Journal of Advanced College of Engineering and Management, Vol. 9, 2024

Enhancing the efficiency of deep learning models for handwritten text recognition by utilizing meta-learning optimization techniques.

Rajendra Paudyal[1*], Dhiraj Pyakurel[2*]

[1] Advanced College of Engineering and Management
Email: rajendra.paudyal@acem.edu.np, dhiraj@acem.edu.np

Abstract

Recognizing handwritten text plays a crucial role in converting scanned documents, whether printed or handwritten, into editable and searchable formats. In this study, various models such as CRNN, TCN, and Transformer have been utilized for Handwritten Text Recognition (HTR), where input data consists of sequences of image patches representing English text. The CRNN model employed comprises three layers: a CNN for extracting feature maps from handwritten text images, and Bidirectional Long-Short Term Memory (BLSTM) and Bidirectional Gated Recurrent Unit (BiGRU) in the RNN layer to address the gradient vanishing/exploding issue of simple RNNs. Additionally, TCN and Transformer models are employed for HTR. Optimizers including SGD, RMSprop, Adam, and Adamax, along with fine-tuning of hyperparameters, are utilized to enhance model accuracy. Model performance is evaluated using metrics such as f1 Score, precision, and recall. Meta learner optimization is subsequently employed to enhance the performance of deep learning models. The IAM dataset in English is utilized for training, validation, and testing. The Bi-LSTM model achieves an accuracy of 90.04%, precision of 91.62%, recall of 88.98%, and an f1 Score of 0.9025. With TCN, similar metrics are achieved. The Transformer model achieves an accuracy of 85.86%, precision of 88.94%, recall of 83.86%, and an f1 Score of 0.8626. Furthermore, Bi-GRU achieves an accuracy of 90.32%, precision of 91.56%, recall of 89.53%, and an f1 Score of 0.9050. Following the basic models, a meta model is constructed for the best performing model, demonstrating significant enhancement in Handwritten Text Recognition with an accuracy of 92.30%, precision of 94.80%, recall of 93.18%, and an f1 Score of 0.9425.

Keywords: RNN, CNN, BLSTM, BiGRU, TCN, Meta learner.

Introduction

Handwritten Text Recognition (HTR) is a field within optical character recognition focused on digitizing handwritten text. Despite digital advancements, handwritten notes remain prevalent, posing challenges for efficient recognition, storage, access, sharing, and modification. Hence, a lot of important physical form of document gets missing or damaged with age or be unreadable due to rough handling [1]. OCR is a method converting various documents, like scanned papers, PDFs, and digital photos, into editable and searchable data. Unlike human eyes, which interpret images for the brain, OCR utilizes optical systems to automatically recognize characters. While it can recognize both print and handwritten text, its effectiveness depends on the quality of input documents. Nonetheless, OCR still lags behind human reading capabilities. OCR is intended to process nearly all text in photos, with very little non-textual clutter obtained from a mobile camera image [2]. Automatic recognition of document text has been a challenging area of research for many years. Automatic document processing motivation is to convert physical real-world document into digital form. The main area of application is data extraction of documents such as commercial form, government records, historical document, Engineering document, postal documents, tabular form, bank cheques etc. [3].

Recognizing the text written by hand is a process of converting the hand written text to machine readable form. As the handwritten text varies a lot, it is quite difficult to efficiently recognize the handwritten text. Hence HTR need to be built with fundamental characteristics to efficiently extract the character from handwritten documents and translate the handwritten text to machine readable format. Nowadays, algorithms from mobile devices are used to recognize and identify properties of handwritten text, which is then translated to a machine-readable format. Similar to this, machine learning is used in many different sectors and industries, including image processing, regression, classification, medical diagnosis, prediction, and learning association [4]. With the advancement of deep learning, there has been continuous improvement in computer vision, and HTR is one of the primary fields of study. HTR is an important type of image-related series recognition problem. Numerous studies have been conducted on this field for improving HTR accuracy and performance, and new fields of applications of this powerful technology are still being explored [5]. Over the past few years, Deep Learning techniques have shown a significant improvement over conventional technique for the HTR task. Convolutional neural networks have revolutionized the field of handwriting recognition and achieved state-of-the-art results. Handwritten image categorization is a good fit for the standard convolutional neural network architecture, which consists of three convolutional layers. [6].State-of-the-art results for the HTR problem have been routinely achieved by CNN along with RNN. Convolution operation is used by the CNN's convolutional layer, which applies different filters to extract the image's information. Instead, then treating each character separately, the RNN is able to connect their relationship by capturing sequences and contextual information. Compared to a standard RNN design, the Bidirectional Long Short-Term Memory (BLSTM) RNN is utilized to capture long-term dependencies and obtain more context. [7]. Over time, many ensemble methodologies have developed, leading to improved learning model generalization. The Bagging, Boosting and Stacking categories apply to ensemble techniques in general. [8]:

