

# Interpreting Doctor's Handwritten Prescription Using Deep Learning Techniques

Rizwanullah Mohammad, Ajay Kumar Varma Nagaraju and Suneetha Manne

Department of Information Technology, Velagapudi Ramakrishna Siddhartha Engineering  
College, Andhra Pradesh, India

[mailtorizwanullah@gmail.com](mailto:mailtorizwanullah@gmail.com)

[ajaynagaraju32@gmail.com](mailto:ajaynagaraju32@gmail.com)

[hodit@vrsiddhartha.ac.in](mailto:hodit@vrsiddhartha.ac.in)

**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 as an input to the model and the model processes the image with deep neural network 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 Long Short Term Memory, 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 consequences 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 with the help 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].Through this paper we want to demonstrate which Architecture among

Bi-LSTM and Bi-GRU is best suitable to implement for this application of Doctor's handwritten prescription recognition[23].IAM Dataset is used for training of our model.IAM is a dataset which contains handwritings of many people.The both Algorithms i.e Bi-Directional LSTM and Bi-Directional GRU are trained with images present in IAM Dataset and the accuracies obtained are being compared and analyzed.This paper gives a brief understanding of how the quality of dataset and quantity of dataset varies the model's accuracy.

## 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.By, varying network topologies still more accuracy predictions can be produced and complexity will get reduced.

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, a data augmentation technique, increased the size of the data sets. For predicting the handwriting of doctors, an online character recognition system utilising Bi-LSTM was employed.The dataset size should be increased in order to make the model to improve its knowledge and predicting ability and varying handwriting styles should be included in dataset.

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 the image of doctors' cursive handwriting that has been captured, as well as to provide the normal text version of the handwriting.The testing image count is low and accuracy is not sufficient for health care 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.Uni directional LSTM provides less accuracy than Bi-Directional 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 may increase complexity of model and interpretability will get decreased.

A. Harikrishnan et al. [6] presented a model for handwritten digit recognition with

MLP and CNN 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 drawback here is the model was not being evaluated based on character error rate which is an important metric to make model 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 network topology in proposed architecture need to be changed i.e number of filters to be increased upto 1024 which may give better results.

Ul Sehr Zia et al. [10] proposed a model for handwriting recognition based on CRNN. This model was trained with NUST Urdu handwriting dataset. And demonstrated how model can be trained with english and urdu bilingual handwriting. Alrobah, N et al. [11] proposed applying of robust methodologies to handle challenges 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] demonstrated the usage of various scripts like devanagari, Gurmukhi and other language datasets to train the model. A model was developed with CNN architecture. A RNN usage is recommended so that model decoding of features can be done in effective manner.

Sethy et al. [14] demonstrated 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. A RNN architecture must be employed to recognize text.

S. Haboubi et al. [17] presented a model for recognition of urdu hand writing. They demonstrated how the Bi-GRUs are giving better accuracy with less complexity and consuming less memory. This is a competitor for LSTMs.

A. Abdallah et al. [23] presented a model for recognition of Russian handwritten text. They implemented CNN for feature extraction and Multi-dimensional GRU for feature decoding. They proved that Multi-dimensional GRU performance is good and we employed this model as competitor for Bi-Directional LSTM.

### **3 Proposed Work**

#### **3.1 Dataset**

The proposed model was trained with 86800 images in gray scale. IAM Dataset[1] was used which contains handwritings of different people with different fonts[16]. For training, the dataset is splitted into 90:5:5 Ratio. 90% of training data, 5% of validation data and 5% of testing data.

### 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. This 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.

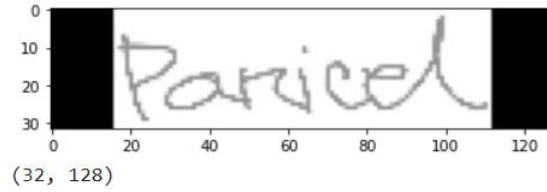


Fig. 1. Distortion free image pre-processing

### 3.3 Design Methodology

The below diagram describes architecture of our work.

