Enhancing Handwritten Text Recognition Performance with Encoder Transformer Models

(M.Sc. Project)

Presented By

Nagendra Lal Karn

[078MSIISE012]

Supervisor

Dr. Subodh Nepal

Department of Electronics and Computer Engineering

Institute of Engineering, Thapathali Campus

August 23 2024

Outline

- Motivation
- Background
- Problem Statement
- Objectives
- Scope of Project
- Originality of Project
- Project Applications
- Literature Review
- Methodology
- Expected Results
- Remaining Task
- Tentative Schedule
- References

Motivation

- Potentially improve the accuracy and speed of recognizing handwritten text by better capturing the dependencies and patterns with their attention mechanisms.
- HTR poses unique challenges due to the variability in handwriting styles, distortions, and noise.
- Digitizing historical documents, automating data entry, enhancing accessibility for the visually impaired, and more.

Background

- HTR is a crucial aspect of digitizing handwritten documents, essential for applications such as historical document preservation, digital archiving, and automated data entry.

- Traditional HTR systems have relied on techniques such as Optical Character Recognition (OCR), which struggle with the variability and complexity of handwriting styles.

Background

- **Classical Approach** :-
  - Early methods involved feature extraction and pattern matching, but these were limited by their inability to generalize across different handwriting styles.
- **Machine learning advances** :-
  - The introduction of neural networks, particularly Convolutional Neural Networks (CNNs), improved accuracy by learning hierarchical features directly from the data.
- **Sequence Models:-**
  - Recurrent Neural Networks (RNNs) and Long Short-Term Memory networks (LSTMs) addressed the sequential nature of text, providing further improvements.

Background

- Transformer models, initially designed for NLP tasks, have revolutionized various fields due to their ability to handle long-range dependencies and parallelize computations.
- Transformer use self-attention mechanisms to weight the importance of different parts of the input, making them highly effective for recognizing patterns in a text.
- Transformers have set new benchmarks in text recognition tasks, showing promise in handling the complexities of handwritten text.

Problem Statement

- The goal of the project is to use Transformer-based architectures to overcome the shortcomings of conventional HTR models.

- Transformers have the ability to improve the precision and effectiveness of HTR because of their capacity for parallel processing and attention processes.

Problem Statement

- **Key Challenges**
- Complexity of Handwriting:
  - Handwriting varies greatly between individuals, including differences in letter shapes, sizes, and writing speed.
- Long-Range Dependencies:
  - Capturing the dependencies between distant characters or words in a sequence is crucial for accurate recognition.
- Data Scarcity:
  - High-quality, annotated handwritten text data is limited, which complicates the training of deep learning models.
- Computational Resources:
  - Training Transformer models can be computationally intensive, requiring optimization and efficient resource management.

Objective

- Implement Transformer-based Handwritten Text Recognition (HTR) System.
- Optimize Model Performance through Hyper parameter Tuning

Scope of Project

Capabilities:

- Potential to revolutionize the way we process and interpret handwritten text in a variety of fields.
- Applicable across various languages and contexts, ensuring practical deployment and integration into real-world systems.

Limitation:

- The project is limited to only hand written text recognition not able to identify writer of text.

Originality Of Project

- Enhancing model capability to interpret ambiguous handwriting through contextual understanding.

Potential Applications

- **Document Digitalization:**
  - Digitizing handwritten historical documents to preserve and make them accessible for research and education.

- **Business and Finance:**
  - Automating the extraction of information from handwritten forms, applications, and surveys to streamline data entry processes.
- **Mobile App**
  - Creating mobile apps that can recognize and digitize handwritten notes, memos, and to-do lists in real-time.

Literature Review[1]

Paper	Year	Authors	Methodology	Results	Weakness	Strengths
Handwritten Text Recognition using Deep Learning	2017	Batuhan Balci , Dan Saadati, Dan Shiferaw	The combination of CNNs and RNNs with CTC loss.	It involves a comprehensive process of data preprocessing, model architecture design, training, evaluation, and deployment.	data-related issues, model limitations, insufficient evaluation metrics, computational constraints,	advancing the field of HTR through innovative deep learning techniques
Handwritten Digits and Optical Characters Recognition	2023	Kartik Sharma, S.V. Jagadeesh Kona, Anshul Jangwal1, Dr. Aarthy M, Dr.Prayline Rajabai C, Dr. Deepika RaniSona	For training:- SVM, Random Forest, K-NN, CNN, RNN for extracting features. For Evaluation using confusion matrix and using cross validation	Addition of a hidden layer in the neural network model makes the model more accurate and efficient.	Detecting custom handwritten digits.	Recognition of handwriting is very crucial to aid automation and reduce human efforts.

