IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁒ Impact Factor 7.39 ⁒ Vol. 11, Issue 5, May 2022

DOI: 10.17148/IJARCCE.2022.115202

Handwritten Recognition with Language Translation

Dr.Maria Manuel Vianny[1], Harshitha K C[2], Keerthana L[3],Pavithra S[4] , Varshitha Y[5]

Assistant Professor, Dept of CSE,Data Science,Jain University,Bangalore, India[1]

Dept of CSE,Data Science,Jain University,Bangalore, India[2-5]

Abstract: Handwritten recognition has been one of the most challenging researches. . Major motives consist of the styles, and strokes of the huge variety of handwritings. Handwritten recognition is the cap potential of a pc to acquire and interpret handwritten entries from sources consisting of paper documents, photographs, contact screens, and different devices. Present techniques in the field of Handwritten Text Recognition are Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), Ngram, and Long-short Term Memory (LSTM). They predict the handwritten word with good accuracy.

Language translation cannot always translate a language 100% because of the slang structure and linguistics. Machine translation techniques are majorly used for language translation to get culturally and linguistically appropriate translations. In our project, we have used CNN and LSTM (BLSTM) for handwritten recognition and For Language recognition, we have used Encoder-Decoder using LSTM. Our main objective is to combine handwritten recognition and language translation to interpret handwritten words appropriately and translate them into one of the native languages in India (Hindi) acquiring good accuracy.

Keywords: Handwritten recognition, IAM dataset, Language Translation

I INTRODUCTION

Handwriting started as pictographs and slowly emerged as scripts. Handwritten text is used in our daily lives such as handwritten notes, ancient writings, memos, whiteboards, medical records, historical documents, text images, etc. Every individual has unique handwriting, even the forgery ones are not 100 percent accurate. Therefore, it is very difficult for everyone to understand every handwriting. Due to this wide range of Handwritten text, there is a need for research in the area of handwriting recognition systems for many languages and scripts. Many of the recent offline handwritten text recognition (HTR) systems have adopted line-level recognition strategies that use a combination of convolutional neural networks (CNN) and Long Short-Term Memory (LSTM) recurrent neural networks for feature extraction, trained with Connectionist Temporal Classification (CTC) loss. Language translation is the key that providing the adequate translation of a foreign language to one's own native language. Machine translation (MT) is automated translation. It is the process used to translate a text from one language to another

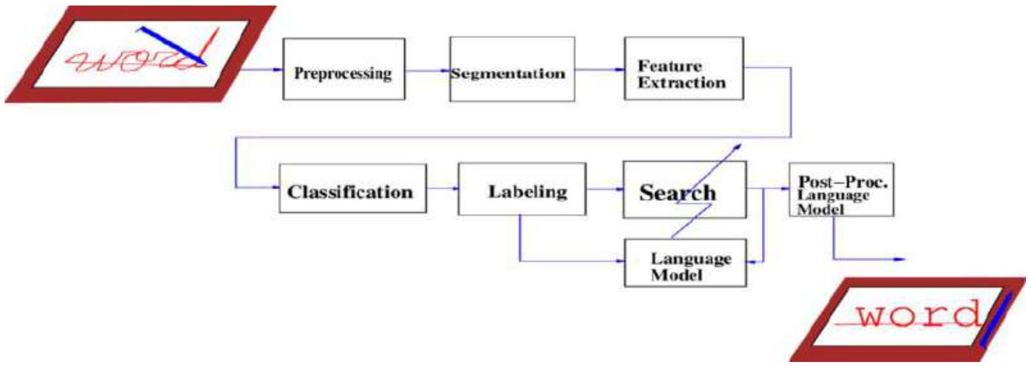


Figure.1 Working diagram for handwritten recognition

It has been remarkable progress that linguists and computer engineers have worked together to achieve the current status of machine translation. The Machine Translation task was initially handled with dictionary matching techniques and slowly upgraded to rule-based approaches. During the last two decades, most machine translation systems were based on a statistical machine translation approach.