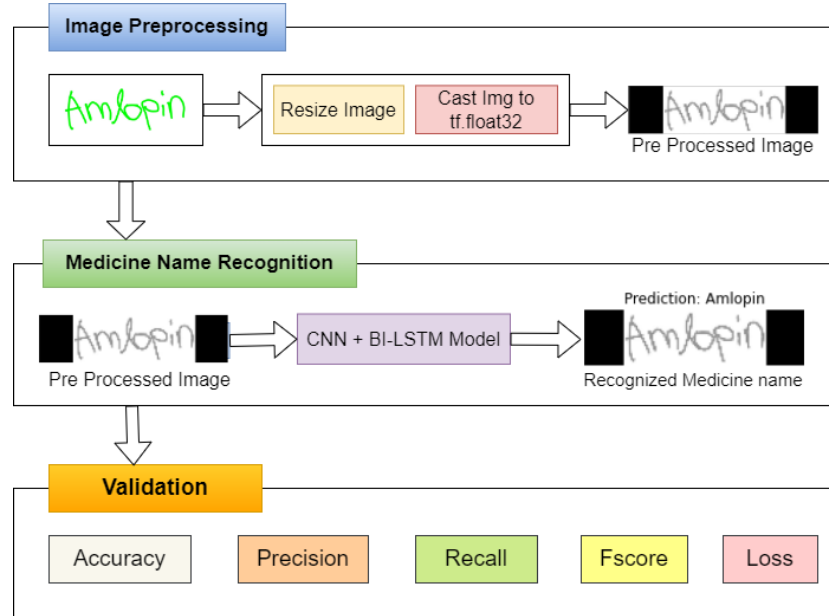


Fig. 2. Architecture Diagram for the proposed methodology

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 Bi-directional LSTM layer aids in the 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 number of correct 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 a preprocessing layer from keras which is a part of proposed methodology. It does the required preprocessings jobs like converting characters in each word while training into integers. It will build character vocabulary and maps each character in vocabulary to integer. During the prediction, the integer values are getting converted back to characters with the help of StringLookup Layer. 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-Directional LSTM:

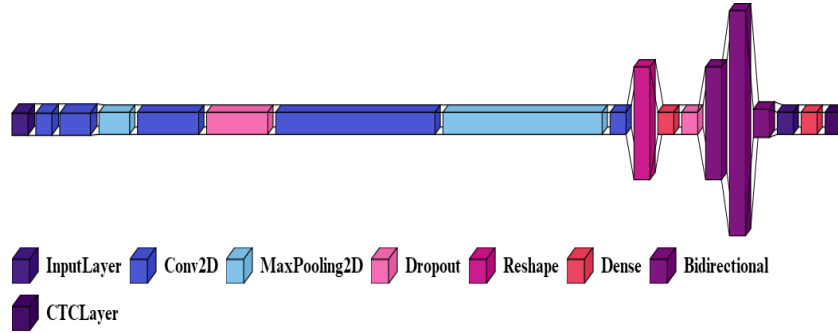


Fig. 3. Model Architecture for Bi-Directional LSTM

Fig(3) shows model design for Bi-Directional LSTM Model. The Bidirectional layer from fig(3) represents the Bi-Directional LSTM Layer. The initial layer is input layer of height 32, width 128 and the channel is 1 because its a grayscale image. Next upcoming 2 layers are convolutional layers of filter size 32,128 respectively. Next, from the above Figure one can identify different layers with filter sizes given as 3<sup>rd</sup> parameter for CNN layers. For Bi-Directional 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:

From fig (3) the first CTC Layer functionality is to recognize and remove redundancy. The CTC layer in proposed architecture will merge the similar characters which are repeated in continuous sequence. The initial phase in this layer is predicting the tokens in a sequential order. The second phase is to merge the repeated characters and drop noisy tokens. After successfully completion of phase 1 and phase 2, we can get the final output from CTC Layer. This Layer also provides Loss function in order to find loss of predicted value vs true value. This maybe either Character error rate or Word error rate.

