incorporates an attention mechanism. Max-pooling and a dropout layer come after the two convolutional layers that make up the CNN portion. The encoder captures the relationships between the features, and the decoder generates the transcription. Trained on the IAM database, the model achieved 87% accuracy, aided by the use of a dictionary for transcription.

This paper [7] focuses on applying deep learning techniques to handwritten text recognition (HTR) systems. The process begins with gathering data for training on handwritten texts, followed by extracting features from these datasets. The model is then trained using deep learning methods. Instead of recognizing individual characters, the model is designed to recognize words, which enhances accuracy. The LSTM-based deep learning model developed in this work shows promising results. Ultimately, the HTR system is integrated into an OCR framework, and a comparison of results is presented. The study compares two approaches using the IAM handwritten dataset, with the 2DLSTM-based approach demonstrating superior performance.

The authors [9] of this study have examined different methods to obtain consistent and dependable outcomes. The introduction of neural networks, CNNs and RNNs, has led to substantial advancements in this field. In this study, we built a model capable of identifying complete handwritten words without requiring disassembly into separate characters. The model integrates a CNN for feature extraction, an RNN for text prediction, and a closing CTC (Connectionist Temporal Classification) layer for decoding the outputs. Following comprehensive tests on the IAM handwriting dataset, the model reached an accuracy of 77.22% and a CER of 10.4%.

III. PROPOSED METHODOLOGY

A. *Dataset Description*

The most widely used database for handwritten word and text recognition tasks is the IAM handwriting database [21]. 657 authors have contributed 115,320 words to the IAM Words Dataset. The dataset's examples are grayscale pictures in jpg format on a white background. Figure 1 illustrates this:

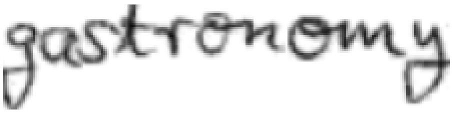

Fig 1: Image from the IAM dataset

B. *Preprocessing*

The preprocessing process starts by loading and decoding the images from their file paths. Each image is resized to the desired dimensions, while preserving its original aspect ratio. Any necessary padding is added to ensure the image fits within the target size. The pixel values of the image are then normalized to a range of [0, 1], preparing the image for input into the model. The labels are processed by converting each character into an integer using a String Lookup layer. If a label's length is shorter than a predefined maximum, it is padded with a special token to ensure consistency across all labels.

These steps are applied to each image-label pair, with the data being grouped into batches for efficient training. To further optimize performance, the dataset is cached in memory and preloaded for faster access, minimizing delays during training. The preprocessing pipeline also leverages parallel processing to speed up the transformation of the data, ensuring that the images and labels are ready for the model in the most efficient manner possible.

C. *Proposed Architecture*

This network is designed for sequence-to-sequence tasks like handwriting recognition, where the objective is to map variable-length image sequences to corresponding text sequences. The input layer receives grayscale images with dimensions of 128x32 pixels. Initially, a convolutional layer with 32 filters of size 3x3 extracts basic features from the images, resulting in feature maps of shape (None, 128, 32, 32). A max-pooling layer, using a 2x2 kernel, reduces the spatial dimensions to (None, 64, 16, 32), preserving key features while lowering computational demands.

The second stage involves another convolutional layer with 64 filters of size 3x3, which refines the features and produces output with dimensions (None, 64, 16, 64). A subsequent max-pooling operation further reduces the spatial dimensions to (None, 32, 8, 64). At this point, the feature maps are reshaped into a flattened vector of shape (None, 32, 512), preparing the data for the dense layers.

A dense layer with 64 units is then applied, followed by a dropout layer to reduce the risk of overfitting by randomly disabling a portion of the network's units while it is being trained.

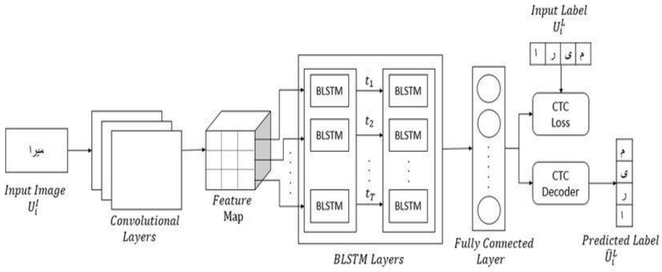

Fig 2: Model Architecture

The model then processes the sequence of features using two bidirectional LSTM layers [20]. The first LSTM layer, with256 units, captures both forward and backward dependencies in the sequence, producing output of shape (None, 32, 256). The second LSTM layer, with 128 units, further refines this sequence representation, outputting a tensor of shape (None, 32, 128). These LSTM layers are essential for learning temporal dependencies in the sequence.