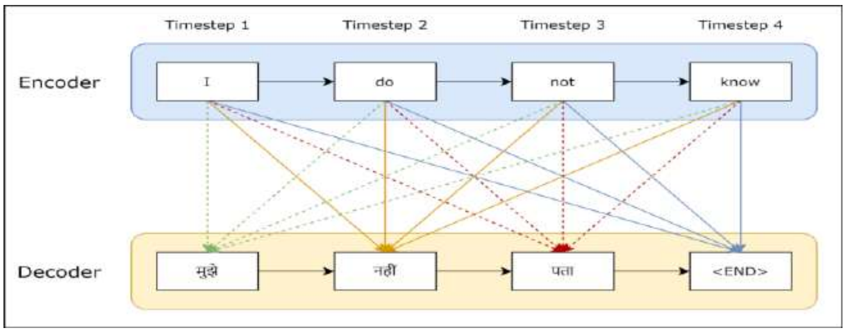


Figure.2 Working diagram for language translation

II SEQUENCE DIAGRAM

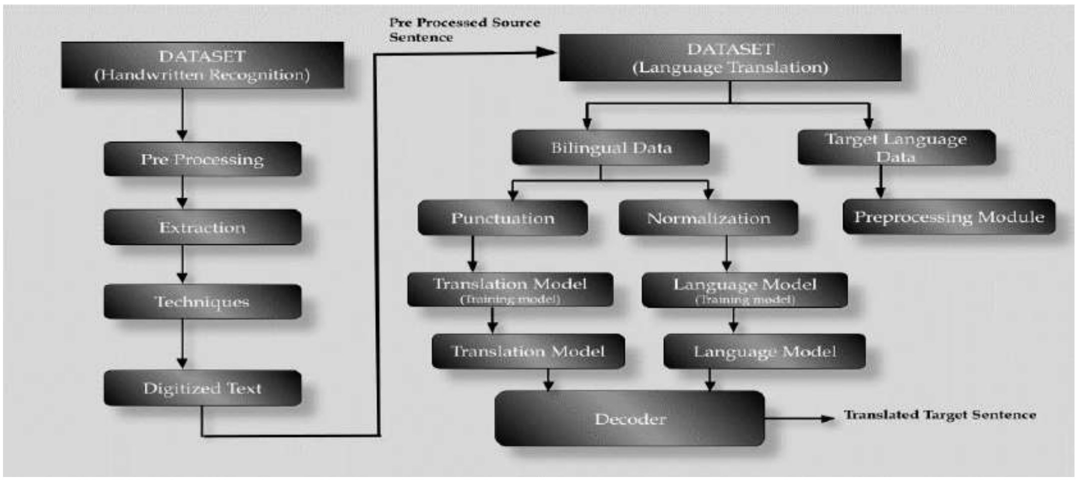

DATASET (Handwritten Recognition)
Pre Processing
Extraction
Techniques
Digitized Text
Pre Processed Source Sentence
DATASET (Language Translation)
Bilingual Data
Target Language Data
Punctuation
Normalization
Preprocessing Module
Translation Model (Training model)
Language Model (Training model)
Translation Model
Language Model
Decoder
Translated Target Sentence

Figure.3 Sequence diagram

III METHODOLOGY

Handwritten Recognition

The dataset used for handwritten recognition is the IAM Handwritten Dataset which contains forms of handwritten English text which can be used to train and test handwritten text recognizers and to perform writer identification and verification experiments.

The database used contains 1539 pages of scanned text sentences written by 600+ writers. This project uses the top 50 writers with the most amount of data. Dataset is full of sentences. Data is grouped by writers having written a collection of sentences.

It consists of a total number of 9'862 text lines. It provides one training, one testing, and two validation sets.

Feature Extraction using CNN: Convolutional Neural Networks (CNNs) can mechanically examine the critical visible capabilities from images. CNN is constructed with distinct layers which include the convolution layer, max-pooling layer, activation functions, etc. The convolution layer extracts the functions primarily based totally at the kernel that's mechanically learned.

Sequence Labelling the usage of BLSTM Recurrent Neural Networks (RNNs) is the version of neural networks in which the relationship among numerous nodes follows a temporal order to process the temporal and sequential data. The Connectionist temporal classification (CTC) loss is taken into consideration as the goal characteristic for the training of the model. The CTC loss works through including the possibilities of all likely alignments among the label and the input.

Language Translation:

The dataset used for English to Hindi language translation is the output of handwritten recognition which consists of 42600 images that are then digitized to text by removing duplicates. It consists of 6566 samples. Pre-processing is the basic phase of character recognition and it's crucial for a good recognition rate.

Padding is added to the frame of the image to allow for more space for the kernel to cover the image. In our study, we do zero padding to enlarge the image by adding rectangular strips of zeros outside the rectangular edge of the image, so that we get a new larger rectangular image with a black frame around it

A neural machine translation system consists of an encoder representing a source sentence and an attention-based decoder that produces the translated sentence. Word embedding-based methods have been utilized in many different tasks, using the word vectors to translate between languages. Once the word vectors of the two languages have been obtained, it builds a translation matrix that transforms the source language word vectors to the target language space. The main advantage of word embeddings is that it does not suffer from data sparsity problems.

The encoder-decoder model is a way of using recurrent neural networks for sequence-to-sequence prediction problems. The approach involves two recurrent neural networks, one to encode the input sequence, called the encoder, and a second to decode the encoded input sequence into the target sequence called the decoder. Encoder - It accepts a single element of the input sequence at each time step, processes it, collects information for that element, and propagates it forward. Intermediate vector - This is the final internal state produced from the encoder part of the model. It contains information about the entire input sequence to help the decoder make accurate predictions. Decoder - Given the entire sentence, it predicts an output at each time step.

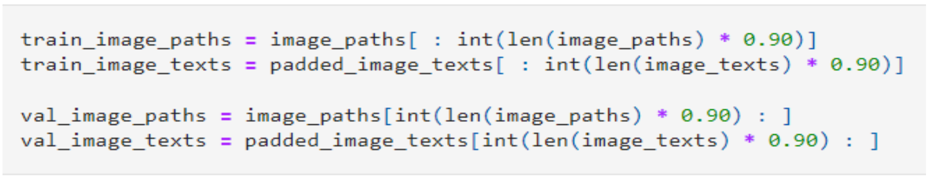
IV. IMPLEMENTATION:

Handwritten Recognition

In this project, the developed set of rules detects handwritten textual content reputation. Handwritten textual content may be located in lots of sorts of images: handwritten notes, memos, whiteboards, scientific records, historic documents, textual content enter via way of means of stylus, etc. Therefore, a whole OCR answer has to consist of assist for spotting handwritten textual content in images. This emphasizes the want for studies into the vicinity of constructing large-scale handwriting reputation structures for plenty languages and scripts. Many of the current offline handwritten textual content reputation (HTR) structures have followed line-degree reputation techniques that use a mixture of convolutional neural networks (CNN) and Long Short-Term Memory (LSTM)recurrent neural networks for characteristic extraction, educated with Connections Temporal Classification. We educated our version with the BLSTM approach and purchased the anticipated result.

i. Dataset and Data preprocessing

The IAM offline handwriting database contains scanned pages of handwritten text passages, which were written by 500 different writers using prompts from the Lancaster-Oslo/Bergen (LOB) text corpus. The database contains line images that have been partitioned into writer disjunct training, validation, and test sets containing 6161, 976, and 2915 lines, respectively. Most of the written items are individual words, partial words, or even individual letters. In most cases, the writing is in a single line, but in some cases, the data was written over multiple lines, and neural network models were implemented with TensorFlow. A standard asynchronous stochastic gradient descent with multiple workers was used to train all models. Gathered data is successfully padded and vocabs are sorted. The learning rate decayed exponentially through the training steps. The hyperparameters of the training were tuned on the development set if available; otherwise on a holdout set of the training data.



```
train_image_paths = image_paths[ : int(len(image_paths) * 0.90)]
train_image_texts = padded_image_texts[ : int(len(image_texts) * 0.90)]

val_image_paths = image_paths[int(len(image_paths) * 0.90) : ]
val_image_texts = padded_image_texts[int(len(image_texts) * 0.90) : ]
```

Figure.4 Padding images

ii. Training

The model needs line images along with their transcriptions for training. One way to get such data is to manually annotate handwritten text lines in images collected from various sources. This provides the most valuable data. However, it is time-consuming and costly, it is not trivial to find a sufficiently large number of images containing handwritten text. Thus, the availability of manually labeled data tends to be limited especially when the goal is to support many languages.

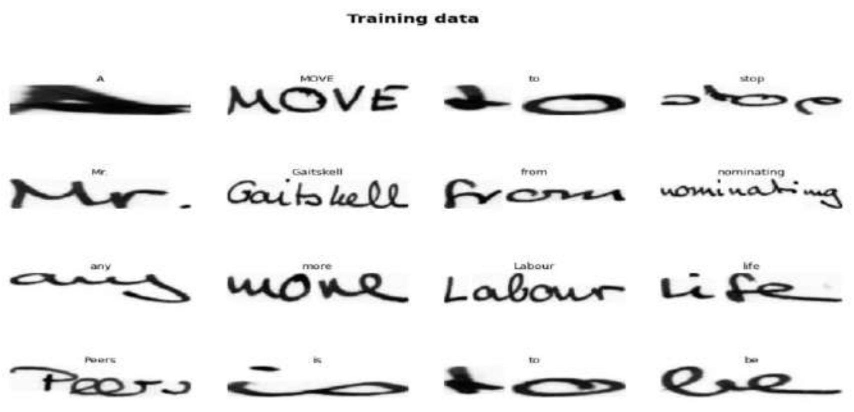


Figure.5 Training data

BLSTMS:

LSTMs and other recurrent neural networks are dominantly used for modern handwritten text line recognition models. Our model is inspired by the CLDNN (Convolutions, LSTMs, Deep Neural Network). The well-known Long Short-Term Memory (LSTM) execution of RNNs is utilized, as it can spread data through longer separations and gives more vigorous preparing qualities than vanilla RNN For the convolutional layers, we use the inception style

architecture described. For the LSTM layers, we use between one and four stacked bidirectional LSTMs (BLSTMs). In order to simulate the recurrency and gating mechanism of LSTMs with a feedforward structure, we follow an idea, we use 2-D gated recurrent convolutional layers as a fully-feedforward alternative to LSTMs. In this work, we used a bidirectional hierarchically sub-sampled RNN with Long Short-Term Memory (LSTM) architecture for the recognition of text. We trained a single-layer BLSTM network on the same dataset and observed that deep networks give higher accuracy in very a smaller number of training steps. Our approach explores a deep recurrent neural network with stacked hidden layers as BLSTM layers to use the power of long-range context as well as several levels of refined data representation obtained by using deep networks.