Currently, meta-learning is being used extensively in computer vision and natural language processing domains such as text categorization, object identification, picture recognition, and image classification. Meta-learning technology learns a limited number of target samples to rapidly master new tasks by using meta-knowledge gathered from previous tasks as prior knowledge. This technology has significant adaptability and robustness to the unforeseen conditions, and it successfully enhances the training time and style. [9].

This research evaluates and enhance performance of Bidirectional LSTM, Bidirectional GRU, TCN and Transformer for Hand Written Text Recognition with meta learner. In this paper, three separate HTR models using BLSTM, BGRU, TCN and Transformer are used to develop ensemble learner model using majority voting technique. Again, Meta Model is implemented for handwritten text recognition for further performance enhancement. Lastly, all models were validated using precision, recall and F1 Score.

Literature Review

The practical uses of Handwritten Text Recognition (HTR) in digitizing old documents, automating data entry, and improving accessibility have attracted a lot of attention. Initially, predetermined templates of characters or strokes were compared with the input image to detect matches in a process known as Template Matching and Pattern Recognition.

To facilitate the translation of handwritten writing into a digital format, Batuhan Balci, Dan Saadati, and Dan Shiferaw [1] categorize each individual handwritten word. To complete this objective, the author uses two basic approaches: character segmentation and direct word classification. Various architectures of Convolutional Neural Networks (CNNs) are used to train a model that can precisely classify words. In addition, convolutional Long Short-Term Memory networks (LSTM) are employed to build bounding boxes for every character. Following segmentation, the segmented characters are fed to a CNN for classification, which reconstructs each word based on the classification and segmentation outcomes.

In order to classify machine-printed and handwritten text from AEC Documents, Supriya Das, Purnendu Banerjee, Bhagesh Seraogi, Himadri Majumder, Rahul Roy, Srinivas Mukkamala, and B.B. Chaudhuri [3] developed an innovative approach. Prior to Handwritten and Machine-Written ClassificationPre-processing of the printed text from the documents involves word segmentation, text graphic separation, and binarization. The words in a document are divided into segments according to specific structural characteristics of the Isothetic Covers (IC) that firmly enclose the words. A statistical examination of the related components of the document is used to select the grid size features of the IC. After that, features based on word level Gabor filters are recovered along with spooling data for categorization. The text is classified using a typical classifier that is based on SVM. This activity is completed at the word level of AEC papers with remarkable accuracy

For the purpose of handwritten Chinese character recognition (HCCR), Xu-Yao Zhang, Yoshua Bengio, and Cheng-Lin Liu [5] combined the deep convolutional neural network (con-v Net) with domain-specific knowledge of shape normalization and direction decomposition (direct Map). The author's performance on the ICDAR-2013 competition database for both online and offline HCCR is impressive.

Using Convolutional Neural Networks (CNN), Savita Ahlawat, Amit Choudhary, Anand Nayyar, Saurabh Singh, and Byungun Yoon [6] demonstrated Improved Handwritten Digit Recognition. By employing a pure CNN architecture devoid of ensemble design, Aurther attains accuracy levels that are equivalent. Using the Adam optimizer for the MNIST dataset, the CNN architecture's hyper-parameters are fine-tuned to increase accuracy.

Anuj Sharma, Lalita Kumari, Sukhdeep Singh, and VVS Rathore [7] present an attention-based handwritten text recognition system based on lexicon. The author has taken two state-of-the art neural networks systems and combined the attention mechanism with it. It achieves an accuracy of 4.15% for characters and 9.72% for words on the IAM dataset, and 7.07% for characters and 16.14% for words on the GW dataset.