The network includes a final dense layer with 81 units, corresponding to the possible character set (such as letters, digits, and special characters), and generates predictions for each time step in the sequence. The architecture concludes with a Connectionist Temporal Classification (CTC) loss layer, which compares the predicted sequence with the actual labels. CTC loss is especially beneficial for sequence-to-sequence tasks, as it allows the model to handle sequences of different lengths without requiring precise alignment between inputs and outputs.

To enhance training efficiency, batch normalization and ReLU activations are applied after each convolutional layer, ensuring faster convergence and mitigating issues like vanishing gradients. The architecture efficiently combines convolutional layers for feature extraction, bidirectional LSTMs for sequence processing, and CTC loss for training, making it well-suited for tasks like handwriting recognition, where both the input and output sequences can vary in length.

D. *Machine Learning Models*

Various Machine Learning algorithms [3] such as Support Vector Machine and Random Forest are utilized. These techniques represent the input word image using Fisher Vectors, which aggregate the gradients of GMM. It subsequently trains a collection of linear SVM classifiers.

E. *Bayesian Models*

Bayesian models are effective in handwriting text recognition by managing uncertainty in handwriting styles. **pgmpy**, a Python library for probabilistic graphical models, helps build Bayesian networks that model relationships between variables. These networks use **Conditional Probability Tables (CPTs)** to define the likelihood of outcomes based on observed data. In handwriting recognition, CPTs represent the probability of recognizing characters from image features. Hidden Markov Models (HMMs) are often used to model character sequences, while **Bayesian Networks** improve accuracy by accounting for uncertainty in model parameters, making them useful for complex recognition tasks.

F. *Training*

Given the input and label, the model is being trained to minimize loss. 50 epochs are used to train the model, and the training loss is decreasing. When the character error rate does not improve for three consecutive epochs, the training procedure is terminated.

Algorithm Used:

Load and Preprocess Data:

Load the IAM dataset, and split into train, validation, and test sets.

Define CNN-BILSTM Model:

This network processes 128x32 grayscale images using convolutional layers for feature extraction, followed by bidirectional LSTMs to capture sequence patterns.

With dropout to prevent overfitting and CTC loss to align input and output sequences of varying lengths, it’s well-suited for handwriting recognition tasks.

Compile and Train Model:

Compile utilizing sparse categorical cross-entropy loss and an Adam optimizer valued at a learning rate of 0.0001, then train for 50 epochs on training data with validation on a validation set.

Evaluate Model:

Evaluate the model on the test set and output the test loss and accuracy metrics.

IV. RESULTS AND DISCUSSIONS

We tested our model on the IAM words dataset. We tested our model on IAM and it achieved a CER of 7.4% and achieved an accuracy of 87% Table 1 highlights the results as you can see in the Fig 3

In this model the accuracy is calculated by Eq 2

Eq 2:

$$\text{Word Accuracy} = \frac{No\ of\ correct\ word\ predictions}{Total\ no\ of\ word\ \ predictions}$$

Eq 3:

$$\text{CER} = \frac{Number\ of\ Incorrect\ Characters}{Total\ number\ of\ characters\ in\ the\ text}$$

The outcomes of additional techniques for identifying handwritten words in the well-known IAM database are displayed in Table 3.

DATASET	PROPOSED MODEL	CER (%)	ACCURACY (%)
IAM Handwriting Dataset	CNN+2DLSTM+ CTC LAYER	7.8%	87%

Table 1: Accuracy and CER

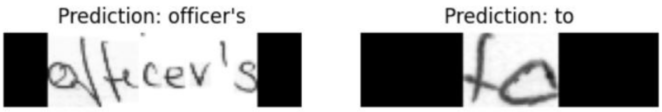

Fig 3: Example of Output

Model	Accuracy	Precision	Recall
SVM	72	70	74
RF	80	79	77
DT	71	72	70
XG BOOST	72	73	75
Bayesian Network	61	61	60

Table 2: Results of Other Models:

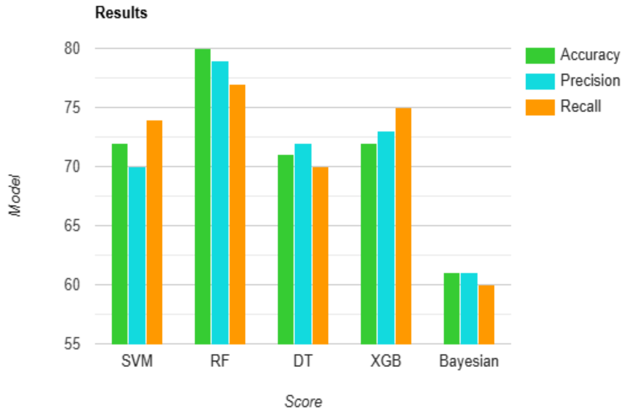

Fig 4: Evaluation Results of Models Used

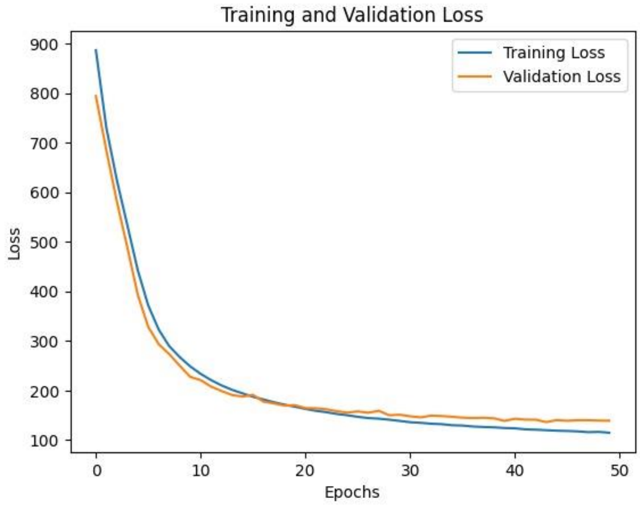

Fig 5: Training and Validation Loss Curves

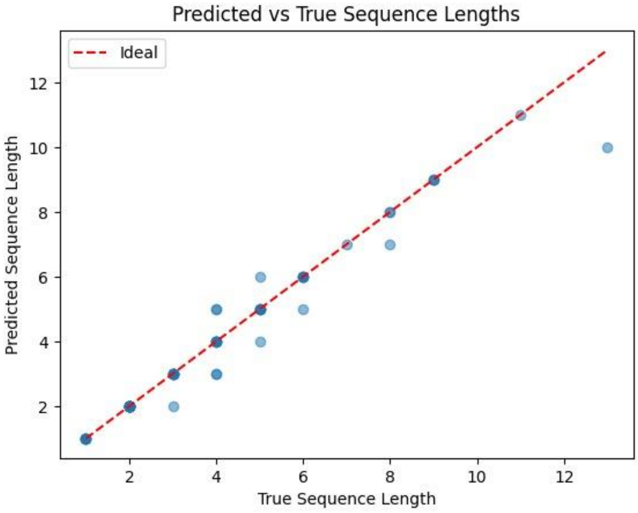

Fig 6: Predicted vs True Sequence Lengths

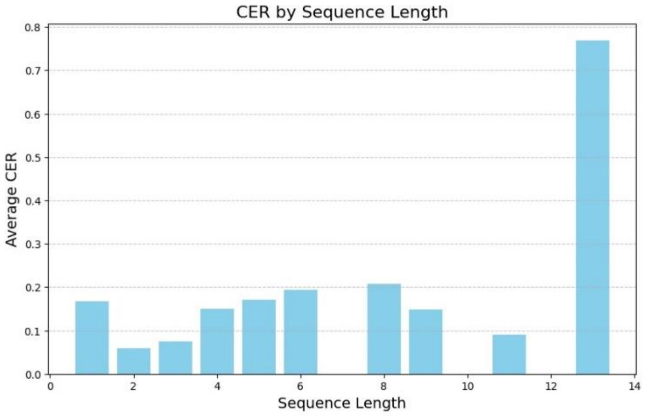

Fig 7: Length Based Character Error Analysis

MODEL USED	**CER (%)**	**ACCURACY (%)**
CNN-LSTM [2]	11.27%	79.99%
CNN-RNN [3]	13.92%	68.52%
CNN-LSTM [16]	8.8%	76.2%
CNN-BILSTM [20]	12.6%	79.6%

Table 3: Evaluation Results: Insights from Existing Studies

V. CONCLUSIONS

Several methods are processed and evaluated and recently using the neural networks in particular scam of CNN and RNN, shown impressive results in the handwriting recognition. Here we present a model that can be used for incremental on-line handwriting recognition of handwritten words, but does not perform pre-segmentation into characters. Our model includes a CNN for the feature extractor, RNN to perform the prediction process and at last, use a CTC layer to help us in decode the prediction stage. In the well-known IAM handwriting database, we did a lot of experiments and achieved 87 % accuracy and 7.4 % CER (character error rate) and also Machine Learning models like SVM and RF obtained 72% and 80% accuracy respectively and finally we also implemented Bayesian Model which obtained an accuracy of 61%.