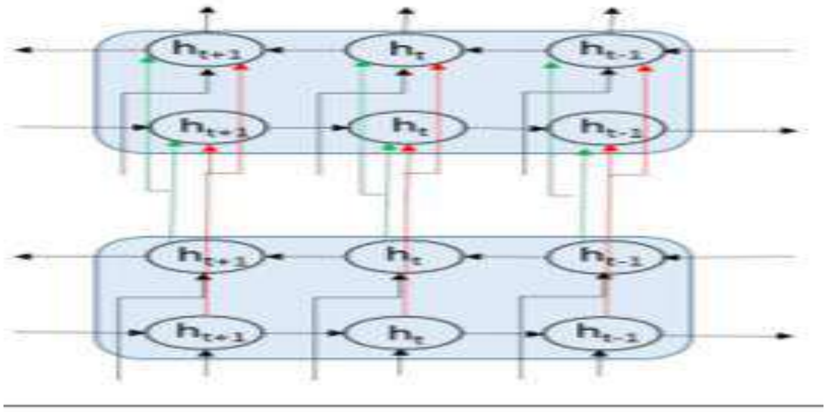


Figure.6 Block diagram of Deep Bidirectional Long Short-Term Memory

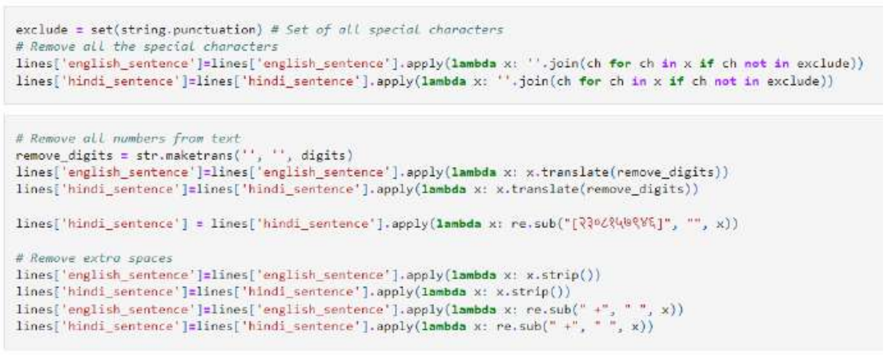
Language Translation

In this project, the developed algorithm performs language translation. The translation is more than just changing the words from one language to another. Our study is based on translating the handwritten text into the Hindi language using the deep neural technique called LSTM (Long short-term memory). To develop different experiments for evaluating the proposed approach and demonstrating its generality, we consider real-world data sets.

i. Dataset and Data preprocessing

The dataset used for English to Hindi language translation

The dataset used for English to Hindi language translation is the output of handwritten recognition which consists of 42600 images that are then digitized to text by removing duplicates. It consists of 6566 samples. Pre-processing is the basic phase of character recognition and it's crucial for a good recognition rate. Fortunately, the gathered dataset contains no null values or missing values. Splitting into train, validation & test data, for tokenization and pre-processing purposes. We create a vocabulary for both languages and removed all special characters, extra spaces, and numbers in the text. The vocabulary size for English is 3598 and the vocabulary size for Hindi is 3954. We add START_ and END_ tokens for the decoder to recognize it.



```
exclude = set(string.punctuation) # Set of all special characters
# Remove all the special characters
lines['english_sentence']=lines['english_sentence'].apply(lambda x: ''.join(ch for ch in x if ch not in exclude))
lines['hindi_sentence']=lines['hindi_sentence'].apply(lambda x: ''.join(ch for ch in x if ch not in exclude))
```

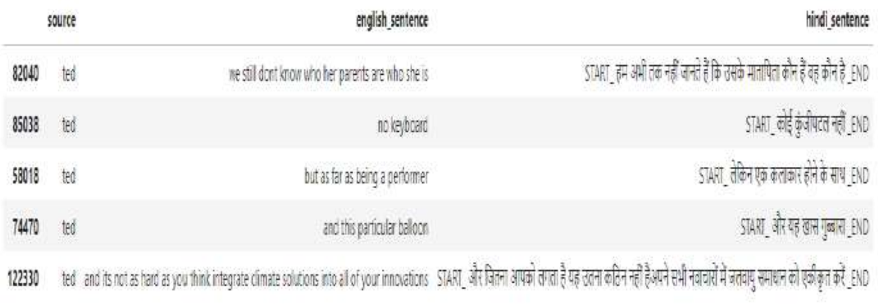
```
# Remove all numbers from text
remove_digits = str.maketrans('', '', digits)
lines['english_sentence']=lines['english_sentence'].apply(lambda x: x.translate(remove_digits))
lines['hindi_sentence']=lines['hindi_sentence'].apply(lambda x: x.translate(remove_digits))

lines['hindi_sentence'] = lines['hindi_sentence'].apply(lambda x: re.sub("[२३०८१५७९४६]", "", x))

# Remove extra spaces
lines['english_sentence']=lines['english_sentence'].apply(lambda x: x.strip())
lines['hindi_sentence']=lines['hindi_sentence'].apply(lambda x: x.strip())
lines['english_sentence']=lines['english_sentence'].apply(lambda x: re.sub(" +", " ", x))
lines['hindi_sentence']=lines['hindi_sentence'].apply(lambda x: re.sub(" +", " ", x))
```

Figure.7 Removing special characters, numbers from text, and extra spaces

DOI: 10.17148/IJARCCE.2022.115202



	source	english_sentence	hindi_sentence
82040	ted	we still dont know who her parents are who she is	START_ हम अभी तक नहीं जानते हैं कि उसके मातापिता कौन हैं वह कौन है _END
85038	ted	no keyboard	START_ कोई कुंजीपटल नहीं _END
58018	ted	but as far as being a performer	START_ लेकिन एक कलाकार होने के साथ _END
74470	ted	and this particular balloon	START_ और यह खास गुब्बारा _END
122330	ted	and its not as hard as you think integrate climate solutions into all of your innovations	START_ और जितना आपको लगता है यह उतना कठिन नहीं हैअपने सभी नवाचारों में जलवायु समाधान को एकीकृत करें _END

Figure.8 Adding tokens

Encoder and Decoder

The encoder-decoder version can be a way to use iterated neural networks for serial-to-serial prediction problems. The technique involves recurrent neural networks, one to encode the input string, called the encoder, and an ordinal to rewrite the encoded input string into the target string, called the decoder. Encoder At each step, accepts a single detail from the input stream, layers it, collects recordings for that detail, and passes it on. The intermediate vector is usually the most recent interior produced by a version district's encoder. It keeps records of the full range of inputs to help the decoder make correct predictions.Decoder Given the total set, predict the associated output at each step.