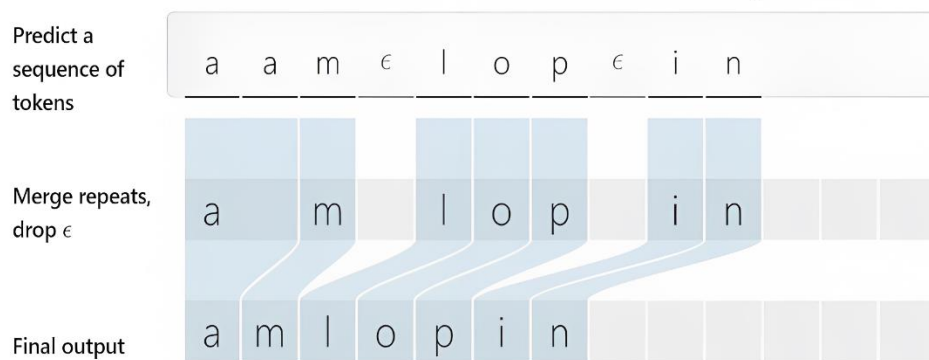


Fig. 4. Functionality of CTC Layer

Fig(4) demonstrates the working of CTC Layer. It doesn't use any traditional aligning methods, instead it will eliminate the process of Alignment. The word 'Amlopin', which is the name of a medicine when Bi-LSTM or Bi-GRU outputs it there maybe some sequence of characters coming side by side. In order to handle this, we have added this custom CTC Layer to filter characters in each word. In proposed network topology, the input and output shapes of CTC Layer are (None,32,81) and (None,32,81) respectively. The CTC loss is being calculated at every step of training. The  $y_{pred}$ ,  $y_{true}$ ,  $label\_length$  and  $input\_length$  parameters are passed as arguments to `ctc_batch_cost` function. This function will return the loss of each element.

#### CNN + Bi-Directional GRU:

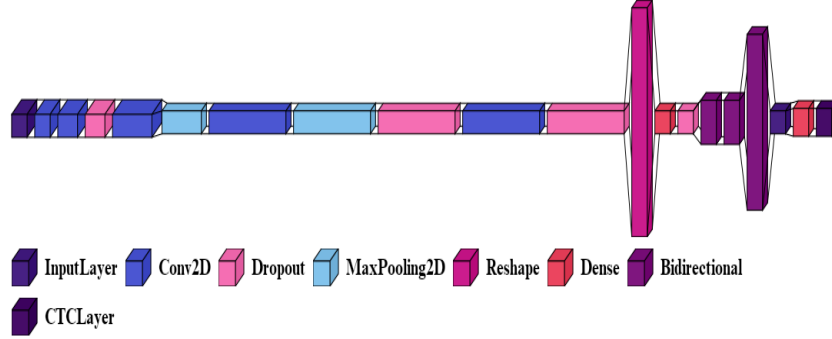


Fig. 5. Model Architecture for Bi-Directional GRU

Fig.5 shows model building phase with Bi-Directional GRU. The Bidirectional in fig(5) represents Bi-Directional GRU Layer. The input layer parameters are same as how we done for the Bi-LSTM model and it can be notice that there are some dropout layers in between convolutional and pooling layers in order to control overfitting of the model. There are total of 5 convolution layers of filter sizes 32,128,256,512,1024 respectively. And next there are 3 Bi-Directional GRU layers with hidden cells count 128,128,512 respectively. Here the CTC LAYER working is same as which we seen in Bi-Directional LSTM model.

After comparison of both models, Bi-Directional LSTM is having more number of parameters for training than Bi-GRU. It depends on our model's number of hidden units. Considering individual characters error rate is necessary in order to get good efficient and accurate model.

### 3.4 Selection of Algorithm

The implementation of our model is done with two different algorithms, they are Bi-Directional LSTM and Bi-Directional GRU. The CNN is common in both models. But the type of RNN used is different. Here the model uses algorithms of CNN + Bi-Directional LSTM[1] in first model and CNN + Bi-Directional GRU[17] in second model. Here, we are trying to make comparison between this two RNN's which is performing well. From Fig. 2 in medicine name recognition phase the algorithm we used are 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. By using IAM dataset and training data size is 86,800 Images which is large size. So, in order to handle this large data among all RNN's the Bi-Directional LSTM is best and the next one is GRU. If the dataset size is small then GRU is most preferred. The gates present in LSTM are Input, Output and Forget. Whereas in GRU the gates



present are only update and reset. It is told that GRU is less complexive as compared with LSTM. So, its better to use GRU for small sized data and LSTM for large sized data. We have selected both algorithms to test which algorithm will perform well.