Literature Review[2]

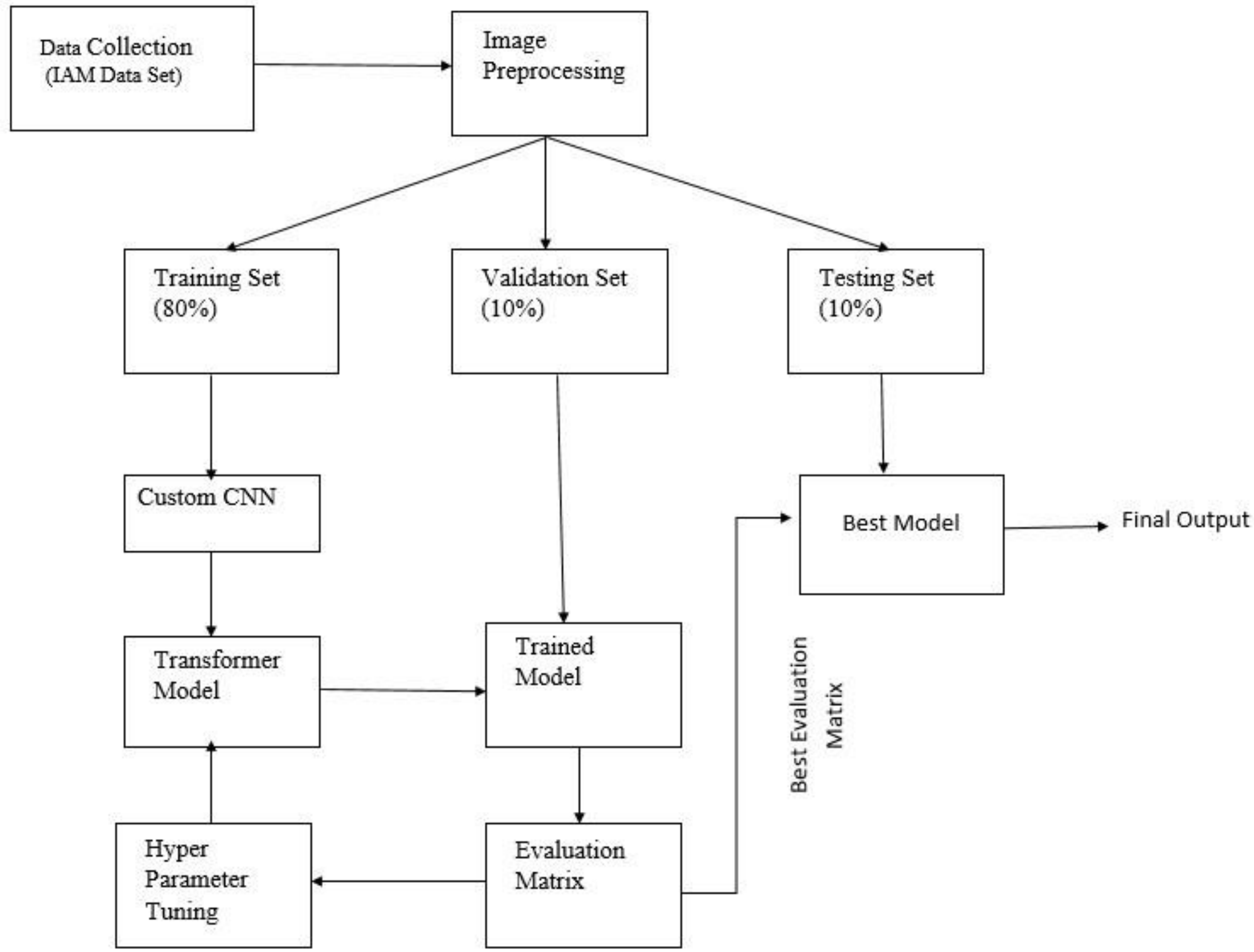
Paper	Year	Authors	Methodology	Results	Weakness	Strengths
HTR-Flor: A Deep Learning System for Offline Handwritten Text Recognition	2020	Arthur Flor de Sousa Neto,Byron Leite Dantas Bezerra, Alejandro Hector Toselli, Estanislau Baptista Lima	Use Gated CNN approach for feature extraction, also use BGRU instead of the traditional BLSTM.	Has a significantly lower CER and WER in the test partitions of each tested dataset.	Evaluation metrics in the paper may not capture different aspects such as accuracy, speed, and robustness.	Managed to combine the low complexity with the better recognition rate.
Improved Handwritten Digit Recognition Using Convolutional Neural Networks (CNN)	2020	Savita Ahlawat, Amit Choudhary, Anand Nayyar, Saurabh Singh and Byungun Yoonlakshmi	CNN_3L and CNN_4L architectures are recorded and analyzed.	CNN_3L accuracy 99.89% and using CNN_4L accuracy 99.35%	Tested only on MINIST dataset.	Avoid complex pre-processing, costly feature extraction and a complex ensemble approach.

Literature Review[3]

Paper	Year	Authors	Methodology	Results	Weakness	Strengths
Online and offline handwritten Chinese character recognition: A comprehensive study and new benchmark	2017	Xu-Yao Zhang, Yoshua Bengio, Cheng-Lin Liu	Integrating the deep convolutional neural network (con-vNet) with the domain-specific knowledge of shape normalization and direction decomposition (directMap),	comprehensive performance metrics that highlight the strengths and weaknesses of various recognition techniques.	May have limitations in the generalizability of results across diverse handwriting styles and languages.	comparison of online and offline Chinese character recognition methods and introduces a new benchmark dataset.