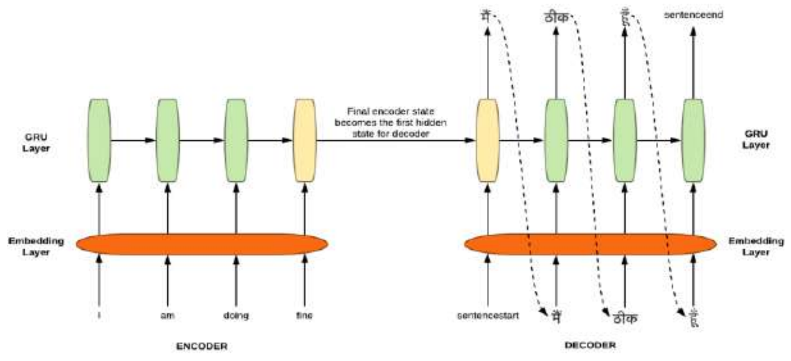


Figure.9 Encoder and Decoder structure

V. RESULTS AND ANALYSIS

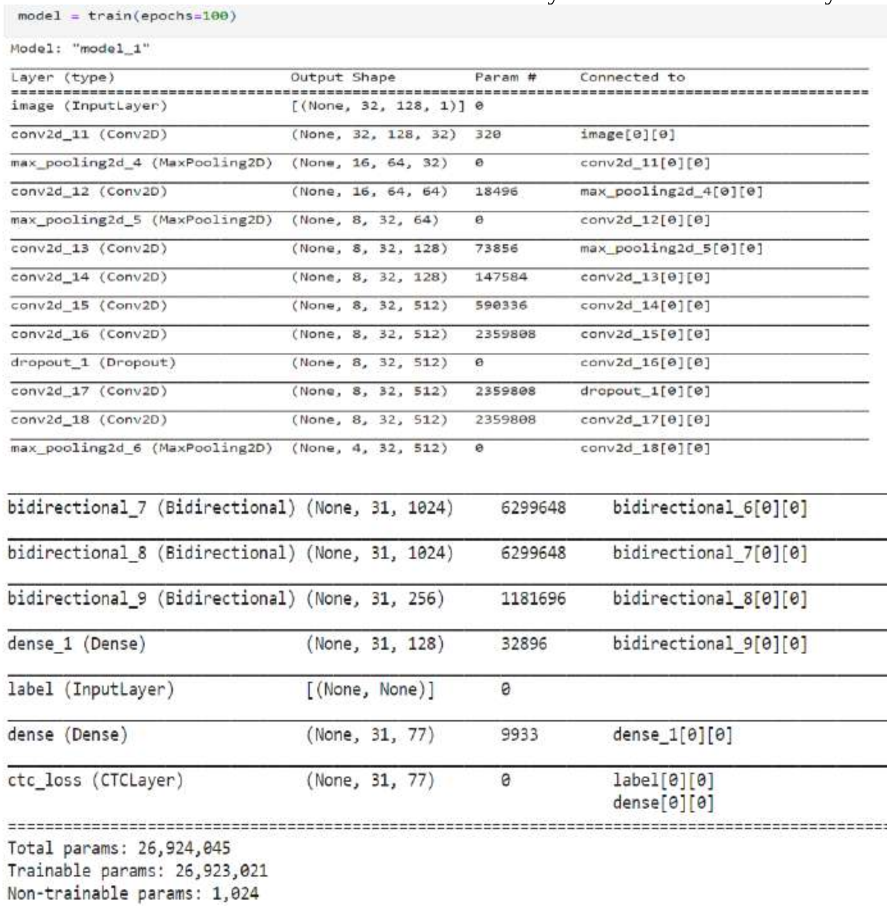
Experimental Results:

Handwritten Recognition

This structure more often than not has 3 bidirectional LSTM layers (BLSTM) used because the 3 hidden layers stacked among the enter and output layers. Bidirectional LSTM has been used in order that preceding and destiny contexts with appreciate to the contemporary role may be exploited for series studying in each the ahead and backward instructions in layers. The stacking of hidden layers enables gain better-stage function abstraction and is used for textual content recognition. This deep community is a 3-layer deep bi-directional hierarchically subsampled RNN this is composed of 3 stages: the enter layer because the 1st stage, 3 hidden recurrent layers withinside the second stage, and the output layer because the third stage. Each hidden layer is a BLSTM pair and is separated through feedforward layers with tanh as an activation function. Sampling home windows are implemented to the enter series, output of 1st hidden stage, and output of second hidden stage. Sampling window length is a turnable parameter, despite the fact that we take into account a set size for a sure layer in our experiments. As the community proceeds to better stages or next layers the window length decreases as defined withinside the phase below. The ahead by skip for this structure differs from that of an RNN, so the sum over enter is once more summed over the dimensions of the sub-sampling window. Sub-sampling is likewise required

