DEEP LEARNING PROJECT

By Vanshika Gupta

Title: Deep Learning OCR Model which can work with cursive handwritten data

Problem Statement:

OCR (Optical Character Recognition) is a technique of reading textual information directly from digital documents and scanned documents without any human intervention. These documents could be in any format like PDF, PNG, JPEG, TIFF, etc.

Handwritten characters are much more difficult to recognize than printed characters due to differences in writing styles for different people due to deformations, inclination, size, different handwriting styles, and incomplete strokes having a place with ligatures, continuous characters, and noise. Thus the need for deep learning that provides solutions for text information extraction from images arises.

Meta Data of the Dataset:

The IAM On-Line Handwriting Words Database (IAM-OnDB) contains forms of handwritten English text acquired on a whiteboard. It can be used to train and test handwritten text recognizers and to perform writer identification and verification experiments.

The IAM Online Handwriting Database is structured as follows:

- 115,320 isolated and labeled words
- 221 writers contributed samples of their handwriting
- more than 1700 acquired forms
- 10257 isolated and labeled text lines in on-line and off-line format
- 86272 word instances from a 11059 word dictionary

Exploratory Analysis:

1. **Preprocessing:** Real-world images are not always clicked/scanned in ideal conditions, they can have noise, blur, skewness, etc. That needs to be handled before applying the DL models to them. For this reason, image preprocessing is required to tackle these issues.
2. **Text Detection/ Localization:** Different models are used to detect text in the images. These models usually create bounding boxes (square/rectangle boxes) over each text identified in the image or a document.
3. **Padding the Images:** As the dataset contains words of various lengths (the maximum length being 21) and different sizes due to difference in handwriting, padding has been added to the images so that all of them are of the same size- 128x32 (width x height).

Project Objectives:

- The objective of OCR is to achieve modification or conversion of any form of text or text-containing documents such as handwritten text into an editable digital format for deeper and further processing.
- The application areas for the proposed character recognition system are for recognizing medicine names from doctor's prescriptions, historical document recognition, automatic reading of bank cheques, automatic postal code identification, converting handwriting to text in real-time, extracting data from filled-in forms, etc
- Makes the scanned documents completely text searchable. This helps professionals to quickly lookup numbers, addresses, names, and various other parameters that differentiate the document being searched.

Literature Review to identify models to be implemented:

S.No	**Author**	**Type of Paper**	**Published**	**Models Implemented**	**Pros & Cons**
1. *DOI:* 10.1109/TEMSCON-EUR52034.2021.9488622 *Title:* Recognition of Doctors' Cursive Handwritten Medical Words	Shaira Tabassum, Ryo Takahashi, Md Mahmudur Rahman, Yosuke Imamura,Luo Sixian, Md Moshiur Rahman, and Ashir Ahmed	Conference Paper	2021 https://ieeexplore.ieee.org/document/9488622 *Dataset:* Handwritten Medical Term Corpus Dataset	LSTM	*Pros:* Bidirectional LSTM uses both the past and the future line data to write the current line data for the parameter calculation. Writing speed, order, character shape information are also preserved in sequential data. This valuable information is difficult to get from static images. *Cons:* The handwritten data is collected from 39 doctors and medical professionals. 12 of the data providers gave incomplete data which caused 1,289

					missing data. New data augmentation method is applied on the preprocessed images to increase the number of data samples which may have caused the model to be overfit
2. *DOI:* 10.3390/s20123344 *Title:* Improved Handwritten Digit Recognition Using Convolutional Neural Networks	Savita Ahlawat, Amit Choudhary, Anand Nayyar, Saurabh Singh, Byungun Yoon	Journal Paper	2020 https://www.mdpi.com/1424-8220/20/12/3344 *Dataset:* MNIST	CNN	*Pros:* The ability to automatically detect the important features of an object (here an object can be an image, a handwritten character etc.) without any human supervision or intervention makes CNNs more efficient than their predecessors (Multi layer perceptron (MLP), etc.). The high capability of hierarchical feature learning results in a highly efficient CNN. *Cons*: The optimized CNN variant having four layers has less accuracy than the similar variants with three layers. The increased number of layers might cause overfitting and consequently can influence the recognition accuracy.
3. *DOI:* 10.1109/INCOS45849.2019.8951342	R.Parthiban, R.Ezhilarasi, D.Saravanan	Conference Paper	2020 https://ieeexplore.ieee.org/abstract/document/9262379?casa_token=I5YUu_qI1_0AAAAA:i	RNN	*Pros:* Remembers each and every information through time. Designed in such a way that it can process inputs of any