Methodology[1]

System Block Diagram



Methodology[2]

System Block Diagram

- Data Collection:
  - ✓ IAM English data set containing images of hand written text is used for training, evaluation and testing of the system.
  - ✓ The collected data set is pre-processed to refine the data and enhance the hand written text recognition system.
  - ✓ Images of Handwritten text is pre-processed first to fit on the image input size of model and change to grey scale too.
  - ✓ collected data sets is break down into training and testing data

Methodology[2]

System Block Diagram

- Feature Extraction:
  - ✓ CNN is used to extract the features associated with the Image containing handwritten text.
  - ✓ Then after base models are built using BERT Transformer in the Sequence modelling layer.
  - ✓ Fine tuning of hyper-parameter and optimizers are used so that better performance of the models can be achieved.

Methodology[2]

System Block Diagram

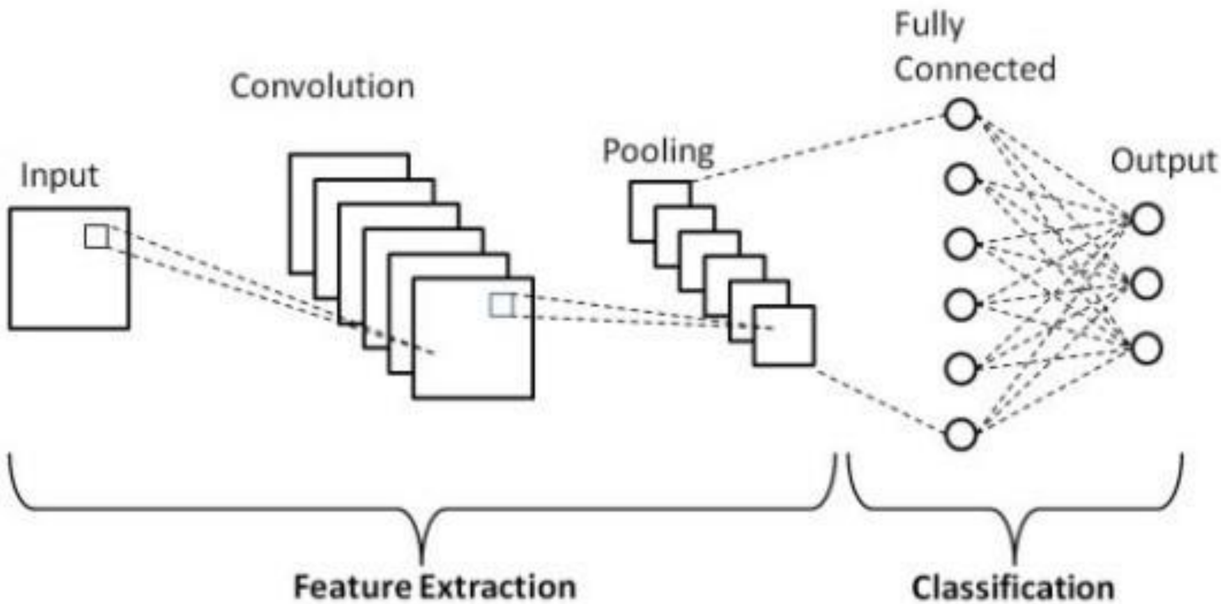
- Output Analysis:
  - ✓ Examining the results obtained from the Model to determine its performance in the context of the Accuracy, Precision, Recall and F1 score.
  - ✓ Calculating the Character error rate and word error rate for different words containing different letter size.
  - ✓ Enhance the performance using Hyper parameter Tuning and shows the best accuracy, precision recall and f1 score graph based on Training and Validation.

Methodology[3]

Theoretical Formulations

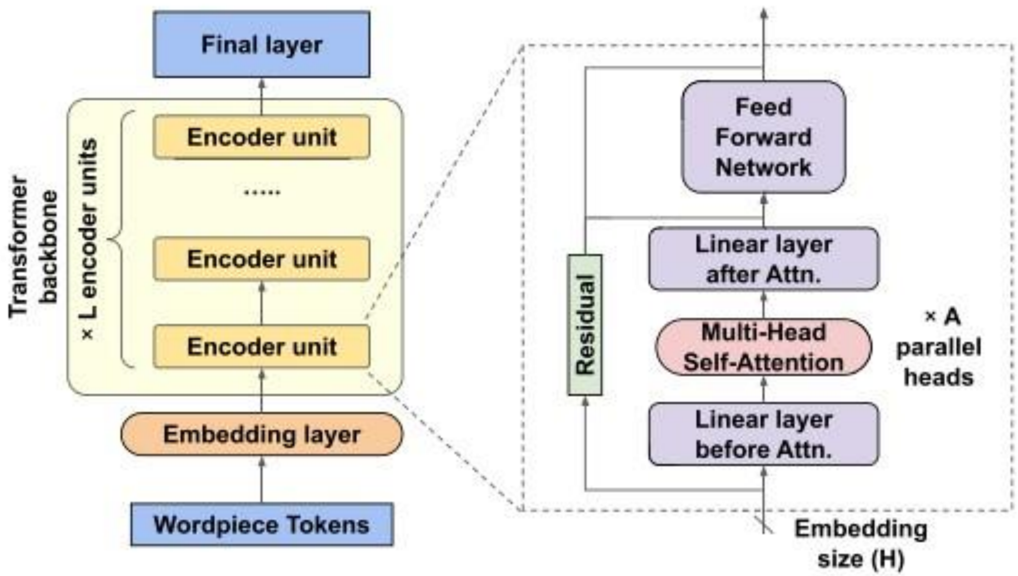
- **Convolutional Neural Network**
  - Convolutional and max-pooling layers from a typical CNN model are cascaded to create this convolutional layer combination.
  - the convolutional layers generate a series of feature vectors, with each feature vector being generated by the column on the feature maps from left to right.
  - Convolution layers, max-pooling layers, and element-wise activation functions are used in combination in a standard CNN to work on the local areas.
  - The feature map that is created at the conclusion of the CNN model is used to extract a series of feature vectors.