DOI: 10.17148/IJARCCE.2022.115202

for the discount of separation among records factors withinside the enter series at better stages. This in flip offers some other benefit withinside the records illustration because the community advances to the better layers.



```
model = train(epochs=100)

Model: "model_1"

Layer (type)                    Output Shape          Param #     Connected to
==================================================================================================
image (InputLayer)              [(None, 32, 128, 1)]  0

conv2d_11 (Conv2D)              (None, 32, 128, 32)   320         image[0][0]

max_pooling2d_4 (MaxPooling2D)  (None, 16, 64, 32)    0           conv2d_11[0][0]

conv2d_12 (Conv2D)              (None, 16, 64, 64)    18496       max_pooling2d_4[0][0]

max_pooling2d_5 (MaxPooling2D)  (None, 8, 32, 64)     0           conv2d_12[0][0]

conv2d_13 (Conv2D)              (None, 8, 32, 128)    73856       max_pooling2d_5[0][0]

conv2d_14 (Conv2D)              (None, 8, 32, 128)    147584      conv2d_13[0][0]

conv2d_15 (Conv2D)              (None, 8, 32, 512)    590336      conv2d_14[0][0]

conv2d_16 (Conv2D)              (None, 8, 32, 512)    2359808     conv2d_15[0][0]

dropout_1 (Dropout)             (None, 8, 32, 512)    0           conv2d_16[0][0]

conv2d_17 (Conv2D)              (None, 8, 32, 512)    2359808     dropout_1[0][0]

conv2d_18 (Conv2D)              (None, 8, 32, 512)    2359808     conv2d_17[0][0]

max_pooling2d_6 (MaxPooling2D)  (None, 4, 32, 512)    0           conv2d_18[0][0]


bidirectional_7 (Bidirectional) (None, 31, 1024)     6299648     bidirectional_6[0][0]

bidirectional_8 (Bidirectional) (None, 31, 1024)     6299648     bidirectional_7[0][0]

bidirectional_9 (Bidirectional) (None, 31, 256)      1181696     bidirectional_8[0][0]

dense_1 (Dense)                 (None, 31, 128)      32896       bidirectional_9[0][0]

label (InputLayer)              [(None, None)]       0

dense (Dense)                   (None, 31, 77)       9933        dense_1[0][0]

ctc_loss (CTCLayer)             (None, 31, 77)       0           label[0][0]
                                                                 dense[0][0]
==================================================================================================
Total params: 26,924,045
Trainable params: 26,923,021
Non-trainable params: 1,024
```

Figure.10 Handwritten recognition Model summary

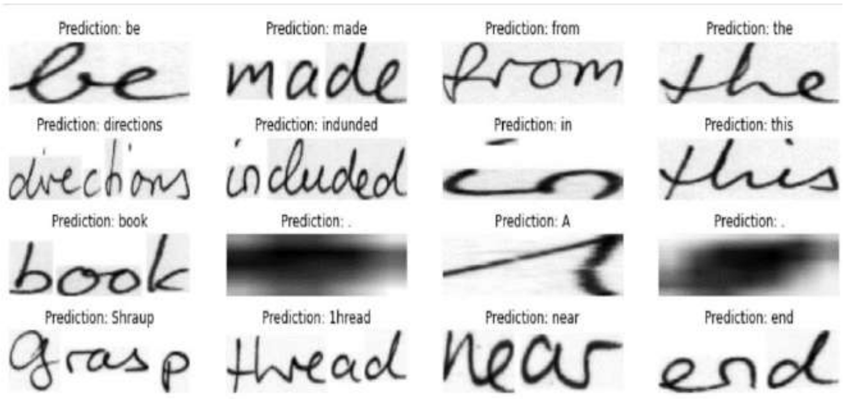


Figure.11 Experiment results

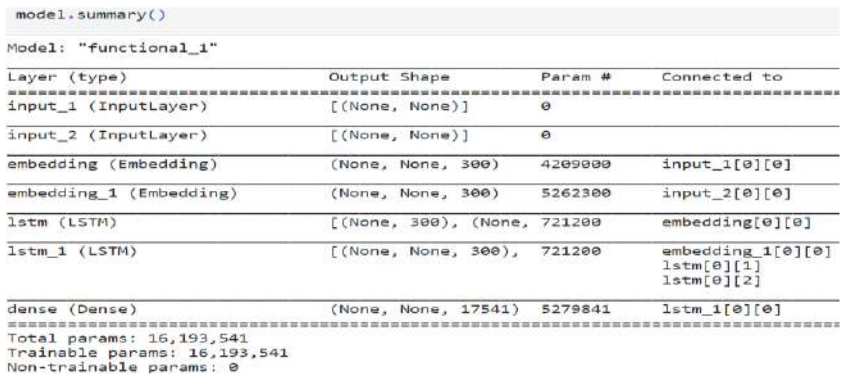
Language Translation

Experimental Results:

This section explains the planned approach for Language translation using associate degree encoder-decoder creator supported LSTM and neural network.

The model consists of 3 elements encoder, intermediate vector, and decoder. The encoder can scan a people sentence word by word and store the ultimate internal states (known as an intermediate vector) of the LSTM generated when the last time step and since the output are going to be generated once the whole sequence is read, thus outputs (Yt) of the Encoder at when step are discarded. The encoder is essentially an LSTM/GRU cell. associate degree encoder takes the input sequence and encapsulates the data because the internal state vectors. Outputs of the encoder are rejected and solely internal states are used.