Title: Optical Character Recognition for English Handwritten Text Using Recurrent Neural Network			yiJvso6riG6In-Y-P48wJLCVx-mJiB2_pnWoe-IzivuIv6_76u2gzqpGrsap958N3oELvyIW7Q *Dataset:* Handmade Dataset		length. *Cons:* need to examine the issue further for discovering better arrangement by planning a totally new engineering for English content. Accuracy displayed by the model is 90%
4. *DOI:* 10.1007/978-3-030-39431-8_44 *Title:* Offline Arabic Handwriting Recognition Using Deep Machine Learning	Rami Ahmed, Kia Dashtipour, Mandar Gogate, Ali Raza, Rui Zhang, Kaizhu Huang, Ahmad Hawalah, Ahsan Adeel, Amir Hussain	Conference Paper	2020 https://link.springer.com/chapter/10.1007/978-3-030-39431-8_44 *Dataset:* Arabic Dataset	CNN & SVM	*Pros:* CNN as features extractor and SVM as a recognizer, in addition to the dropout technique helps protect the model from over-fitting. *Cons:* model reproduction process failed with kernel size 6x6 for the fifth layer so the experiment was conducted using the original LeNet-5 CNN architecture kernel size (5x 5). Model finds it difficult to encode the position and orientation of the object.
5. *DOI:* 10.3390/jimaging6120141 *Title:* Attention-Based Fully Gated	Abdelrahman Abdallah, Mohamed Hamada, Daniyar Nurseitov	Journal Paper	2020 https://www.mdpi.com/2313-433X/6/12/141 *Dataset:* Handwritten Kazakh and Russian Cyrillic alphabet.	CNN & BGRU	*Pros*: It has a high recognition rate, is more compact and faster, and has a lower error rate compared with the other models. *Cons:* The model couldn’t achieve a high recognition rate

CNN-BGRU for Russian Handwritten Text					at the symbol, word, sentence, and paragraph level. Demonstrates better performance on Russian and Kazakh as compared to English.
6. DOI: 10.1109/SeFeT55524.2022.9908836 *Title:* Moving Vehicle Number Plate Detection using Hybrid Deep Convolutional and Recurrent Neural Network Algorithm	P. Tejomayi Gayathri, A. Bhavana, M. Anuhya, Merugu Kavitha, N.S. Kalyan Chakravarthy, D. Mohan Reddy	Conference Paper	2022 https://ieeexplore.ieee.org/abstract/document/9908836?casa_token=O1U5HmjKRL8AAAAA:WIQsnZJp3bVLOHwgbGrtAMKqHgfBWOdsFGrNOFyLmRJUB6MXtNCPwUjdcKQJfkHuNWfng6H7hA0 Dataset: Handmade dataset containing images taken of cars from static traffic cameras	DCNN & RNN	*Pros:* Show good performance on skewed, blurry and irregular images. The model can also work through poor lighting and glare. *Cons:* Requires more pre-processing compared to other models.
7. *DOI:* 10.14738/tmlai.104.12831 *Title:* Automated Evaluation of Handwritten Answer Script Using Deep Learning Approach	Md. Afzalur Rahaman, Hasan Mahmud Department of CSE, Hamdard University Bangladesh	Conference Paper	2022 (PDF) Automated Evaluation of Handwritten Answer Script Using Deep Learning Approach (researchgate.net) Dataset: Automated Student Assessment Prize (ASAP)	CNN & BiLSTM	*Pros:* *The proposed models have the ability to perform both handwritten answers recognition and grading them as accurately as a human expert grader.* *Cons:* Developing a full-proof automated answer script evaluation system is really a big challenge since in an answer there may be a combination of figures, mathematical equations, and text