The Mathematical Notation of Bi-Directional LSTM as follows:

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

From eq.(1) , The  $\hat{x}$  represents the output and  $\hat{x}^{(t)}$  represents the output at time t. 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 at time t and  $\vec{a}^{<t>}$  gives the backward hidden sequence at time t. As we are employing Bi-directional layers, we can have two directional sequences. 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:

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

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

Below are our model's testcase results. We passed some custom images to model and we got the below predictions.

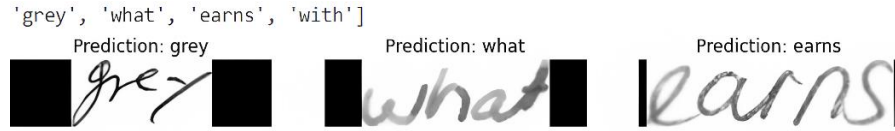


Fig. 6. Output Test Case predicted by the model

Fig(6) shows the test case results from IAM Testing dataset. The model was trained for 25 Epochs and we got this predictions. We passed 4 images custom image inputs from testing data in IAM Dataset and we got 4 images predicted correctly. Firstly, we have done preprocessing image as given in sub chapter 3.2 and passed as input to model to predict them.

#### 4.2 Observations and Analysis

The observations of Accuracies with two different models i.e Bi-Directional LSTM and Bi-Directional GRU at different epochs are noted down and displayed in below table.

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

Epochs	Accuracy (%)	
	BI-LSTM	BI-GRU
<b>20</b>	75	69
<b>25</b>	78	74
<b>30</b>	81	77

From the above table, we trained two models with Bi-Directional LSTM and Bi-Directional GRU Algorithms separately. The accuracies obtained are noted in above tabular form. At 30th Epoch Bi-Directional LSTM gave 81% Accuracy and where as at same epoch the Bi-Directional GRU gave only 77% Accuracy. It took around 5 hours to train the model upto 30 epochs with Bi-Directional LSTM Algorithm and for training Bi-Directional GRU upto 30 epochs it took 3 hours of time. Bi-Directional GRU which is less complex than Bi-Directional LSTM model got completed 100% training of 30 epochs within lesser time as compared to Bi-Directional LSTM, but Bi-GRU gave lesser Accuracy than Bi-LSTM. Where there is larger sized datasets the Bi-Directional LSTMs are performing well than Bi-Directional GRUs. So, it is recommended to use Bi-Directional LSTM by considering dataset sizes in order to get more Accuracy with a given dataset.

The 88% training data for Bi-Directional GRU model and 90% of training data for Bi-Directional LSTM model. Controlling validation loss is important to avoid overfitting of model. By adding sufficient dropout value overfitting can be controlled and also size of data we are using for validation may vary the Validation

loss accordingly.

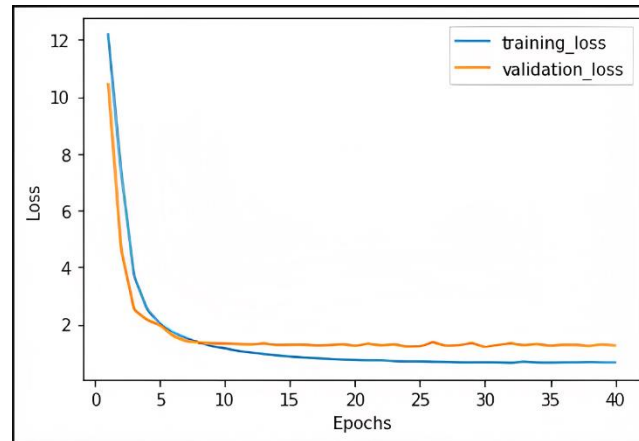


Fig. 7. Loss vs Epochs with Bi-Directional LSTM

The fig(7) shows the graph plotted between Epochs and Loss values. These Loss values are obtained when we trained our model with 40 epochs and saved the best weights with the help of checkpoints. At 30th epoch the training and validation loss are minimum. The training loss is 0.6 and validation loss is 1.24. It is clearly observed from the graph that there is no overfitting as the training loss and validation loss are very near to each other. If there is any abnormal deviation of validation loss i.e if validation loss is getting increased but training loss is fine then that case can be called as model overfitting. One of the ways to avoid this overfitting is by adding dropout layer. The number of neurons to be dropped from each hidden layer that percentage is dropout percentage.