Methodology[3]
Theoretical Formulations



Methodology[4]
Theoretical Formulations

- **Transformer Model**
  - Model used is BERT Transformer model..



Methodology[5]

Theoretical Formulations

- **BERT Architecture**
- The encoder converts it into a fixed-length vector.
- The encoder of the Transformer architecture comprised six identical layers.
- The encoder consists of two sublayers in each of those six layers:
  - Basic feed forward network
  - Multi-head attention layer

- Every sublayer has a layer normalization and a residual connection.
- The output of the last convolutional layer is then passed through a series of transformer blocks using the transformer encoder function.
- After the transformer blocks, there's a global average pooling layer and a series of dense layers with dropout for further processing.

Methodology[6]

Dataset Explanation

- The dataset consist of 1,15,320 images of handwritten text. Among them Images contain English handwritten with digits and special characters.

- Data sets have 78 characters in all, including the image "!"#&'()*+,./0123456789:;?ABCDEFGHIJKLMNO PQRSTUVWXYZabcdefghijklmnopqrstuvwxyz," make up the handwritten text.

- Complete data is split into training images, validation and test images in the ratio of 8:1:1. And lastly model is evaluated using loss and accuracy graph.

Methodology[7]

Pre-Processing

- **Resizing**: Input image is resized while keeping the aspect ratio, with a fixed height of 32 pixels.
- **Padding**: If the resized image is smaller than (32, 128), it pads the image with white pixels to make it exactly (32, 128).
- **Clipping**: If the resized image is larger than (32, 128), it clips it to (32, 128).
- **Inversion**: It subtracts the image from 255, inverting the pixel values. This is done in when dealing with black text on a white background.
- **Channel Expansion**: It expands the dimensions of the image to have a third dimension (channel) with size 1.

- **Normalization**: By dividing each pixel value by 255, it normalizes the pixel values to be in the range [0, 1].

Methodology[6]
Hyper parameter Tuning

- Hyper parameters for Transformer Model that is used to performance enhancement.

S.N.	Parameters	Values
1.	Optimizer	SGD, RMSprop, Adam and Adamax
2	Learning Rate	0.01, 0.001, 0.0001
3	Transformer Block	2, 4, 6, 8
4	Head size	8, 16, … , 256
5	No. of heads	2, 4, 8, 16

Methodology[8]
System Requirements

- **Software Required:**
  - ➢ Google Colab
  - ➢ Keras Framework with Tensor FLOWnRF Connect
  - ➢ Pandas, Numpy

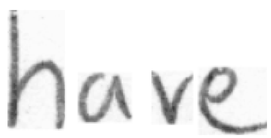
- **Hardware Required:**
  - ➢ Server with 64GB RAM 8 core Processor.

Result[1]