					with a variety of length, shape, and approach to the solution. In addition, evaluating handwritten answers of large size requires a complex network structure.
8. *DOI:* 10.1007/s11042-022-13320-1 *Title:* Recognition of offline handwritten Urdu characters using RNN and LSTM models	Muzafar Mehraj Misgar, Faisel Mushtaq, Surinder Singh Khurana & Munish Kumar	Journal Paper	2022 https://link.springer.com/article/10.1007/s11042-022-13320-1 Dataset: Urdu Nastalique Handwritten Dataset (UNHD)	RNN & LSTM	*Pros:* As deep learning doesn't need any feature engineering and feature extraction unlike conventional machine learning models that is why RNN and LSTM used in this paper have outperformed most of the previous attempts for character recognition. *Cons*: It has been observed that it is difficult to train the RNNs to deal with long-term sequential data, as the gradients tend to vanish. Very high accuracy and less misclassification is reported for numerals because numerals of Urdu script are quite distinguishable. From confusion matrices it is clear that miss-classifications in the case of LSTM are less than RNN.
9. *DOI:* 10.1109/ICIICT1.2019.874141	P V Bhagyashree, Ajay James, Chandran Saravanan	Conference Paper	2019 https://ieeexplore.ieee.org/document/8741412	DAG-CNN	*Pros:* DAG-CNN allows us to have multiple inputs and outputs which allow every layer to be

2 *Title:* A Proposed Framework for Recognition of Handwritten Cursive English Characters using DAG-CNN			Dataset: Handmade dataset		connected to the final classifier layer directly by using skip connections. So that all the low, mid and high level features can contribute to enhance the performance of the network. Also, all the features will be already processed by some layers and are available for free. *Cons:* Total number of features are very high, thus creating a problem of over fitting. Moreover, using basic CNN is greatly affected by noise.
10. *DOI:* 10.1109/HNICEM48295.2019.9073521 *Title:* Doctor's Cursive Handwriting Recognition System Using Deep Learning	Mideth Abisado, Joseph Marvin Imperial, Ramon Rodriguez, Bernie Fabito	Conference Paper	2019 https://www.researchgate.net/publication/340895372_Doctor's_Cursive_Handwriting_Recognition_System_Using_Deep_Learning Dataset: Samples of doctors cursive handwriting collected from several clinics and hospitals of Metro Manila, Quezon City and Taytay, Rizal.	CRNN	*Pros:* CRNNs can handle long term dependencies. Through recurrent connections, it can accumulate representations of previous input events in form of activations permitting them to model complex structures and it can have multiple layers which made them very effective in sequence modeling. Also, CRNNs require less preprocessing compared to other models. *Cons:* The deformations and loops made it difficult to recognize

					cursive characters. For future research, the location, lighting, and the distance of the image should be taken into account as it can affect the quality of the image data. Also, having more training data would be helpful.

Shortlist models for implementation:

1. CRNN
2. CNN - BiLSTM
3. CNN - BiGRU

Baseline Model:

As OCR has both spatial and sequential characteristics both convolutional and recurrent networks will be implemented.

- A **CNN** model uses images as an input. With the help of weights and biases, it extracts features from those images. If a standard Deep Neural Network(DNN) model is used to extract features from an image, the pixels from the images would have to flatten which disorients the original features in the image. This would lead to little to no accuracy when the pixel dependency is very high in the input images. Whereas, filters or kernels are used in CNN to perform convolution operations on the image to capture the spatial and temporal dependencies in the image. Max-pooling layers are used to scale down the image which helps to reduce the computational power and focus on extracting the dominant features. It also helps to reduce the noise present in the image.
- The output from CNN is fed to a bidirectional **Recurrent Neural Network(RNN)**. The recurrent layer is used to predict a label for each frame in the feature sequence. The recurrent layer has three advantages.
    1. Firstly, it can capture the contextual information from a sequence. While predicting the label for any alphabet, it may be easier to distinguish it by combining it with neighboring alphabets rather than considering them individually.
    2. Secondly, the error can be back- propagated to the convolutional layers which help to train the entire CRNN model using a single loss function.
    3. Thirdly, the input of arbitrary lengths can be fed as an input to the RNN layer.