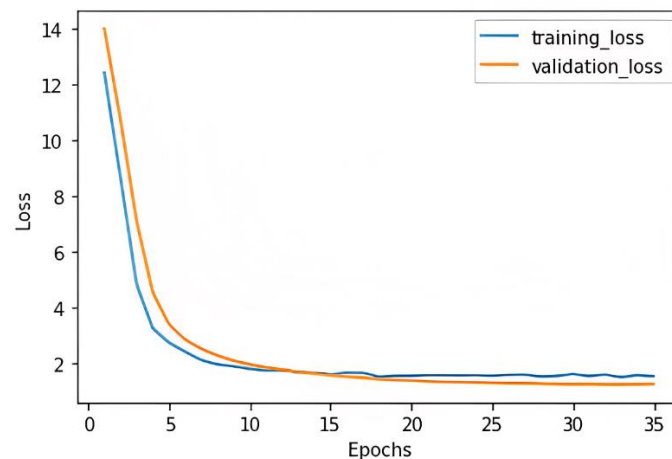


Fig. 8. Loss vs Epochs with Bi-Directional GRU

The fig(8) shows loss vs epochs graph with Bi-Directional GRU model. The model was trained upto 35 epochs and the best weights are saved that is at 30th epoch the model is giving minimum training and validation loss. 1.5 and 1.3 are training and validation losses got with Bi-Directional GRU. With Bi-Directional LSTM model achieved 81% Accuracy, 0.96 Precision value, 0.83 Recall value and the harmonic mean of precision and recall gives us F-score that is 0.89. The Bi-Directional GRU got 77% Accuracy, 0.95 Precision value, 0.80 Recall value and F-score of 0.86.

By observing this Metrics, It can be notice that Bi-Directional LSTM is performing well with large datasets for training and validation. As the LSTMs are having more gates than GRUs, always the algorithm which was proposed to implement, if its having more gates then its more complexive but it can perform better than others which is having less gates. For realtime use case the main important thing to be done is quality of dataset which is being used for training model should be maintained. The present model which is being used for handwriting recognition should be trained with minimum 50 Doctor's handwriting styles. This allows model to recognize any handwriting and it can output the correct prediction with more Accuracy. Thus, quantity of training and validation dataset and they should have different calligraphy styles. This allows model to be more robust in all cases so that it can predict maximum of given inputs correctly.

## 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. The accuracy can be maximised by increasing the size of the training data. And also when comparision is done between Bi-Directional GRU[23] and Bi-Directional LSTM, the Bi-Directional LSTMs are performing well and more accurate when dataset is of large size.

**Future work:** The Accuracy for this model can be further improved by training with more handwritten prescriptions. Further an API can be created for this model which can be useful Mobile Applications or web application to use this model for Recognizing text from Prescription or from input of cropped Medicine Name. And more layers can be added by keeping complexity of model in mind. Training same dataset with new and Advanced Deep Learning Algorithms or Advanced Neural

Networks can improve Accuracy.

## 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, doi: 10.1109/I2CT51068.2021.9418153. (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] España-Boquera, S., Castro-Bleda, M.J."A Spanish dataset for reproducible benchmarked offline handwriting recognition". *Lang Resources & Evaluation* 56, 1009–1022 (2022).
- [5] Maalej, R., Kherallah, M. New MDLSTM-based designs with data augmentation for offline Arabic handwriting recognition. *Multimed Tools Appl* 81, 10243–10260 (2022).
- [6] A. Hari Krishnan, 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 (ISPPC), 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 (ISPPC 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).