VI. References

[1] Sueiras, J., Ruiz, V., Sanchez, A., Velez, J.F.: Offline continuous handwriting recognition using sequence to sequence neural networks. Neurocomputing 289, 119–128 (2018). https:// doi.org/10.1016/j.neucm.2018.02.008

[2] Almazan, J., Gordo, A., Fornes, A., Valveny, E.:Word spotting and recognition with embedded attributes. IEEE Trans. Pattern Anal. Mach. Intell. 12, 2552–2566 (2014)

[3] Kang, L., Toledo, J.I., Riba, P., Villegas, M., Fornés, A., Rusiñol, M.: Convolve, Attend and Spell: An Attention-based Sequence-to-Sequence Model for Handwritten Word Recognition. In: Brox, T., Bruhn, A., Fritz, M. (eds.) GCPR 2018. LNCS, vol. 11269, pp. 459–472. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-12939-2 32

[4] Tran, H.-P., Smith, A., Dimla, E.: Offline Handwritten Text Recognition using Convolutional Recurrent Neural Network In: 2019 International Conference on Advanced Computing and Applications (ACOMP) (2019).

[5] Pham, V., Bluche, T., Kermorvant, C., Louradour, J.: Dropout Improves Recurrent Neural Networks for Handwriting Recognition In: 2014 14th International Conference on Frontiers in Handwriting Recognition. (2014) https://doi.org/10.1109/icfhr.2014.55

[6] Priya, A., Mishra, S., Raj, S., Mandal, S., Datta, S.: Online and offline character recognition: A survey In: Communication and Signal Processing (ICCSP), 2016 International Conference on, pp. 0967–0970, (2016)

[7] I. Maglogiannis et al. (Eds.): AIAI 2023 Workshops, IFIP AICT 677, pp. 347–358, 2023.https://doi.org/10.1007/978-3-031-34171-7_28

[8] Singla, P., Munjal, S.: A Review On Handwritten Character Recognition Techniques IJIRT 2(11) ISSN: 2349–6002 (2016)

[9] Handwritten Text Recognition using Deep Learning DOI: 10.1109/RTEICT49044.2020.9315679

[10] Shahbaz Hassan, Ayesha Irfan, Ali Mirza, Imran Siddiqi, ”Cursive Handwritten Text Recognition using Bi-Directional LSTMs: A case study on Urdu Handwriting”, in Deep-ML, 2019

[11] P. Voigtlaender, P. Doetsch, and H. Ney, “Handwriting recognition with large multidimensional long short-term memory recurrent neural networks,” in ICFHR, 2016.

[12] Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Computation 9(8), 1735– 1780 (1997)

[13] Bluche,T., Ney, H., Kermorvant, C.: Feature Extraction with Convolutional Neural Networks for Handwritten Word Recognition In: 12th International Conference on Document Analysis and Recognition (ICDAR). IEEE, pp. 285–289 (2013)

[14] Fogel, S., Averbuch-Elor, H., Cohen, S., Mazor, S., Litman, R.: ScrabbleGAN:SemiSupervised Varying Length Handwritten Text Generation In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)

[15] Mor, N., Wolf, L.: Confidence prediction for lexicon-free OCR In: Proceedings of the IEEE Winter Conference on Applications of Computer Vision, pp. 218–225 (2018)

[16] Fogel, S., Averbuch-Elor, H., Cohen, S., Mazor, S., Litman, R.: ScrabbleGAN:SemiSupervised Varying Length Handwritten Text Generation In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)

[17] Bluche,T., Ney, H., Kermorvant, C.: Feature Extraction with Convolutional Neural Networks for Handwritten Word Recognition In: 12th International Conference on Document Analysis and Recognition (ICDAR). IEEE, pp. 285–289 (2013)

[18] Bhuia, A.K., Das, A., Bhunia, A.K., Kishore, P.S.R., Roy, P.P.: Handwriting Recognition in Low-resource Scripts using Adversarial Learning In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)

[19] Memon, J., Sami, M., Khan, R.A., Uddin, M.: Handwritten optical character recognition (OCR): A comprehensive systematic literature review (SLR). IEEE Access 8, 142642–142668 (2020)

[20] **CNN-BiLSTM model for English Handwriting Recognition: Comprehensive Evaluation on the IAM Dataset**

[21] IAM Handwriting Database