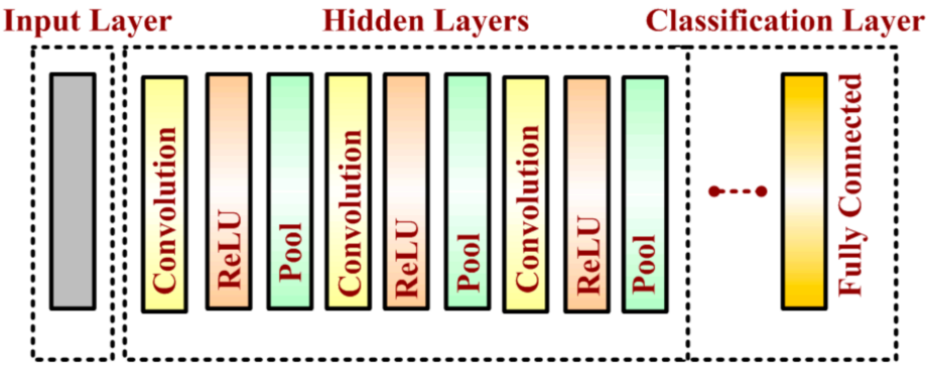


Figure 1: Structure of CNN

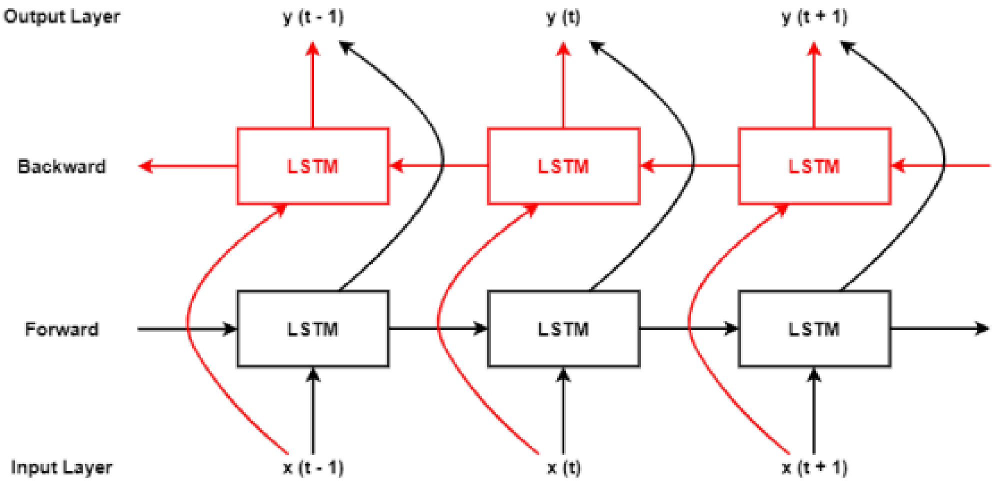


Figure 2: Bi-Directional LSTM

- **Connectionist Temporal Classification (CTC)** is a type of Neural Network output helpful in tackling sequence problems like handwriting and speech recognition. Using CTC ensures that one does not need an aligned dataset, which makes the training process more straightforward. CTC is formulated in such a way that it only requires the text that occurs in the image. We can ignore both the width and position of the characters in an image.

