Interpreting Doctor’s Handwritten Prescription Using Deep Learning Techniques

Rizwanullah Mohammad, Ajay Kumar Varma Nagaraju and Suneetha Manne
https://orcid.org/0000-0002-8917-276X

IT Department, Velagapudi Ramakrishna Siddhartha Engineering College, Vijayawada, Andhra Pradesh, India

mailtorizwanullah@gmail.com
ajaynagaraju32@gmail.com
hodit@vrsiddhartha.ac.in

* Corresponding Author : Rizwanullah Mohammad

Abstract: A Doctor’s Handwriting Recognition model can predict (recognize) the text present in the doctor’s prescription, by feeding image of that medicine name to the model and it predicts the text present in the image and it gives the final medicine name as digital text. This model is suitable only for Text written in English Language and not suitable for other languages of texts written in prescription. The model based on training dataset the output it produces may get varied and based on training images count. Both convolution layers and Bi-Directional LSTM layers can be used for feature extraction and recognizing text respectively.

Keywords: Bi-Directional LSTM units, Bi-Directional Gated Recurrent Units, Convolution Layers, Adam optimizer, Relu Activation Function.

1 Introduction

It is most common that people can’t understand and interpret the doctor’s handwriting. The calligraphy they follow which is always challenging for ordinary people and even for pharmacist to understand doctor’s handwriting [1-3]. Until they understand correctly then cant give correct medicine to Patient. Due to usage of wrong medicines they may face severe consequencies with respect to their health. This problem need to be solved with the latest technologies those are present[5]. The solution for this is deep learning models[11]. A deep learning model can take large data inputand can process with help of neural network and layers[13]. They can give high accuracy and more reliable. Now by making use of deep learning techniques,involving all the terms in deep learning to provide a optimal solution for this. The Bi-Directional LSTM model can provide a solution which can predict text present doctor’s prescription’s image which are passed as input to our model.The comparision was made between Bi-Directional LSTM and Bi-Directional GRU[17]. For this application of interpreting doctor's handwritten prescriptions, either one of the architectures between Bi-LSTM and Bi-GRU will be chosen and implemented[23].IAM is a dataset which contains handwritings of many people.The two algorithms, Bi-Directional LSTM and Bi-Directional GRU, are trained using images from the IAM Dataset, and the accuracy results are compared and examined. This paper provides a brief explanation of how the performance of model is affected by the purity and amount size of the dataset.

2 Literature Study

Here are the available models, That Researchers proposed and developed. They Proposed different methodologies to recognize text in an image and showcasing their performance.

T. Jain et al. [1] presented a model employing the BI-LSTM Model for the recognition of a doctor's handwriting. They have only created a model; no mobile or web applications have been created to execute the model in real time. In order to minimise overfitting and increase the model's resistance to noise, they employed data augmentation approaches.Changing network topologies increases accuracy and reduces complexity.

S. Tabassum et al. [2] presented a model for the identification of doctors' handwriting. After employing the SRP Augmentation approach, they achieved an accuracy of 89%. Some of the participating physicians' prescription pictures were made available. They made a dataset called handwritten corpus. The introduction of SRP, increased the size of the data sets. For predicting the handwriting of doctors, an online character recognition system utilising Bi-LSTM was employed. Variable handwriting styles should be added in the dataset to improve the model's knowledge and prediction abilities.

L. J. Fajardo et al. [3] presented a model for interpreting the doctor's handwriting was developed. They employed CRNN Model. The accuracy of the tests conducted using the mobile application was 72%. The model is implemented through the use of a mobile application called DCHRS and aims to recognise the name of the medication inside prescription that has been captured, as well as to provide the digital text of the handwriting. Low image count and precision are insufficient for medical applications.

España-Boquera et al. [4] presented a public dataset named SPA sentences. And provided indicators at level of sentences,proposed methodology of combining convolutional blocks with LSTM and CTC blocks. Unidirectional LSTM is less accurate than Bidirectional LSTM.