Recognition of Handwritten and Printed Text of Cursive Writing Using Optical Character Recognition was first shown by Sudharshan Duth and Amulya [10]. The author recognized handwritten and printed cursive writing in the following styles: San-Serif, Tahoma, Comic Sans, and Calibri, using OCR, the k-nearest neighbor technique, and a classifier. The method identifies the aforementioned style's text in both upper- and lowercase with remarkably high accuracy.

The paper Optical Character Recognition for English Handwritten Text Using Recurrent Neural Network was featured by R. Parthiban, R. Ezhilarasi, and D. Saravanan [11]. The author uses a recurrent neural network OCR that provides remarkably accurate handwritten text recognition. Conda is used in conjunction with Tensor-Flow Framework to implement the model.

The use of a deep convolutional neural network trained on various languages for picture character identification was emphasized by Jinfeng Bai, Zhineng Chen, Bailan Feng, and Bo Xu [12]. For character recognition in images, the author employs shared-hidden-layer deep convolutional neural networks, or SHL-CNNs. The design of the network and network training are included in the SHL-CNN framework, from which an appropriate SHL-CNN model for image character recognition is empirically learned. The effectiveness of the learnt SHL-CNN is verified on both English and Chinese picture character recognition tasks, indicating that the SHL-CNN can reduce recognition mistakes by 16-30%.

Manoj Sonkusare and Narendra Sahu [13] provide an overview of English alphabet handwritten character recognition (HCR) methodologies. The author outlines the key strategies used over the past ten years in the field of handwritten English alphabet recognition. There is a thorough discussion of the various pre-processing, segmentation, feature extraction, and classification algorithms. The author points out that even though there have been advancements in methodologies over the years to address the challenge of handwritten English alphabets, more research is necessary before a workable software solution can be made available. The precision of the current handwritten HCR is really low. Therefore, a skillful solution is needed to overcome this challenge in order to maximize overall performance.

An improved offline optical handwritten character recognition using TensorFlow and a convolution neural network was presented by Prajna Nayak and Sanjay Chandwani [14]. The use of a neural network by the author with well-tried layers has the advantage of improved noise tolerance, leading to more accurate results. A wide range of feature extraction, pre-processing, segmentation, and classification techniques are presented in detail. Since Soft Max Regression yields values between 0 and 1 that accumulate to 1, it is used to assign likelihood that handwritten characters are among a variety of characters.

In order to identify the digits using deep neural networks, Kartik Sharma, S.V. Jagadeesh Kona, Anshul Jangwal, Drs. Aarthy M, Praline Rajabai C, and Deepika Rani Sona [15] use the MNIST dataset and track variations in the accuracies. A convolutional neural network for classifying handwritten digits is built using Keras and TensorFlow. The author achieves a notable level of accuracy by utilizing GPU resources and processing time that are made freely available.

In conjunction with two stages of language models, Flor de Sousa Neto, Byron Leite Dantas Bezerra, Alejandro Hector Toselli, and Estanislau Baptista Lima [16] introduced a novel Gated-CNN-BGRU architecture for offline Handwritten Text Recognition systems. For optical models under five well-known public datasets in the HTR field (Bentham, IAM, RIMES, Saint Gall, and Washington), the author employed the same methodology. A high learning rate, low tolerance for early halting with reduction on plateau, and an emphasis on achieving the best outcome at the lowest cost characterize the streamlined architecture with few trainable parameters.

Using smaller training data sets, Jose Carlos Aradillas, Juan Jose Murillo-Fuentes, and Pablo M. Olmos [18] address the offline handwritten text recognition (HTR) challenge. The author employed deep learning neural networks, which are made up of long short-term memory recurrent units (LSTM) and convolutional neural networks (CNN). Furthermore, the labeling effort is substantially facilitated by the use of connectionist temporal classification (CTC), which prevents segmentation at the character level. After the CRNN is trained from the larger database, transfer learning (TL) is used to retrain all of the parameters using smaller datasets based on the parameters learnt from the larger database. After retraining and initializing the entire network, the TL technique performed well.

Deep learning is used by Bhargav Rajyagor and Rajnish Rakhlia [19] to compare handwritten character recognition. Among the deep learning networks that the author compares and contrasts for handwritten character recognition are supervised layer-by-layer training of a deep convolution neural network, Artificial Neural Network (ANN), Convolution Neural Network (CNN), Deep Neural Network (DNNs), DenseNet, K-NN classifier, and Deep Convolution Neural Network.