- **Calculating Character error rate(cer) for different words.**

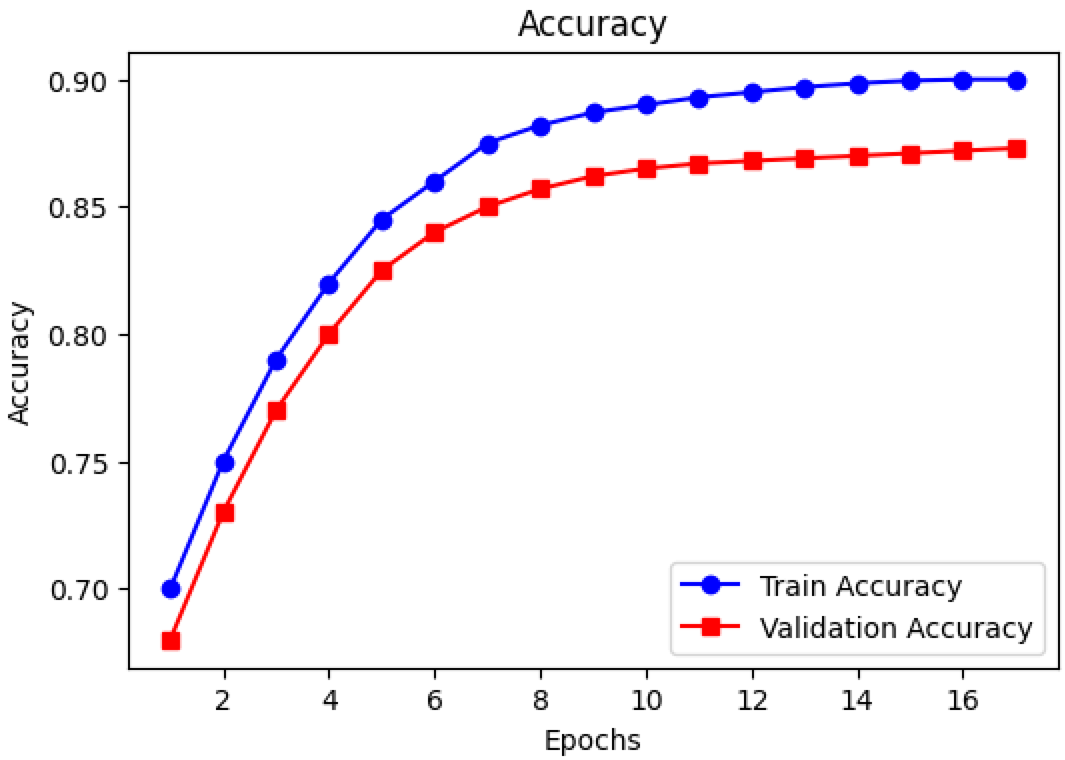
- Input image:- 

[57,66.69]
- Predicted word:- foe [57,66,56]
- Character error rate(cer) →0.33

- Input image:- 

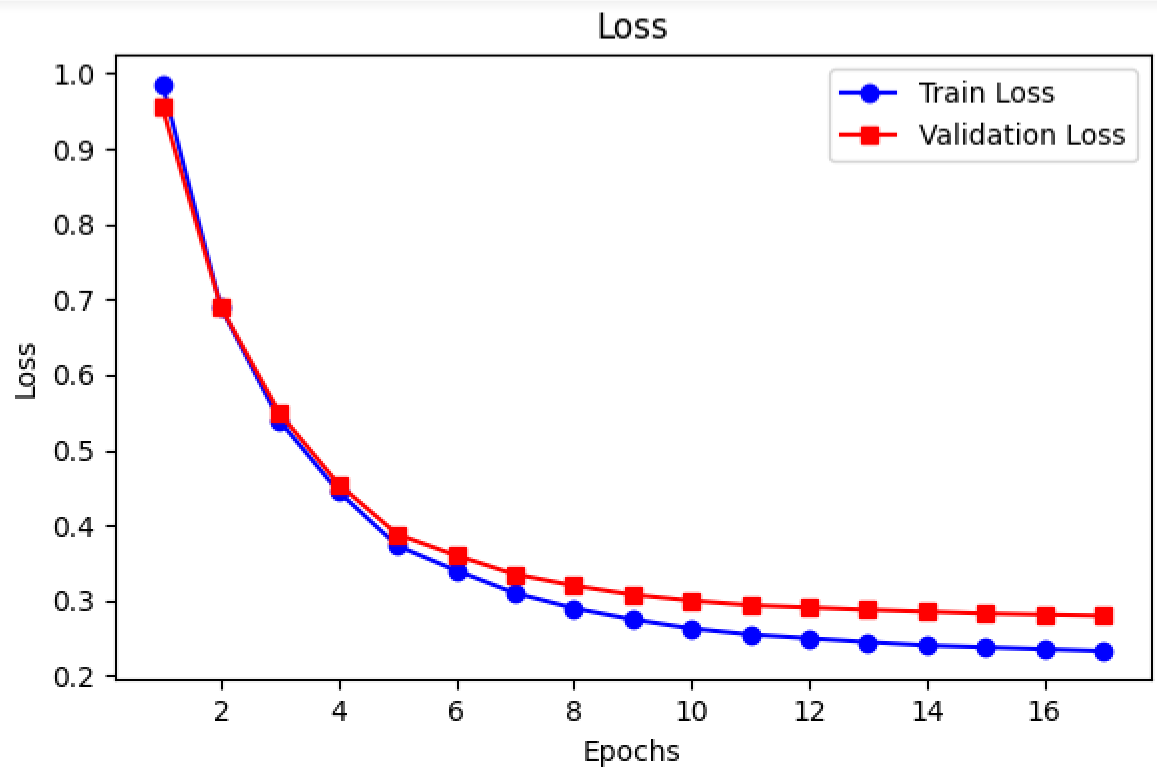
[59,52,73,56]
- Predicted word:- hare [59,52.69,56]
- Character error rate (cer) → 0.25

Result[2]