Maalej et al. [5] proposed a methodology which replaces 1-D LSTM with Multi LSTM Blocks. They connected MDLSTM blocks with CTC-Maxout block and they performed Data augmentation which increases accuracy. MDLIST increases model complexity and the interpretability will get reduced.

A. Harikrishnan etal[6] presented their model for hand-written numbers recognition with CNN and MLP Architectures.The model was trained with CNN and MLP Architectures of models.The results are,CNN achieved 99% Accuracy and MLP achieved 97% Accuracy.CNN performance is more than MLP.According to the proposed model if the number of CNN layers is increased then it may give better Accuracy.

A. Nikitha et al. [7] proposed a handwritten text recognition system which uses word error rate instead of character error rate. The Two dimensional LSTM Architecture is used for model. The model wasn't evaluated on character error rate, a crucial criterion for making it more robust and efficient.

S. Hassan et al. [9] presented a methodology for recognition of handwritten text for any script.Their proposed model achieved an average accuracy of 83%.The proposed architecture's network topology must be adjusted to raise the number of filters to 1024 for better results.

Ul Sehr Zia etal[10] presented a handwritten text recognition model based on CRNN.This model was trained with NUST Urdu handwriting dataset. And demostrated how model can be trained with english and urdu bilingual handwriting.

Alrobah, N et al. [11] proposed applying of robust methodologies to handle challanges in recognition of different language handwriting data. Mainly the arabic handwritten data. The methods to use and how to make feature extractions in most effective way which can help model to be more Robust and Perform well.

U. Shaw et al. [12] proposed a model which can recognize poor legible handwriting and provide a digital text. OCR was used in order to recognize handwriting. The model was trained with MNIST dataset.Generally,for handwriting recognition the model should be trained with dataset containing alphabets rather than numbers.

S. Sharma et al. [13] demostrated the usage of various scripts like devanagari,Gurmukhi and other language datasets to train the model.A model was developed with CNN architecture.RNNs are recommended for successful model feature decoding.

Sethy et al. [14] demostrated the CNN architecture usage for efficient feature extraction and pattern recognition. The system was training with Odia and Bangla Handwritten data. This data was trained with CNN Architecture.RNN architecture must be employed to recognize text.

S. Haboubi et al. [17] presented a model for urdu handwriting recognition.They demonstrated how the Bi-GRUs are giving better acccuracy with less complexity and consuming less memory. Bi-GRU is LSTM’s rival.

A. Abdallah etal[23] presented hand-writing recognition model for Russian scripts.They implemented CNN for feature extraction and Multi-dimensional GRU for feature decoding. They confirmed Multi-dimensional GRU's good performance

The above methods and presentations gives us an in-depth grasp of their models, performance, and design approaches. By understanding their approaches, better ones can be devised to overcome obstacles and design a better model.

3 Proposed Work

3.1 Dataset

The suggested model used 86800 grayscale pictures. IAM Dataset contains different handwriting scripts[16].The dataset is splitted into 90:5:5 ratio(Train_Validate_Test).

3.2 Pre-processing

In this pre-processing phase the images will get reshaped to width of 128 and height of 32 and padding to 99. And then the datatype will be changed to float32 which is called as casting. Thıs may increase the model performance.This is distortion free image processing. Initial, using tf.image is the first stage in the resizing process. The image path should be sent along with width, and height to the resize() function while maintaining the aspect ratio. Following this, the additional padding is added to resized image. Padding can be added to image by subtracting width and height values(128,32 respectively) with image shape which image want to add padding to, with help of tf.shape() function. With tf.transpose() function by giving perm = [1,0,2] This is nothing but setting up the required tensor dimensions. Perform flip_left_right of image inorder to get image flipped along with width dimensions.