The study by Zuo Huahong, Tang Junyi, and Han Ping [20] emphasizes the use of an improved faster RCNN for adhesive handwritten digitizer recognition. Firstly, the NIST19 dataset is used as the main dataset, and a mixed dataset is constructed by establishing different hand-to-hand ratios with varied degrees of overlap, and then randomly add salt and pepper noise and Gaussian noise in the experimental photos. Second, using the aforementioned data sets, a model based on an enhanced Faster RCNN network is constructed and trained to address the issue of numerous overlapping objects in handwritten digital images. Lastly, an assessment is made of the model's average accuracy. The experimental findings demonstrate that the Faster RCNN model has a decent average detection accuracy.

3. Methodology

3.1 Block Diagram

The block diagram of the proposed system of consist of five major parts: Data Collection from IAM data sets and preprocessing of the data, Feature extraction, Sequence modeling, Recognition and Model Evaluation. Data is collected from IAM data sets. Feature extraction is done using CNN model. Hand written text recognition is done using four separate models BLSTM, BGRU, TCN, TRANSFORMER. Finally Meta model is built separately on the top of four models and evaluation is performed.

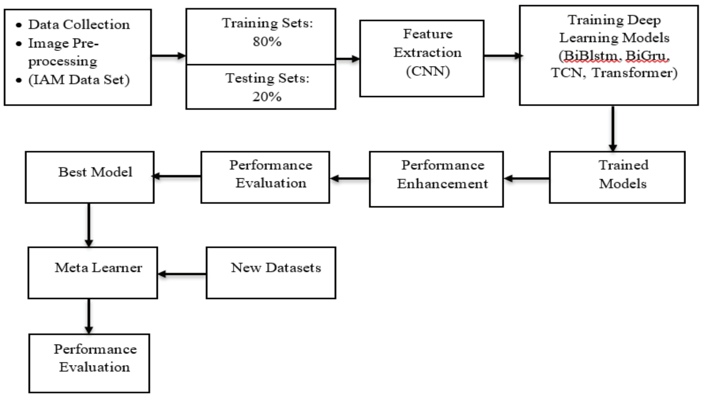


Figure 3.1 Block Diagram of Proposed System

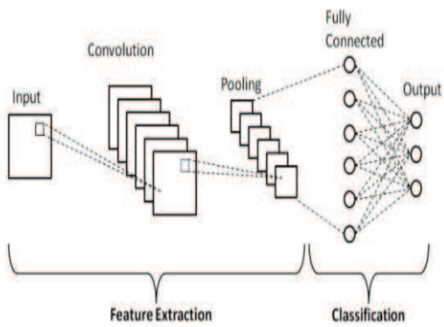


Figure 3.2: Basic CNN Block Diagram [22].

3.2 Reference Models

3.2.1 Bidirectional LSTM Model

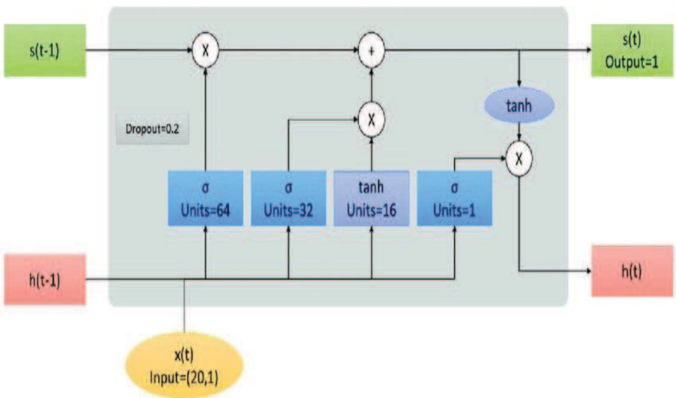


Figure 3.3: LSTM Architecture. [23]

3.2.2 Bidirectional GRU Model

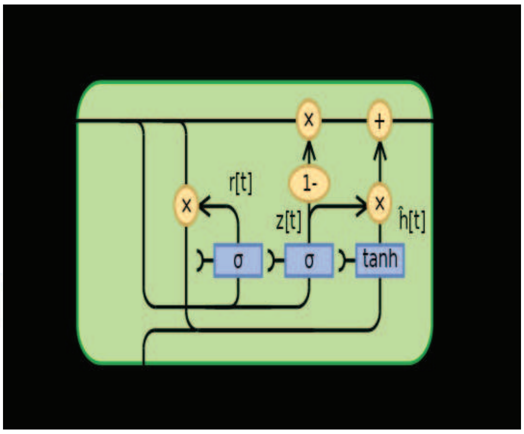


Figure 3.4: GRU Architecture [24]