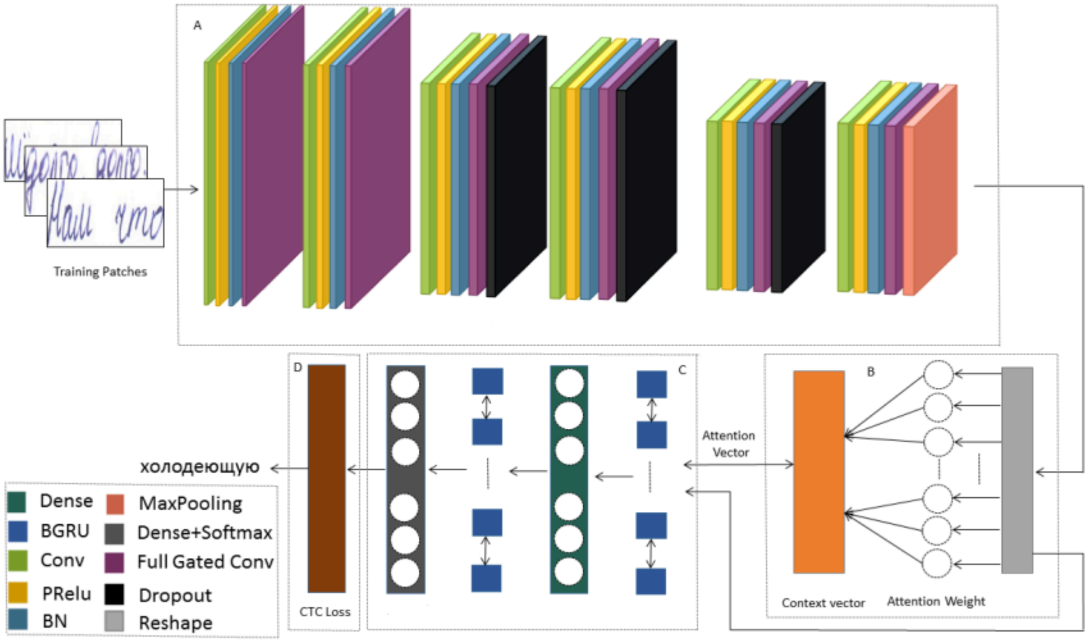
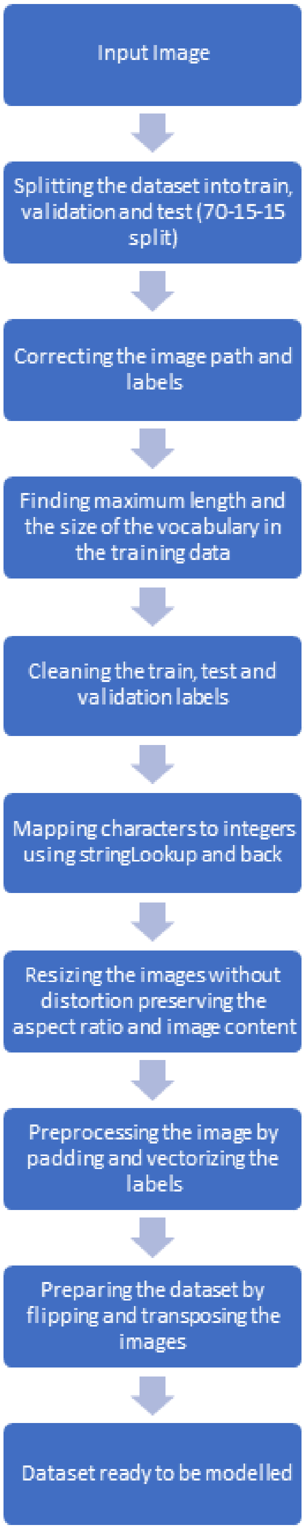
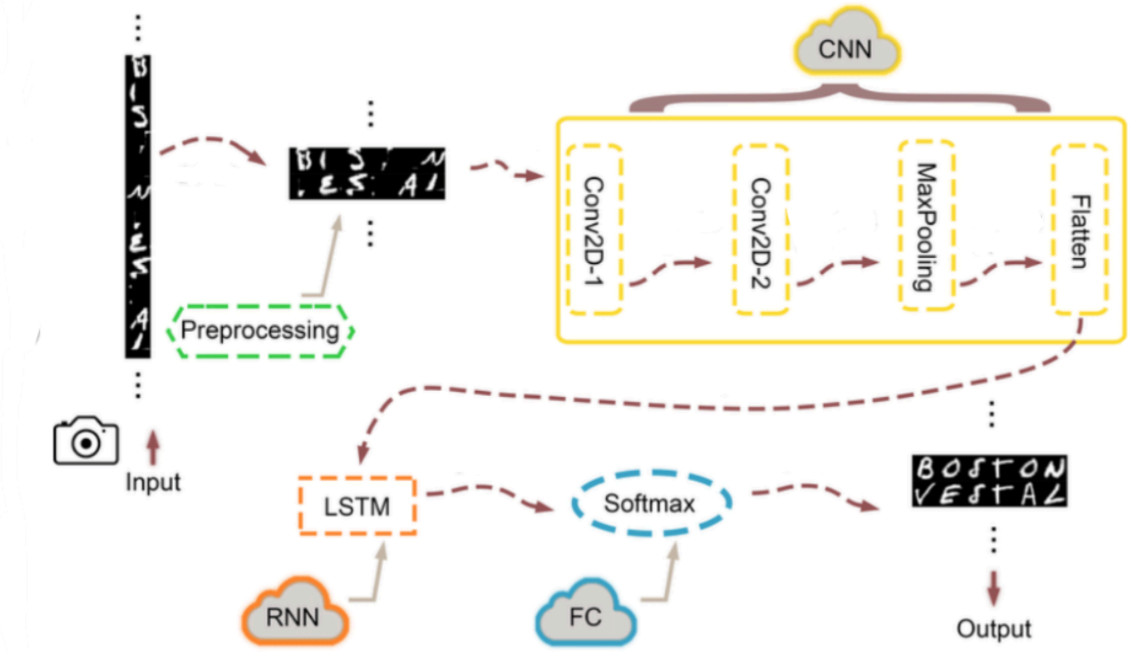