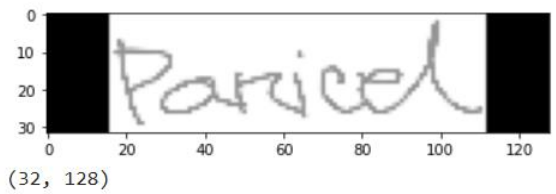

Fig. 1. Distortion free image pre-processing

3.3 Design Methodology

A brief description of proposed methodology:

The cropped images from the IAM Dataset are being utilized and a few medical datasets in conjunction with our model to train. The training dataset, validation dataset, and testing datasets have been divided in a ratio of 90:5:5.The network has a sophisticated design that includes seven convolutional layers, optional batch Normalization layers, Max Pooling layers, Relu activation functions, a Bi-directional LSTM layer, and a CTC layer[1].The procedure increases the number of channels in the first convolution layer from 1 to 64. which, after several layers, is raised to 128.

The below diagram describes architecture of our work.

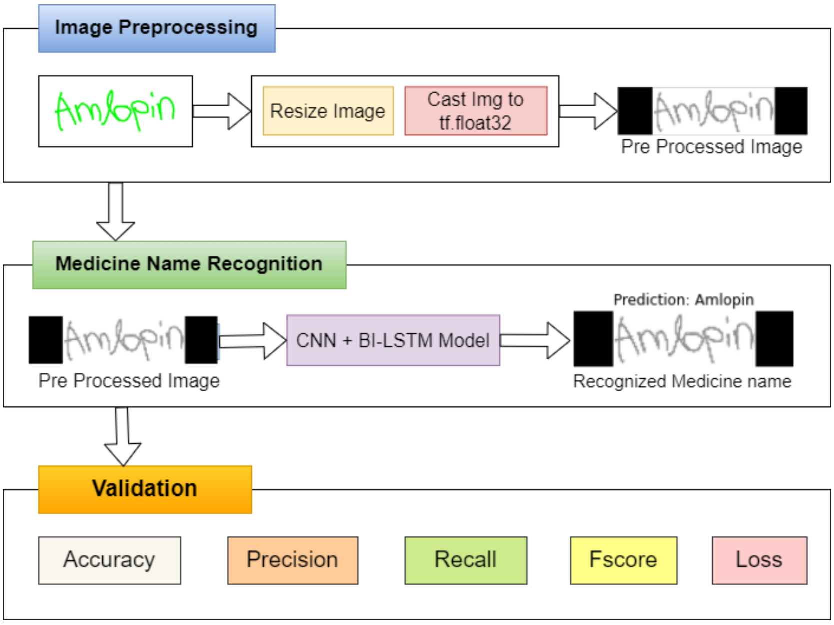

Fig.2. Architecture design for our work

The Bi LSTM layer aids in decoding of the convolution layer-generated feature mapping. Next create a layer for label input for the appropriate images, followed by a thick layer. To find the CTC loss, the last layer would be the CTC layer. The next step is to determine the callback distance. During this stage, ctc decoding will be done using predictions.Then,using tf.sparse.from_dense and dtype as an int64, sparse the predictions made from dense. Identify a point when an increase in epoch values leads to a rise in loss value and the model will no longer improve performance at which point training may be stopped. Next, build the model and train it with various rising epoch values, setting checkpoints as necessary.The count variable can be used to determine the model's accuracy by passing some data as input, checking the total count of right predictions, and calculating the percentage of right predictions.The accuracy will be proportional to the initial weights which are being established in the input layer. Setting appropriate starting weight values for the input layer and subsequent layers, as well as choosing an appropriate activation function, are required. Relu inside a convolution layer and Dense layer are typically utilised as activation functions for models like CNN and LSTMs.

StringLookup Layer:

This is proposed preprocessing layer which is being used. It converts characters in each word while training into integers. It converts each character in a vocabulary to an integer. StringLookup Layer converts integers to characters during prediction.During the conversion of num back to character the invert should be set as True.If the training label is ‘crocin’ then the vocabulary will be {c,r,o,c,i,n}.