3.3.3 Temporal Convolution Neural Network

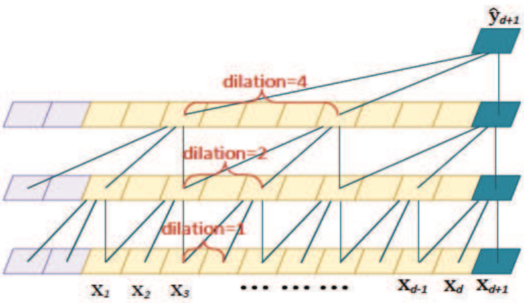


Figure 3.5: LSTM Architecture. [23]

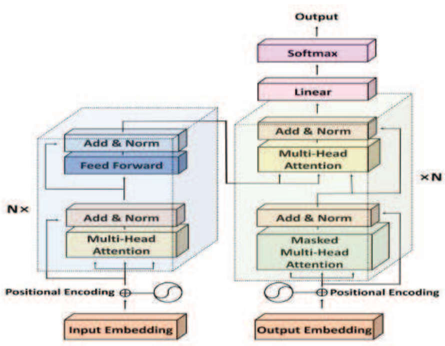


Figure 3.6 Architecture of Transformer Model [32]

3.3 Model optimization using meta learner

All of the data sets are split into three subsets that contain images from different classes and they are referred as training, validation and testing data. The adaptation process of the meta-learner BiGRU consists of 3 different stages: meta-training, meta-validation and meta-testing. We start with the meta-raining stage and train the meta-learner. The learner applies a gradient descent on the episode-training set using the rule determined by the meta-learner and then the meta-learner is evaluated according to the performance of the learner on the episode-evaluation set [28].

The BiGRU model function defines a GRU model using hyperparameters specified in the hp argument, which is expected to be HyperParameters object. The model architecture includes convolutional layers, pooling layers, batch normalization, and bidirectional GRU layers. The hyperparameters like the number of GRU layers, units, and dropout rates are tunable. The model is compiled with a choice of optimizer, learning rate, and the custom F1 metric.

Then After, the Meta Learner function is defined that defines a meta-learner model. This model takes hyperparameters as input features and outputs a predicted performance metric. The hyperparameters for the meta-learner include the choice of optimizer and learning rate.

The Update_Hyperparameters_Callback class is used which is a custom Keras callback. It updates the hyperparameters of the main model based on the predicted F1 score from the meta-learner. It uses the on_epoch_end method to get the current hyperparameters, predict F1 using the meta-learner, and update the hyperparameters.

Finally, tuner_meta is used which is a Random Search tuner from Keras-Tuner, configured to search for optimal hyperparameters for the meta-learner. It uses mean squared error as the objective to minimize.

Main_model is instantiated using the GRU_model function with hyperparameters obtained from the best trial of the meta-learner. The Update_Hyperparameters_Callback is added to the main model's training with callbacks. This callback will be called at the end of each epoch to update hyperparameters based on meta-learner predictions.

The main model is trained using the fit method, and the meta-learner callback is used during training to dynamically adjust hyperparameters.Meta model is trained with reduced datasets which consist of 61780 of images. Testing is done with 6150 data sets and validation is done with 6290 number of images.Results and Discussion

3.4 Recognition

Lastly, all base models are trained with the input data set with proper tuning of hyper parameter for handwritten text recognition with the IAM data set. After models are trained, models are used to recognize the test images containing handwritten text of IAM Data set.

3.5 Model Evaluation

To determine the quality and correctness of a classification model, following common evaluation metrics are computed.