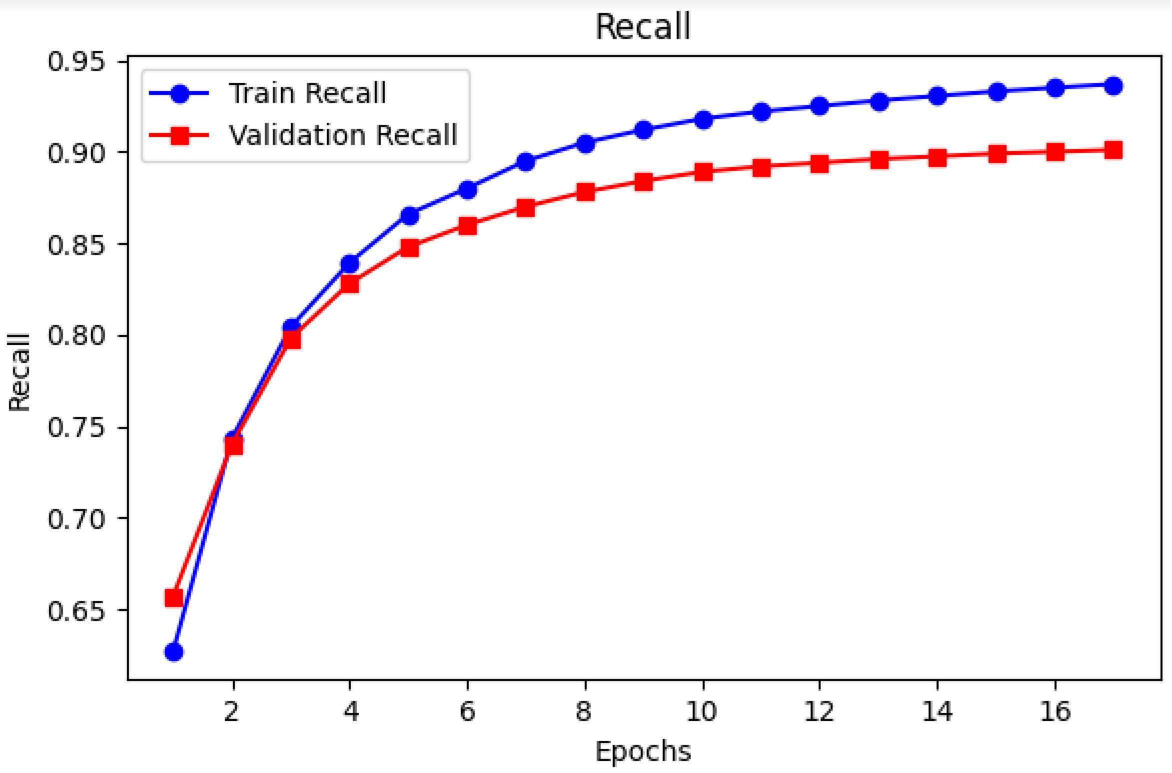
•Figure shows an accuracy of approximately 0.8620 during the testing phase indicates that the model properly classified 86.20% of the occurrences.
•As training goes on, the validation accuracy also gets better and eventually stabilizes at about 0.86, while the training accuracy keeps getting better and eventually reaches about 0.90.

Result[3]



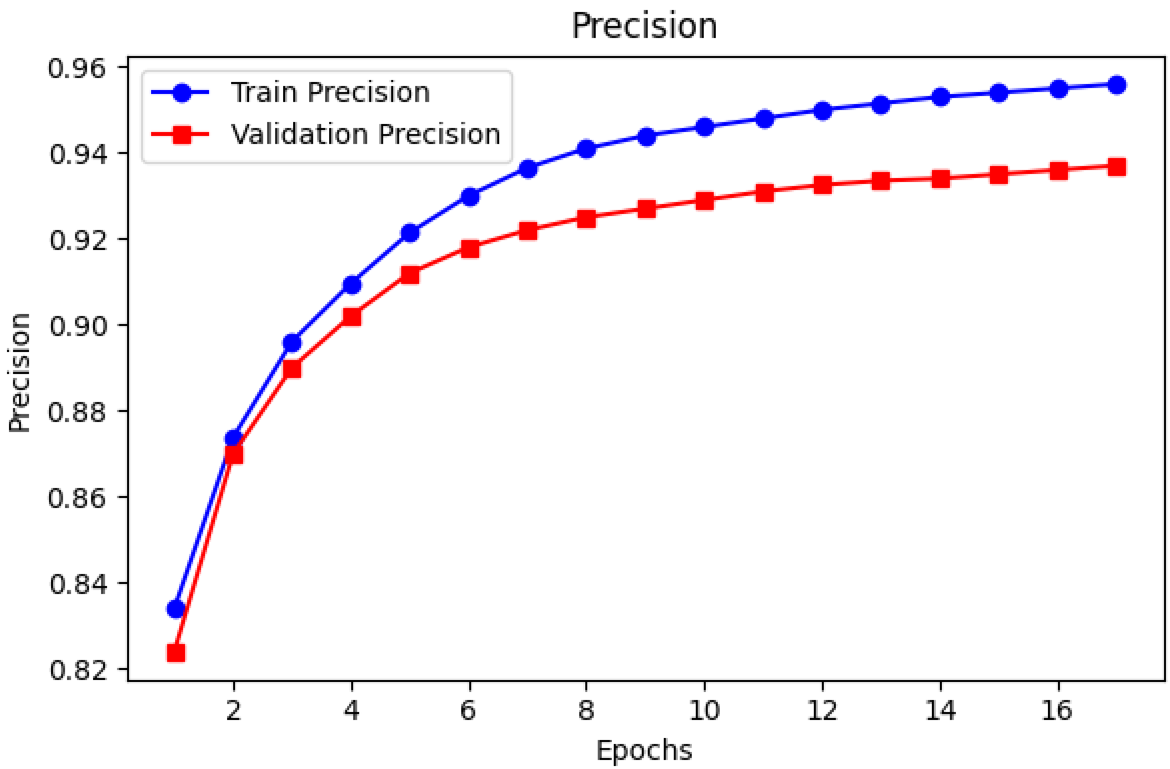
- Shows a notable decrease in both training and validation losses during the early epochs.
- This pattern indicates that the model is close to convergence.