CNN + Bi LSTM:

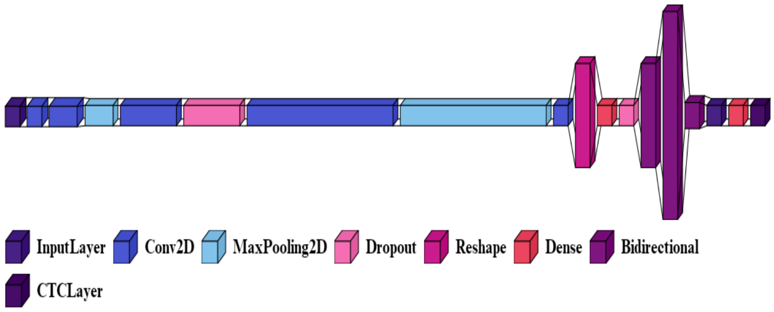

Fig.3. Model topology for Bi LSTM

The fig(3) shows model design for Bi LSTM Model. The Bidirectional layer in fig(3) represents the Bi LSTM Layer. The initial layer is input layer of height 32, width 128 and the channel is 1 because it’s a grayscale image. The proposed model with Bi-LSTM contains five convolutional layers with filters 32,128,256,1024 and 64.For three Bi LSTM layers number of hidden cells are 512,1024,64. The CTC layer computes the character wise error rate instead of word wise error rate and returns loss value for every step per epoch.

CTC LAYER:

The proposed CTC layer merges repeated related characters. The first phase in this layer is predicting the tokens in a sequential order. Second phase is to merge repetitive characters and drop noisy tokens. CTC Layer provides final output after phases 1 and 2. This Layer offers a Loss function to compare anticipated and actual values.

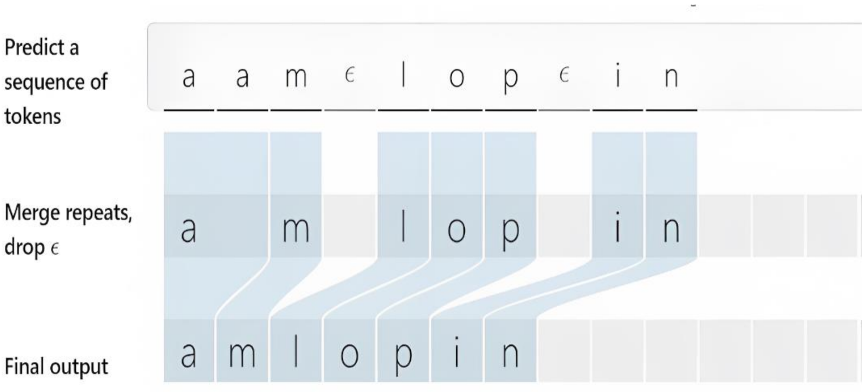

Fig. 4. Functionality of CTC Layer

Fig(4) shows CTC Layer's operation. It doesn’t use any traditional aligning methods, instead it will eliminate the process of Alignment. Bi-LSTM or Bi-GRU may output 'Amlopin' as a series of characters. The custom CTC Layer filters characters in each word. In the proposed network design, CTC Layer input and output shapes are (None,32,81) itself. Every training step calculates CTC loss. CTC batch cost's arguments are y pred, y true, label length, and input length. It returns each element's loss. Image input determines this argument's value.

CNN + Bi-Directional-GRU:

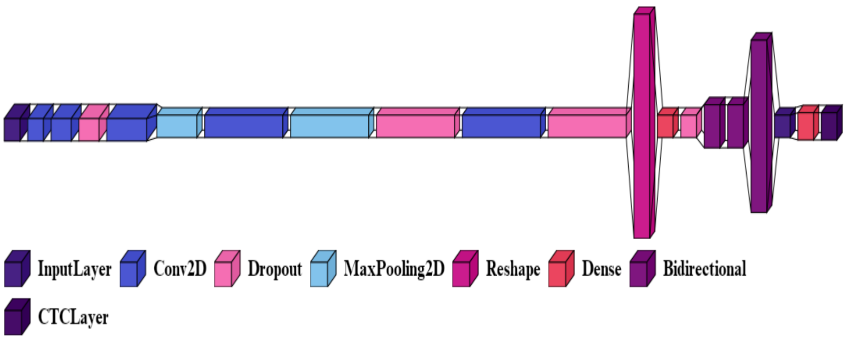

Fig. 5. Model Architecture for Bi-Directional GRU

Fig(5) demonstrates Bi-Directional GRU model building. Input layer shape is (128,32,1) with width, height, and channel. Dropout layers between convolutional and pooling layers prevent overfitting. The Dropout value used is 0.25. 5 convolution layers with filter sizes 32,128,256,512,1024. The next 3 Bi-Directional

GRU layers have 128,128,512 hidden cells. The CTC LAYER works similarly to the prior Bi-LSTM model fig (4).

After comparing the both models, Bi-LSTM has more training parameters than Bi-GRU. The model's hidden units determine the total number of trainable parameters. The total trainable parameters are 6,907,025. Consider character error rate for an effective model.

3.4 Selection of Algorithm

The implemention of proposed model is done with two different algorithms, they are Bi- LSTM and Bi-GRU. The CNN is common in both models. But the type of RNN used is different. The model uses algorithms of CNN + Bi-Directional LSTM[1] in first model and CNN + Bi-Directional GRU[17] in second model. Here, the comparison is done in order to determine which model is performing well. From fig(2) in medicine name recognition phase the algorithms used is different,but other steps remain same.Both LSTM and GRU are Recurrent neural networks but the difference is the size of data they can handle and gates present. Using IAM dataset, the training dataset size is 86,800 images. Bi-Directional LSTM is excellent at handling huge data, followed by GRU. GRU is preferred for small datasets. Input,Output,Forget are LSTM gates. GRU only has update and reset gates. GRU is simpler than LSTM. So, its better to use GRU for small sized data and LSTM for large sized data. The selection of this two algorithms is done in order to test which algorithm will perform well.

The Mathematical Notation of Bi-Directional LSTM as follows:

$$\hat{p}^{\langle t\rangle} = g(W_x[\vec{a}^{<t>}, \overleftarrow{a}^{<t>}] + b_x) \quad (1)$$

From eq.(1) , $\hat{p}$ is used for representation of output, $\hat{p}^{\langle t\rangle}$ will represent output at tth unit of time. The g represents the hidden layer function. W_x denotes the hidden layer weights matrix. And similarly b_x denotes hidden layer vector for bias values. $\vec{a}^{<t>}$ gives the forward hidden sequence and $\overleftarrow{a}^{<t>}$ gives the backward sequence.As the employment of Bi-directional layers is being done,the two directional sequences can be obtained. Here, the iterations are done on forward and backward sequences and then the output will get updated when concatenation is done for each word in forward and backward sequence.

The Mathematical Notation of Bi-Directional GRU as follows:

$$h_t = G_{forward}\left(x_t, \overrightarrow{h_{t-1}}\right) \oplus G_{backward}\left(x_t, \overleftarrow{h_{t+1}}\right) \quad (2)$$

Eq.(2) Similar to Bi-Directional LSTM,the Bi-Directional GRU also allows data sequence in two directions forward and backward.From eq.(2), $G_{forward}$ is a GRU function that denotes the data sequence flow in Forward direction and $G_{backward}$ is a GRU function that denotes the data sequence flow in Backward direction. $\oplus$ is vector concatenation operator for $G_{forward}$ and $G_{backward}$ Data sequence flows. $G_{forward}(x_t, \overrightarrow{h_{t-1}})$ denotes forward GRU's state and $G_{backward}(x_t, \overleftarrow{h_{t+1}})$ denotes backward GRU's state. x_t denotes input vector. h_t denotes output of cell at time t. By performing concatenation between forward and backward GRU states gives the output h_t.From the eq.(2), the concatenation of both forward and backward sequences makes the model to access previous states as well.