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive} \qquad \text{Recall} = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

$$\text{F1 Score} = \frac{2 * Precision * Recall}{Precision + Recall}$$

4. Discussion and Conclusion

4.1 Handwritten Text Recognition Model

Table 4.1: Parameters for Base Model

S.N.	Parameters	Values
1.	Optimizer	SGD, RMSprop, Adam and Adamax
2.	Number of channels	1
3.	Input Image Size	128 X 32
4.	Number of Images	1,15,320
5.	Number of Epochs	100 (Early stopping with patience 10.
6.	Batch Size	32
7.	Learning Rate	0.01 to 0.0001
8.	Train:Valid:Test Ratio	8:1:1
9.	Tools	GPU-RTX 4090

4.2 Hyper -parameter tuning

4.2.1 BiLSTM Model

Table 4.2: Hyper parameters for BiLstm Model

S.N.	Parameters	Values
1.	Optimizer	SGD, RMSprop, Adam and Adamax
2	Learning Rate	0.01, 0.001, 0.0001
3	BiLstm Layers	1, 2, 3
4	Lstm Unit	32, 64, 128
5	Dropout rate	0.2, 0.3, 0.4, 0.5

4.2.2 BiGRU Model

Table 4.3: Hyper parameters for BiGru Model

S.N.	Parameters	Values
1.	Optimizer	SGD, RMSprop, Adam and Adamax
2	Learning Rate	0.01, 0.001, 0.0001
3	BiGru Layers	1, 2, 3
4	BiGru Unit	32, 64, 128
5	Dropout rate	0.2, 0.3, 0.4, 0.5

4.2.3 TCN Model

Table 4.2.5 Hyper parameters for TCN Model

S.N.	Parameters	Values
1.	Optimizer	SGD, RMSprop, Adam and Adamax
2	Learning Rate	0.01, 0.001, 0.0001
3	TCN Layers	1, 2, 3
4	No. of filters	32, 64, 128
5	Kernel size	2, 3, 4

4.2.4 Transformer Model

Table 4.5: Hyper parameters for Transformer

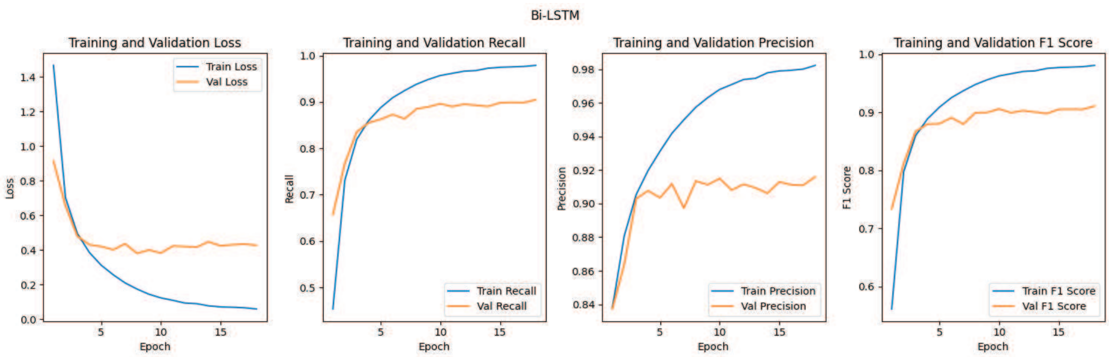
S.N.	Parameters	Values
1.	Optimizer	SGD, RMSprop, Adam and Adamax
2	Learning Rate	0.01, 0.001, 0.0001
3	Transformer Block	2, 4, 6, 8
4	Head size	8, 16, … , 256
5	No. of heads	2, 4, 6, 8, 12, 14, 10, 16

4.3 Best performing hyper parameters

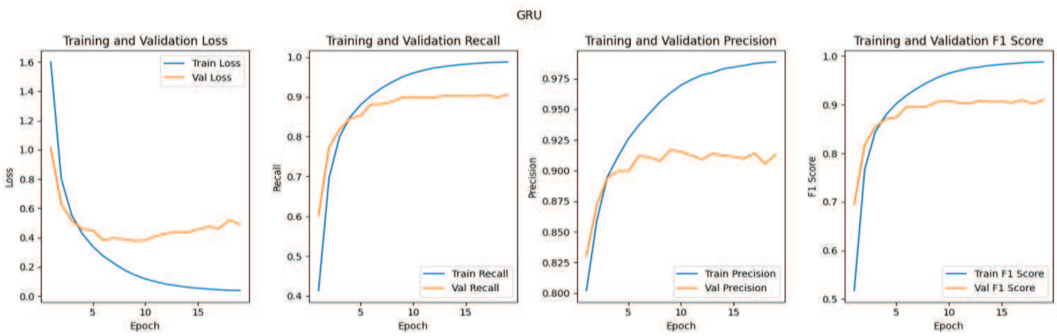
Table 4.6: Best performing Hyper parameters of Models