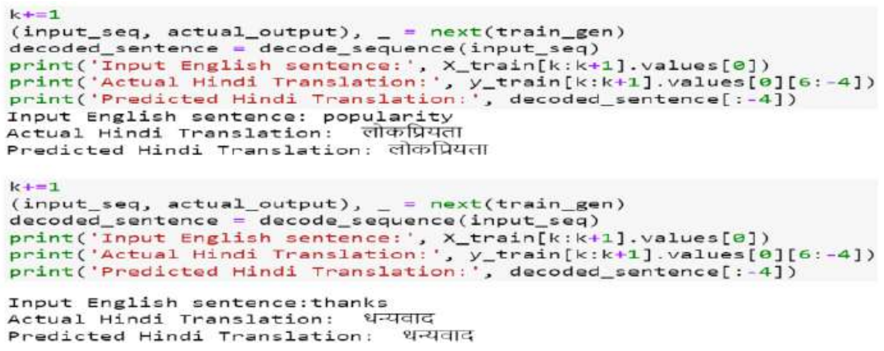
Taking the instance of translating India could be a beautiful country to its Hindi counterpart, rather like the encoder, the decoder additionally generates the output sentence word by word to come up with the output. For the decoder to acknowledge the beginning and end of the sequence, we will add START_ at the start of the output sequence and have a tendency at the top of the output sequence.



```
model.summary()

Model: "functional_1"
________________________________________________________________________________________
Layer (type)                    Output Shape          Param #     Connected to
========================================================================================
input_1 (InputLayer)            [(None, None)]        0
________________________________________________________________________________________
input_2 (InputLayer)            [(None, None)]        0
________________________________________________________________________________________
embedding (Embedding)           (None, None, 300)     4209000     input_1[0][0]
________________________________________________________________________________________
embedding_1 (Embedding)         (None, None, 300)     5262300     input_2[0][0]
________________________________________________________________________________________
lstm (LSTM)                     [(None, 300), (None,  721200      embedding[0][0]
________________________________________________________________________________________
lstm_1 (LSTM)                   [(None, None, 300),   721200      embedding_1[0][0]
                                                                  lstm[0][1]
                                                                  lstm[0][2]
________________________________________________________________________________________
dense (Dense)                   (None, None, 17541)   5279841     lstm_1[0][0]
========================================================================================
Total params: 16,193,541
Trainable params: 16,193,541
Non-trainable params: 0
```

Figure.12 Language translation Model summary



```
k+=1
(input_seq, actual_output), _ = next(train_gen)
decoded_sentence = decode_sequence(input_seq)
print('Input English sentence:', X_train[k:k+1].values[0])
print('Actual Hindi Translation:', y_train[k:k+1].values[0][6:-4])
print('Predicted Hindi Translation:', decoded_sentence[:-4])
Input English sentence: popularity
Actual Hindi Translation:  लोकप्रियता
Predicted Hindi Translation: लोकप्रियता

k+=1
(input_seq, actual_output), _ = next(train_gen)
decoded_sentence = decode_sequence(input_seq)
print('Input English sentence:', X_train[k:k+1].values[0])
print('Actual Hindi Translation:', y_train[k:k+1].values[0][6:-4])
print('Predicted Hindi Translation:', decoded_sentence[:-4])

Input English sentence:thanks
Actual Hindi Translation:  धन्यवाद
Predicted Hindi Translation:  धन्यवाद
```

Figure.13 Predicted English to Hindi Translation

VI COMPARISON OF PREVIOUS STUDIES

Handwritten Recognition

Recognition of cursive handwritten images has advanced well with recent recurrent architectures and attention mechanisms. Here, most of the works focus on improving transcription performance in terms of Character Error Rate (CER) and Word Error Rate (WER). Existing models are too slow to train and test networks.

Furthermore, recent studies have recommended models be not only efficient in terms of task performance but also environmentally friendly in terms of model carbon footprint.

Use of CNN in the deep network for offline handwriting recognition. The model structure is refined empirically. They further showed experimental results including data augmentation and image down sampling/scaling. The work showed that the proposed CNN blstm structure along with the CTC cost function enables faster handwriting recognition. Their model has fewer parameters and takes less training and testing time as compared to the state-of-the-art architectures which

ISSN (O) 2278-1021, ISSN (P) 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified ⁘ Impact Factor 7.39 ⁘ Vol. 11, Issue 5, May 2022
DOI: 10.17148/IJARCCE.2022.115202

are mainly recurrent networks. This in turn could lead to not only less carbon footprint but also better experience model uses for low-resource devices. An analysis of baseline models is also performed to show that the recognition performance is better than the other state-of-the-art methods without data augmentation.

Language Translation

Deep neural networks (DNNs) are with success applied to several areas, as well as speech and vision as a result of their ability to hit the books long-term dependency. additional recently, the Gated continual Unit [8] has additionally adopted the conception of exploitation gates.

The performance of the encoder-decoder model was evaluated on the gathered dataset and detected that the prediction of the model is incredibly correct to the particular output.

we will increase accuracy: *By exploitation bi -encoders, and more data, and *Increasing hidden layers in LSTM, and more padding.