Figure 3: Architecture of CNN+BiGRU

Define working end-to-end pipeline:


Input Image
Splitting the dataset into train, validation and test (70-15-15 split)
Correcting the image path and labels
Finding maximum length and the size of the vocabulary in the training data
Cleaning the train, test and validation labels
Mapping characters to integers using stringLookup and back
Resizing the images without distortion preserving the aspect ratio and image content
Preprocessing the image by padding and vectorizing the labels
Preparing the dataset by flipping and transposing the images
Dataset ready to be modelled



(a) The general architecture of CRNN based image sequence classifier. (b) The integration of CNN unit with LSTM unit.

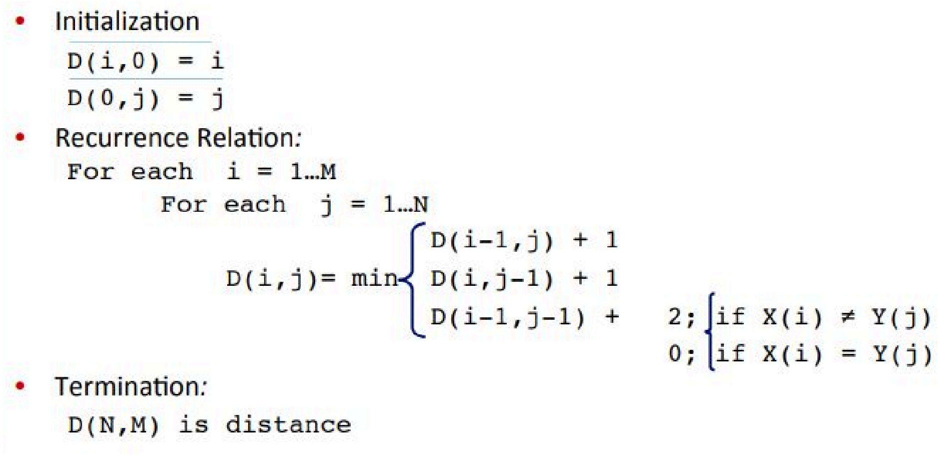
- **Data Engineering:**
  1. *Data Ingestion:* It is the process of obtaining and importing data for immediate use or storage in a database. Here we have used the IAM Online Words Database.
  2. *Exploration and Validation* - Includes data profiling to obtain information about the content and structure of the data. The output of this step is a set of metadata. Data validation operations are user-defined error detection functions, which scan the dataset in order to spot some errors. After this process we found out that the maximum word length is 21 and the vocabulary size is 78.
  3. *Data Wrangling (Cleaning)* - The process of re-formatting particular attributes and correcting errors in data, such as missing values imputation. This process includes padding the images without distortion.
  4. *Data Labeling* - In this phase, each data point is assigned to a specific category. We have cleaned and assigned labels to every image in the training and validation dataset.
  5. *Data Splitting* - Splitting the data into training, validation, and test datasets to be used during the core machine learning stages to produce the Deep Learning model.

- **Model Engineering:**
    1. *Model Training* - The process of applying the deep learning algorithm on training data to train an DL model. It also includes feature engineering and the hyperparameter tuning for the model training activity.
    2. *Model Evaluation* - Validating the trained model to ensure it meets original codified objectives before serving the DL model in production to the end-user.
    3. *Model Testing* - Performing the final model acceptance test by using the hold back test dataset.
    4. *Model Packaging* - The process of exporting the final DL model into a specific format, which describes the model, in order to be consumed by the business application.

- **Model Deployment:**
    1. *Model Serving* - The process of addressing the DL model artifact in a production environment.
    2. *Model Performance Monitoring* - The process of observing the DL model performance based on live and previously unseen data
    3. *Model Performance Logging* - Every inference request results in the log-record.