S. N	Model	Hyperparameter	Value
1	BiLstm	No. of Layers:	2
		BiLstm Unit:	128
		Optimizer:	Adam
		Learning rate:	0.0001
		Dropout Rate:	0.3
2	BiGru	No. of Layers:	2
		BiGru Unit:	128
		Optimizer:	Adamax
		Learning rate:	0.001
		Dropout Rate:	0.4
3	TCN	No. of Layers:	1
		No. of filters	128
		Kernel size	3
		Optimizer:	Adam
		Learning rate:	0.001
		Dropout Rate:	0.4
4	Transformer	No. of Transformer Block	6
		Head size	8
		No. of heads	10
		Optimizer:	Adamax
		Learning rate:	0.0001

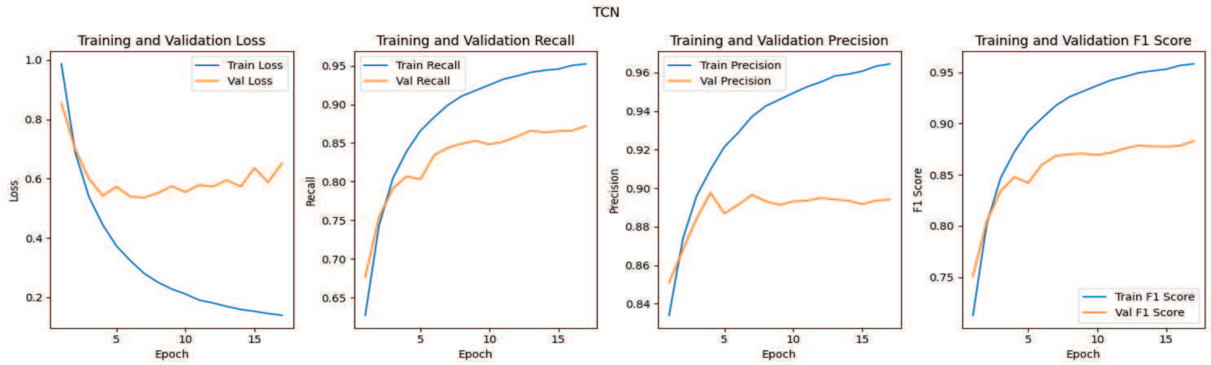
4.4 Results using BLSTM Model