VII. CONCLUSIONS AND FUTURE SCOPE

Handwritten Recognition

Long short-term memory (LSTM) has emerged as a awfully thriving design that has been ready to overcome the drawbacks of RNNs and is being wide used as a sturdy OCR architecture for written and written text. Deep networks have outperformed single-layer LSTM for speech recognition motivating the employment of Deep LSTM architectures for text recognition. This paper presents a Deep BLSTM architecture for the popularity of printed text. we tend to train our deep BLSTM network with 61,61,976 take a look at sets and 2915 lines. many experiments were performed by varied the amount of layers and therefore the number of hidden units in every layer observing that deep networks provide higher accuracy in terribly a way smaller range of coaching steps. At every epoch, label error drastically reduces proving the potency of deep networks. The results conferred here are obtained with 30K words for training that is promising. it absolutely was discovered that within the case of deep networks the label error drastically reduces in each epoch, though the time per epoch accrued substantially. Thus, it's evident that deep networks need lesser information and are ready to turn out higher ends up in a smaller number of epochs.

Language Translation

The vision of our project is to urge the most effective prophetic model to match to alternative studies. So, we tend to started with data, pre-processing, associate degreed padding, extracted our dataset and therefore the supply and target files are fed into the encoder layer to organize the vectors from the sentences. NMT is predicated on an easy encoder-decoder-based network. These layers interpret input word-by-word at a time which ends within the formation of a fixed-size vector illustration of the words seen therefore far. In LSTM-based NMT, we use an encoder. This encoder is based on the construct that the output at any time instant could not solely depend upon past information however conjointly on future data. The decode is intended to decode the vectors back to the target language words. This model gave associate degree expected accuracy in language translation.

FUTURE SCOPE

Handwritten Recognition

In future work, it is a must, to exploit various pretrained and fine-tuned CNNs. Yet, we are looking to apply another way to combine CNN and BLSTM networks in order to test a new model based on the Recurrent Convolutional Neural Network (RCNN).

With an increase in parameters, the need for further training data increases, but also the potential for generalization. Hence, experiments on training one deep network across several databases seem also to be a natural way to go from here.

ISO 3297:2007 Certified ⁜ Impact Factor 7.39 ⁜ Vol. 11, Issue 5, May 2022

DOI: 10.17148/IJARCCE.2022.115202

Further improving the multitask scheme for Handwritten text recognition, Including the exploratory of residual architectures and the hybrid decoding using unigrams and also other branches that are in the multitask model in combination with explicit language information.

Language Translation

Although we did not meet the target of 100% precision in identifying words, we ended up building a program with enough time and resources that could come approximately similar to that goal

Future work would involve fine-tuning the training of long and rare sentences. We would like to explore NMT for International language pairs as well. We would also analyze the performance of models for other Indian languages as the grammar and syntactic of many Indian languages are similar. Also, we look forward to making use of gated convolutional neural networks in state-of-the-art transformer architectures.

REFERENCES

[1] D. Maria Manuel Vianny, A. John, Senthil Kumar Mohan, Aliza Sarlan, Adimoolam, Ali Ahmadian. "Water optimization technique for precision irrigation system using IoT and machine learning" Sustainable Energy Technologies and Assessments, An International Journal. ELSEVIER 2022.
[2] Ingle, R. Reeve, et al. "A scalable handwritten text recognition system." International Conference on Document Analysis and Recognition (ICDAR). IEEE, 2019.
[3] Rajib Ghosh. signature verification and recognition system"
Expert Systems with Applications" Volume 168, ELSEVIER 15 April 2021.
[4] Annapurna Sharma, Dinesh Babu Jayagopi.
"Towards efficient unconstrained handwriting recognition using Dilated Temporal Convolution Network "
[5] Vasiliki Tassopoulou (Jan 11, 2021). 2961.
"Enhancing Handwritten Text Recognition with N-gram Sequence Decomposition and Multitask Learning"
25th International Conference on Pattern Recognition, Underline Science Inc. 2021.
[6] R. Parthiban, R. Ezhilarasi and D. Saravanan.
"Optical Character Recognition for English Handwritten Text Using Recurrent Neural Network"
International Conference on System, Computation, Automation, and Networking (ICSCAN), 2020.
[7] A. Yousaf et al.
"Size Invariant Handwritten Character Recognition using Single Layer Feedforward Backpropagation Neural Networks"
2nd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET), 2019.
[8] Malihe Javidi, Mahdi Jampour.
"A deep learning framework for text-independent writer identification"
Engineering Applications of Artificial Intelligence, Volume 95, ELSEVIER 2020.
[9] N. Alrobah and S. Albahli.
"A Hybrid Deep Model for Recognizing Arabic Handwritten Characters"
IEEE Access, vol. 9, pp 2021.
[10] A. K. Agrawal, A. K. Shrivas, and V. k. Awasthi.
"A Robust Model for Handwritten Digit Recognition using Machine and Deep Learning Technique "
2nd International Conference for Emerging Technology (INCET), 2021.
[11] Zhao Zhang, Zemin Tang, Yang Wang, Zheng Zhang, Choujun Zhan, Zhengjun Zha, Meng Wang.
"Dense Residual Network: Enhancing global dense feature flow for character recognition"
Neural Networks, Volume 139, ELSEVIER 2021.
[12] Nanehkaran, Y.A., Zhang, D., Salimi, S. et al.
"Analysis and comparison of machine learning classifiers and deep neural networks techniques for recognition of Farsi handwritten digits" Springer 2021.