Error metric:

Edit distance quantifies how dissimilar two strings are to one another, that is measured by counting the minimum number of operations required to transform one string into the other.



```
• Initialization
    D(i,0) = i
    D(0,j) = j
• Recurrence Relation:
    For each  i = 1…M
        For each  j = 1…N
            $D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } X(i) \neq Y(j) \\ 0; & \text{if } X(i) = Y(j) \end{cases} \end{cases}$
• Termination:
    D(N,M) is distance
```

1. For the CNN + BiLSTM model, the average edit distance is 17.8659
2. For the CNN + BiGRU model, the average edit distance is 17.9216
3. For the CRNN model, the average edit distance is 18.2965

Reasoning of Hyperparameters:

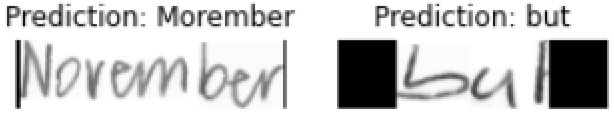
Hyperparameters are variables which determine the network structure and are set before training the model, for example: learning rate, batch size. Hyperparameter tuning to find the best configuration for a model in a high dimensional space is a tough task. The goal is to find the balance between underfitting and overfitting by examining the validation loss. We have used keras.optimizers.Adam() as the optimizer function and CTC_Loss as the loss function. The number of epochs have been fixed at 10 after a lot of trial and error.

For the first 2 layers in every model(CNN-BiLSTM, CNN-BiGRU, CRNN), the size of the input feature is (32*3*3) and the activation function for the layers is Relu. The kernel initializer is ‘he_normal’ and padding has been set to “same” to ensure the dimensions of the resultant image remain the same as the input.

Next we add the pooling layer. Here, we used the max pooling layer which computes the maximum value in a shifting frame and helps in dimension reduction. Further, dropout layer has been introduced to reduce overfitting by randomly deactivating neurons. The final dense layer has the softmax activation function culminating with the output layer measuring ctc loss.

Performance and Accuracy:

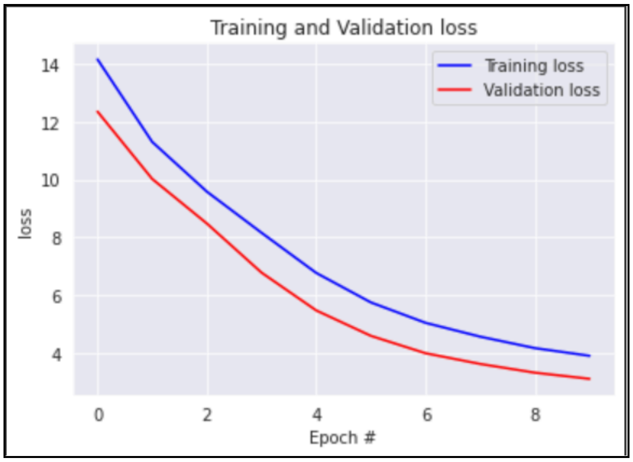
The accuracy of a deep learning model is a way to measure how often the model classifies a data point correctly. Accuracy represents the number of correctly classified data instances over the total number of data instances.
In our case, the model correctly classifies a data instance if all the letters in the word have been predicted correctly. If all the letters in the word aren't predicted correctly, it is a misclassification.
For example,



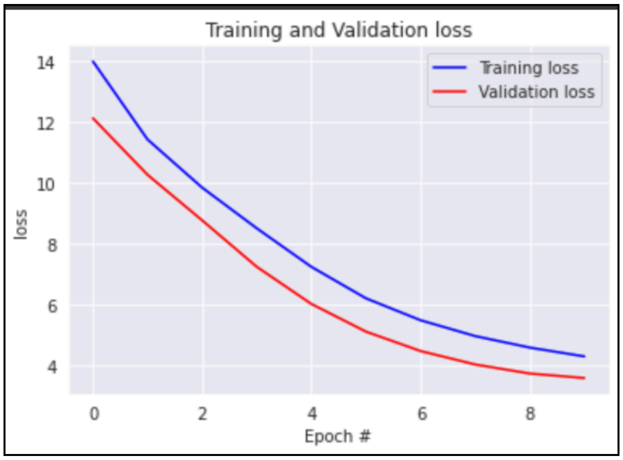
The first word is a misclassification because the prediction is ‘Morember’ instead of ‘November’. The second word has been correctly predicted as ‘but’.