4 Result and Observations

4.1 Test case Results

Testcase results are below. Model predicts based on custom image inputs.
Predictions:

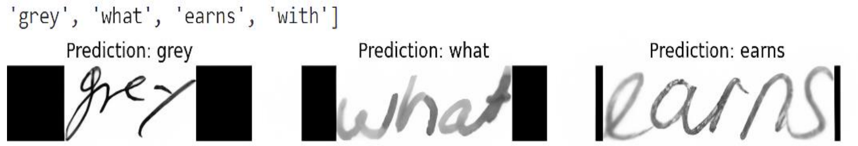

Fig. 6. Output Test Case predicted by the model

Fig(6) displays IAM Testing test case results. Fig(6) shows the model's predictions after 25 Epochs of training. All three testing dataset images were successfully predicted by the model. First, the image was preprocessed and passed to the model for prediction.

4.2 Observations and Analysis

Accuracy observations for Bi-Directional LSTM and Bi-Directional GRU at different epochs are shown below.

Table. 1. Comparison of Bi-LSTM and Bi-GRU Accuracies

Epochs	**Accuracy (%)**	
	BI-LSTM	BI-GRU
20	75	69
25	78	74
30	81	77

From table(1), Two models (one with Bi-LSTM and another with Bi-GRU) are trained and tested to compare accuracy. At 30th Epoch, Bi-Directional LSTM had 81% Accuracy and Bi-Directional GRU 77%. Bi-Directional LSTM Algorithm training took 5 hours and Bi-Directional GRU training took 3 hours. Bi-Directional GRU, a less sophisticated model, trained 30 epochs in less time than Bi-Directional LSTM, but had lower accuracy. Bi-Directional LSTMs outperform Bi-Directional GRUs on larger datasets.

Bi-Directional GRU has 88% training data while Bi-Directional LSTM has 90%. To avoid overfitting, control validation loss. By adding enough dropout value, overfitting can be controlled, and data quantity can affect validation loss.

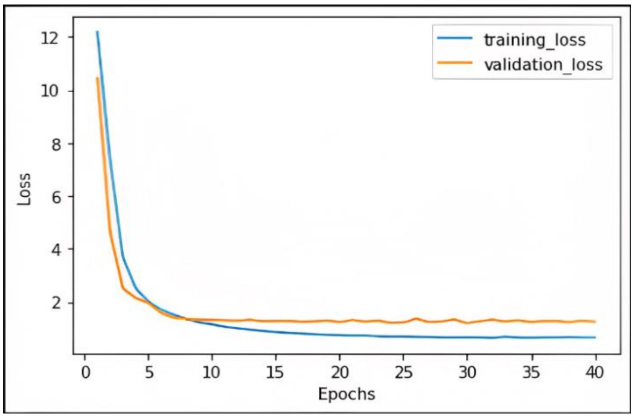

Fig. 7. Loss vs Epochs with Bi-Directional LSTM

Fig(7) plots Epochs versus Loss. Loss values are achieved after 40 epochs of training and saving the best weights with checkpoints. Training and validation loss are minimal at 30th epoch. The training loss is 0.6 and validation loss is 1.24. Training loss and validation loss are close together on the graph, indicating no overfitting. If validation loss increases while training loss doesn't, that's model overfitting. Adding a dropout layer helps to avoid overfitting.The number of neurons to be dropped from each hidden layer is dropout percentage.In proposed architecture all the dropout layers are having 25% of Dropout.