Result[4]



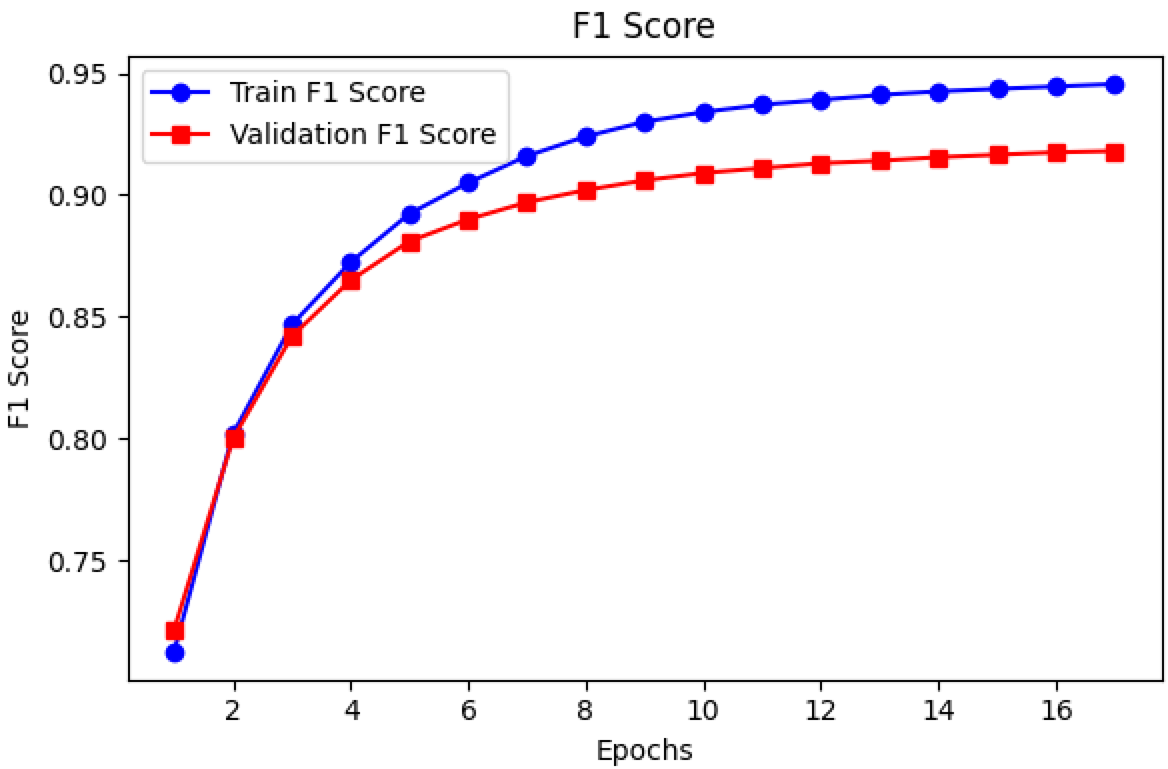
An approximate F1 score of 0.8626 suggests good balance between precision and recall

Result[5]



- Indicates that about 89.39% of the cases that were predicted to be positive really were.
- The sharp increase in precision indicates that the model picks up on character identification and classification quickly.

Result[6]



•The validation F1 score is close to 0.88 by the tenth epoch, however the training F1 score is above 0.90.

•An approximate F1 score of 0.8674 suggests good balance between precision and recall.

Discussion and Analysis[1]

- **Theoretical vs. Simulated Outputs**:
  - An accuracy of 86.26%, which is in line with theoretical expectations for a transformer-based approach on the IAM dataset.
  - The precision of 89.39% indicates that the model effectively distinguishes between different characters, confirming the robustness of the transformer encoder in capturing textual patterns.

- **Error Analysis:**
  - The presence of over fitting, as seen in the gap between training and validation accuracy/precision, suggests that the model might have learned some noise from the training data.
  - Potential sources of error include the inherent variability in handwriting styles and the limited size of the IAM dataset, which may not fully capture the diversity in real-world handwriting.

Discussion and Analysis[2]

- **Comparison with State-of-the-Art:**
  - The model's performance is comparable to recent state-of-the-art models, particularly in terms of precision, where it outperforms many traditional CNN-based methods.
  - The transformer encoder's ability to capture long-range dependencies between characters provides an edge over models that rely solely on convolution layers.

- **Performance vs. Existing Methods:**
  - The multi-head self-attention mechanism allowed to focus on different aspects of the input text simultaneously, leading to improved recognition accuracy.