Training and validation loss curves:

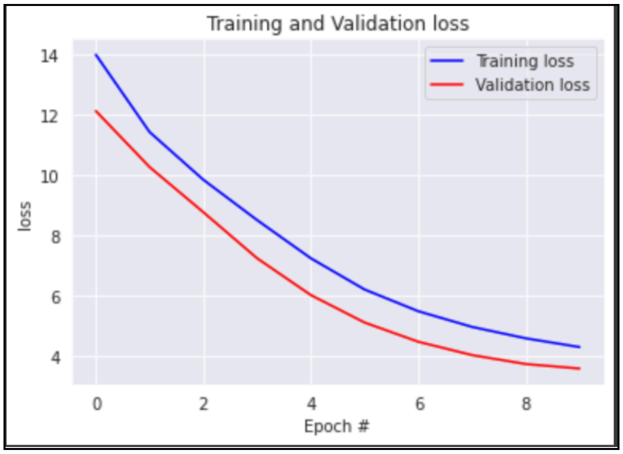
1- CNN + BiLSTM:



2- CNN + BiGRU:



3- CRNN:



Tabulation of performance:

Model	Edit Distance	Accuracy	Loss	Performance
CRNN (Convolutional Recurrent Neural Network)	19.7801	Accuracy on test dataset: 31.27%	Loss: 9.220 Val_loss: 8.648	RNNs have the concept of memory that helps them store the information from inputs to generate the next output in the sequence. However RNN's suffer from the problem of vanishing gradients. The neural network updates the weight using the gradient descent algorithm. The gradients grow smaller when the network progresses down to lower layers. Hence by the time we reach the end of the word, the information from the first few letters. Because of this, in case of longer words, CRNN only predicts the last couple of letters of the word. The reason for low accuracy is that CRNN is only able to classify words that are 4-5 letters long and punctuations like period, parenthesis, commas etc.
CNN-BiLSTM (CNN + Bidirectional LSTM)	17.8539	Accuracy on test dataset: 78.16%	Loss: 3.898 Val_Loss: 3.104	LSTM implements three gates - forget gate, input gate and output gate. During forward propagation the gates control the flow of information. They prevent any irrelevant information from being written to state. Thus it solves the problem of vanishing gradient and long range dependencies that are prevalent in CNN. Therefore CNN-BiLSTM is able to accurately predict much longer words. This model performs much better compared to CRNN but it is prone to overfitting.
CNN-BiGRU (CNN + Bidirectional GRU)	17.9604	Accuracy on test dataset: 76.39%	Loss: 4.318 Val_loss: 3.610	GRU implements two gates - update and reset gate. GRU does not possess any internal memory as it does not have an output gate that is present in LSTM. In GRU reset gate is applied directly to the previous hidden state. GRU is faster as it has fewer parameters to be trained and works better on smaller datasets.

Conclusion:

In order to modify or convert any type of text or text-containing documents, such as handwritten writing, into an editable digital format for deeper and more extensive processing, we set out to construct a deep learning model utilizing OCR technique. IAM words dataset was used in our investigation. Since the dataset consisted of handwritten text images, the majority of them were blurry, skewed, distorted, etc. We did exploratory data analysis to address these issues. Under this, we added padding, cleaned up the labels, resized the images without distorting them, etc. After that, we fed the dataset to various models. The following models were chosen for this purpose: CRNN, CNN+BiLSTM, and CNN+BiGRU. We determined these models' accuracy after training. The model correctly classifies a data instance if all the letters in the word have been predicted correctly. If all the letters in the word aren't predicted correctly, it is a misclassification. Further,since the three gates in the LSTM model address the issue of vanishing gradients and long-range dependencies that are common in CNN, CNN+BiLSTM had the best accuracy of all the models. Also as there are very few differences between their architectures, the accuracy of GRU and the LSTM model is roughly equivalent.The accuracy of CRNN, however, is somewhat low, primarily because RNN depends on long-distance signals. An RNN's state keeps track of data from all prior time steps. The current input modifies the previous information at each new timestep. The data that was stored in the early timesteps entirely changes after a considerable number of steps. Therefore, it cannot predict more than 2-3 letters correctly. Conclusively, the application of our models can be extended for recognizing medicine names from doctor’s prescriptions, historical document recognition, automatic reading of bank cheques, automatic postal code identification, converting handwriting to text in real-time, extracting data from filled-in forms, etc.