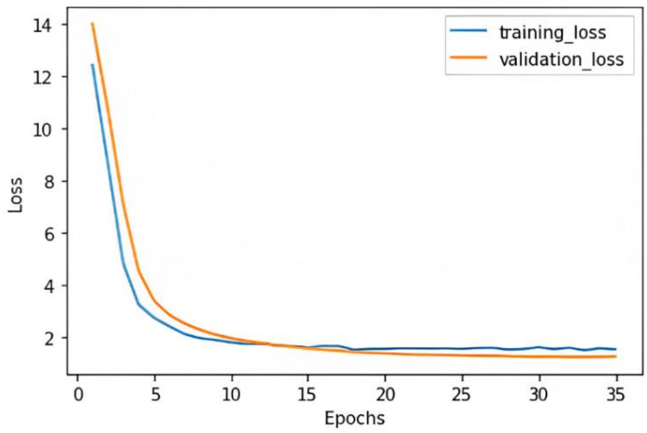

Fig. 8. Loss vs Epochs with Bi-Directional GRU

Fig(8) shows Bi-Directional GRU loss versus epochs graph. Model was trained for 35 epochs and best weights are restored. At 30th epoch model has minimal training and validation loss. Bi-Directional GRU has 1.5 and 1.3 training and validation losses respectively. Bi-Directional LSTM model has 81% accuracy, 0.96 precision, 0.83 recall, and 0.89 F-score. Bi-Directional GRU has 77% accuracy, 0.95 precision, 0.80 recall, and 0.86 F-score.

This Metric shows that Bi-Directional LSTM performs well with large training and validation datasets. As LSTMs have more gates than GRUs, the suggested method is more complex but can perform better than alternatives with fewer gates. For realtime use case, maintain quality of dataset used for training model. The current handwriting recognition model should be trained with 50 Doctors' handwriting styles. This lets the model detect any handwriting and make more accurate predictions. Training and validation datasets should have diverse calligraphy styles. This makes the model more resilient so it can properly predict most inputs.

5 Conclusion and Future Work

This model helps Pharmacists and normal people to recognize the medicine name accurately which is present in the Doctor's Handwritten Prescription. This effectively offers text in all handwriting typefaces. The feature extraction is carried out by Convolutional Neural Networks with many layers[19], and the decoding of the extracted features into English letters is assisted by Bi-LSTMs.The CTC is employed to circumvent the fact that the true alignment between the input and the output is unknown. To accurately identify language specific to prescriptions provided by the doctors, more bias is applied to words that are present in a manually produced corpus. More data increases model accuracy. Compared to Bi-Directional GRU[23], Bi-Directional LSTMs perform better and are more accurate with large datasets..

Future work: Training and validation dataset size can be increased and diverse handwriting images can be used for training in order to improve model's recognizing ability.Advanced algorithms can boost model performance.Pre trained models can be used which may boost model's performance.

References

[1] T. Jain, R. Sharma and R. Malhotra, "Handwriting Recognition for Medical Prescriptions using a CNN-Bi-LSTM Model," 2021 6th International Conference for Convergence in Technology (I2CT), 2021, pp. 1-4. (T. Jain, 2021)

[2] S. Tabassum et al., "Recognition of Doctors' Cursive Handwritten Medical Words by using Bidirectional LSTM and SRP Data Augmentation," 2021 IEEE Technology & Engineering Management Conference - Europe (TEMSCON-EUR), 2021, pp. 1-6.

[3] L. J. Fajardo et al, "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), 2019, pp. 1-6.

[4] Espana-Boquera,S., Castro-Bleda, M.J."A Spanish dataset for reproducible benchmarked offline handwriting recognition". Lang Resources & Evaluation 56, 1009–1022 (2022).

[5] Maalej,Kherallah, M. "New MDLSTM-based designs with data augmentation for offline Arabic handwriting recognition". Multimed Tools Appl 81, 10243–10260 (2022).

[6] A. Harikrishnan, S. Sethi and R. Pandey,"Handwritten Digit Recognition with Feed-Forward Multi-Layer Perceptron and Convolutional Neural Network Architectures", 2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA), 2020, pp. 398-402.

[7] A. Nikitha, J. Geetha and D. S. JayaLakshmi, "Handwritten Text Recognition using Deep Learning," 2020 International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT) (2020).