Future Improvements

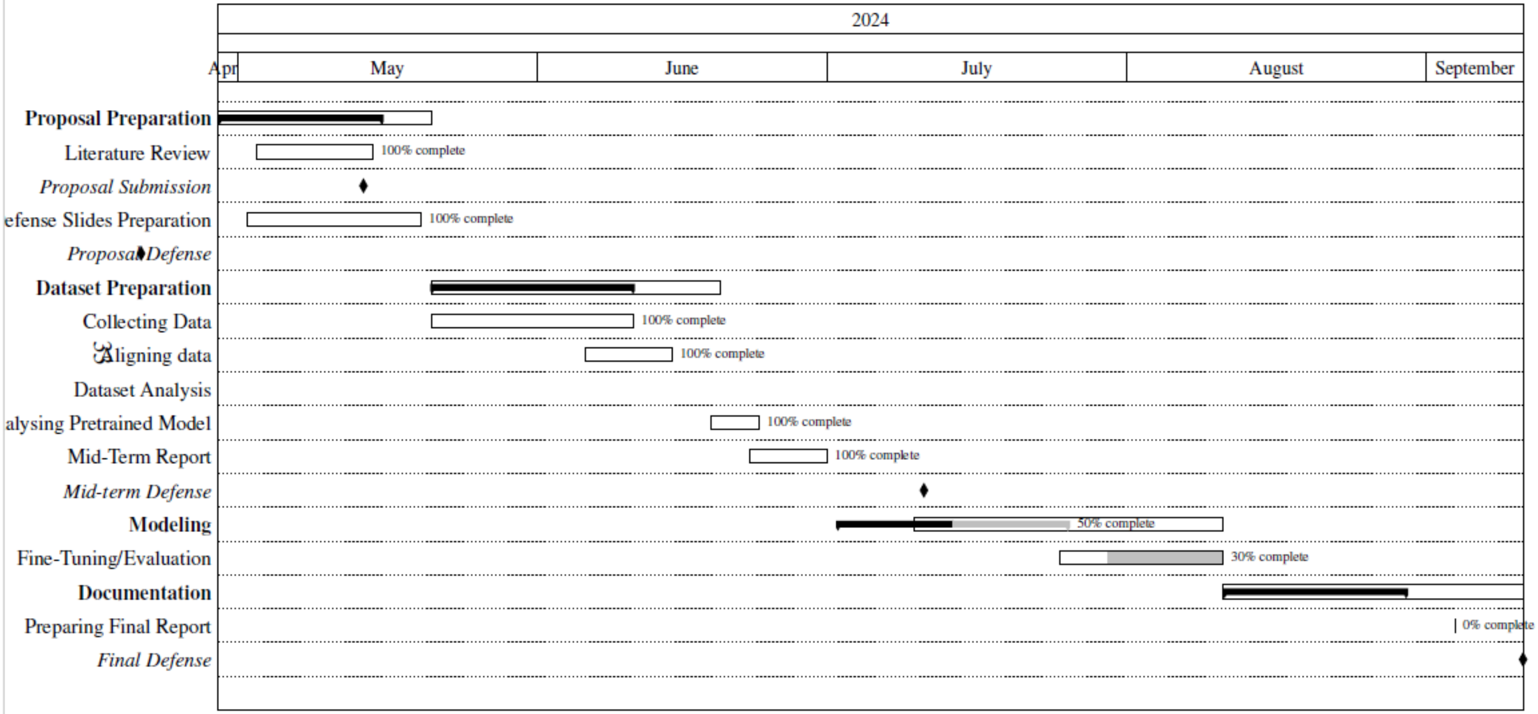
- To reduce over fitting, techniques such as more aggressive data augmentation or the incorporation of dropout layers could be explored.

- Exploring hybrid architectures, where the transformer encoder is combined with RNNs, or Meta learner models could further enhance the model's ability to handle varied and complex handwriting patterns.

Conclusion

- **Model Performance:**
  - Model accuracy and precison indicates that the model effectively recognizes handwritten text with high reliability.
- **Hyper parameter Optimization:**
  - Through hyper parameter tuning, yielded improved accuracy and balanced precision-recall metrics, showcasing the importance of fine-tuning.
- **Comparative Analysis:**
  - The model's results align well with existing state-of-the-art methods, proving the efficacy of the approach.
  - Any minor discrepancies observed can be attributed to the inherent challenges in handwritten text recognition, highlighting areas for potential future improvement.

Tentative Schedule



References - [1]

[1] Bhargav Rajyagor and Rajnish Rakhlia. Handwritten character recognition using deep learning. Int. J. Recent Technol. Eng, 8(6):2277–3878, 2020.

[2] Xu-Yao Zhang, Yoshua Bengio, and Cheng-Lin Liu. Online and offline handwritten chinese character recognition: A comprehensive study and new benchmark. Pattern Recognition, 61:348–360, 2017.

[3] A Nikitha, J Geetha, and DS JayaLakshmi. Handwritten text recognition using deep learning. In 2020 International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT), pages 388–392. IEEE, 2020.

References - [2]

[4] Sanghyuk Roy Choi and Minhyeok Lee. Transformer architecture and attention mechanisms in genome data analysis: a comprehensive review. Biology, 12(7):1033,2023.

[5] Arthur Flor de Sousa Neto, Byron Leite Dantas Bezerra, Alejandro H́ector Toselli, and Estanislau Baptista Lima. Htr-flor: A deep learning system for offline handwrit- ten text recognition. In 2020 33rd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), pages 54–61. IEEE, 2020.

[6] U-V Marti and Horst Bunke. The iam-database: an english sentence database for offline handwriting recognition. International Journal on Document Analysis and Recognition, 5:39–46, 2002.

[7] Kumar, Vaibhav. (2022). Spatiotemporal sentiment variation analysis of geotagged COVID-19 tweets from India using a hybrid deep learning model. Scientific Reports. 12. 10.1038/s41598-022-05974-6.

Thank You