[8] M. I. Swindall et al., "Exploring Learning Approaches for Ancient Greek Character Recognition with Citizen Science Data," 2021 IEEE 17th International Conference on eScience (eScience), 2021.

[9] S. Hassan,A. Irfan,A. Mirza and I. Siddiqi, "Cursive Handwritten Text Recognition using Bi-Directional LSTMs: A Case Study on Urdu Handwriting," 2019 International Conference on Deep Learning and Machine Learning in Emerging Applications (Deep-ML), 2019.

[10] ul Sehr Zia, N., Naeem, M.F., Raza, S.M.K. et al."A convolutional recursive

deep architecture for unconstrained Urdu handwriting recognition". Neural Comput & Applic 34, 1635–1648 (2022).
[11] Alrobah, N., Albahli, S."Arabic Handwritten Recognition Using Deep Learning: A Survey". Arab J Sci Eng 47, 9943–9963 (2022).
[12] U. Shaw, Tania, R. Mamgai and I. Malhotra, "Medical Handwritten Prescription Recognition and Information Retrieval using Neural Network," 2021 6th International Conference on Signal Processing, Computing and Control (ISPCC), 2021, pp. 46-50
[13] S. Sharma and S. Gupta, "Recognition of Various Scripts Using Machine Learning and Deep Learning techniques-A Review," 2021 6th International Conference on Signal Processing, Computing and Control (ISPCC 2021) pp. 84-89.
[14] Sethy, A., Patra, P.K., Nayak, S.R. "A Deep Convolutional Neural Network-Based Approach for Handwritten Recognition System". In: Das, A.K., Nayak, J., Naik, B., Dutta, S., Pelusi, D. (eds) Computational Intelligence in Pattern Recognition . Advances in Intelligent Systems and Computing, vol 1349. Springer, Singapore (2022).
[15] Altwaijry, N., Al-Turaiki, I. Arabic handwriting recognition system using convolutional neural network. Neural Comput & Applic 33, 2249–2261 (2021).
[16] Gupta, D., & Bag, S. (2020). "CNN-based multilingual handwritten numeral recognition: A fusion-free approach". Expert Systems with Applications, pp 1-14
[17] S. Haboubi, T. Guesmi, B. M. Alshammari, K. Alqunun, A. S. Alshammari *et al.*, "Improving cnn-bgru hybrid network for arabic handwritten text recognition," Computers, Materials & Continua, vol. 73, no.3, pp. 5385–5397, (2022).
[18] H. Lamtougui, H. E. Moubtahij, H. Fouadi and K. Satori, "An efficient hybrid model for arabic text recognition," *Computers, Materials & Continua*, vol. 74, no.2, pp. 2871–2888, (2022).
[19] G. Ahmed, T. Alyas, M. Waseem Iqbal, M. Usman Ashraf, A. Mohammed Alghamdi et al., "Recognition of urdu handwritten alphabet using convolutional neural network (cnn)," Computers, Materials & Continua, vol. 73, no.2, pp. 2967–2984, (2022).
[20] K. O. Mohammed Aarif and P. Sivakumar, "Multi-domain deep convolutional neural network for ancient urdu text recognition system," Intelligent Automation & Soft Computing, vol. 33, no.1, pp. 275–289, (2022).
[21] S.Abbas, Y. Alhwaiti, A. Fatima, M. A. Khan, M. Adnan Khan et al., "Convolutional neural network based intelligent handwritten document recognition," Computers, Materials & Continua, vol. 70, no.3, pp. 4563–4581, (2022).
[22] Y. Xue, Y. Tong, Z. Yuan, S. Su, A. Slowik et al., "Handwritten character recognition based on improved convolutional neural network," Intelligent Automation & Soft Computing, vol. 29, no.2, pp. 497–509,(2021).
[23] A. Abdallah, M. Hamada, and D. Nurseitov, "Attention-Based Fully Gated CNN-BGRU for Russian Handwritten Text," *Journal of Imaging*, vol. 6, no. 12, p. 141(2020).