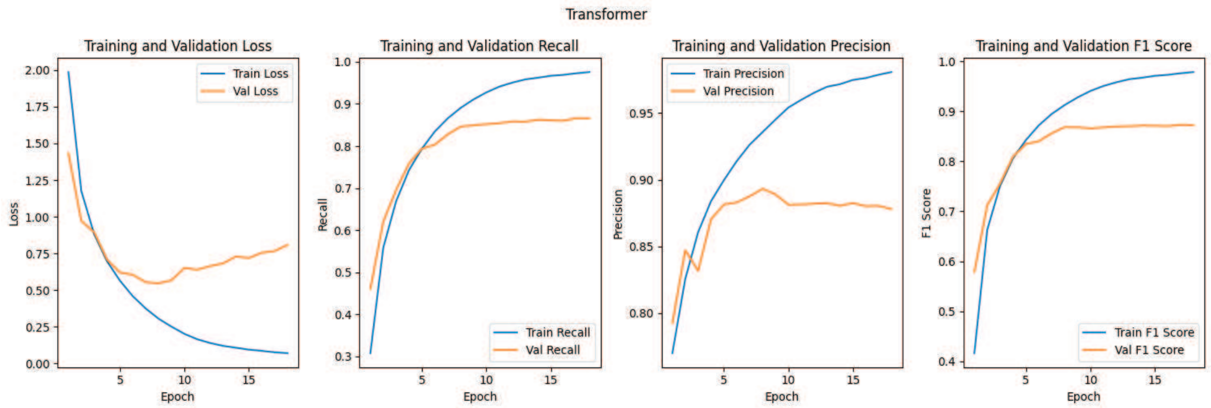
4.4 Results using GRU Model



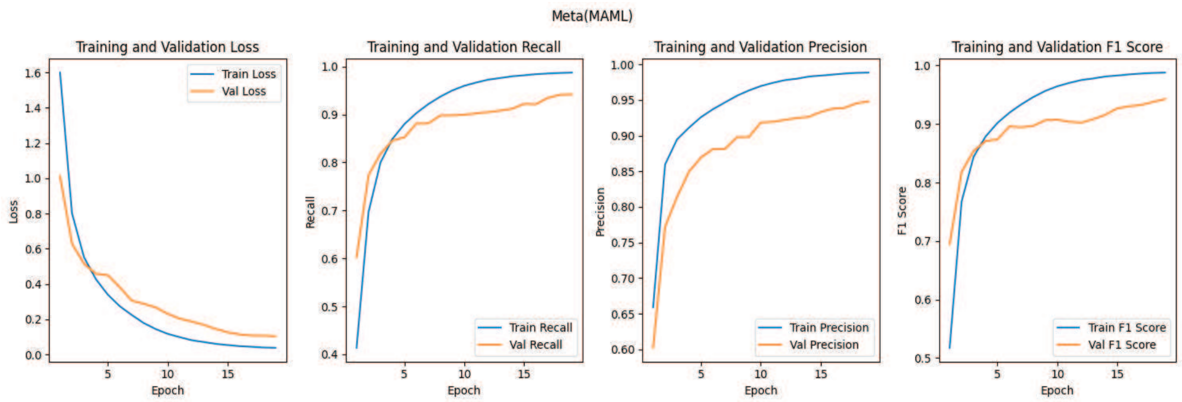
4.5 Results using TCN Model



4.6 Results using Transformer Model



4.7 Results using Meta Model



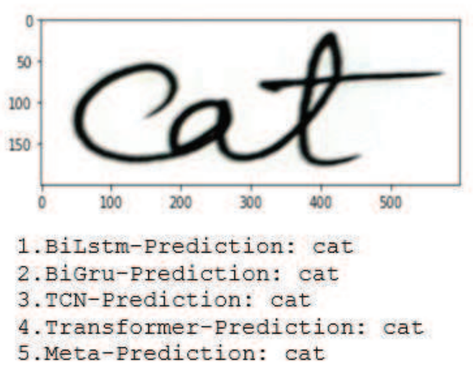
4.8 Summary Report of all Models

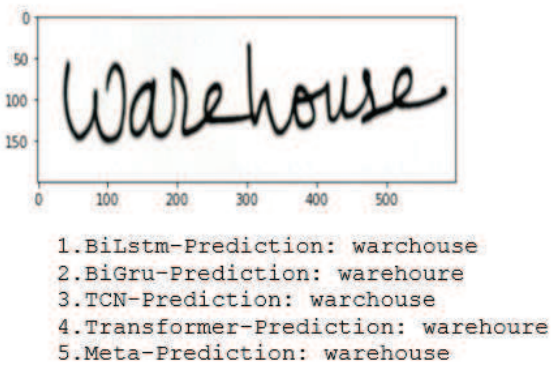
The summary of evaluation of different models is summarized below.

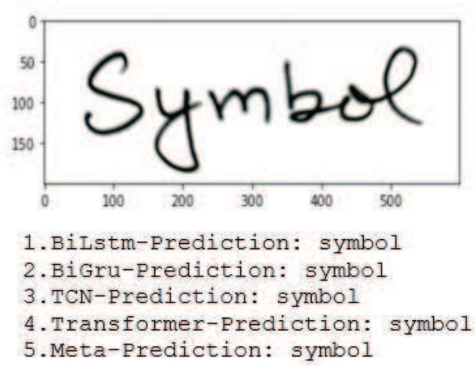
Table 4.12: Summary of all models Model

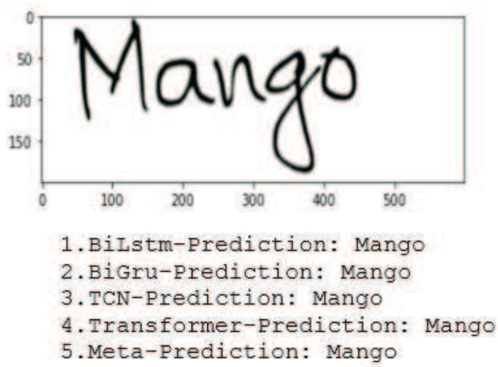
S.N.	Models	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)	Epochs
1.	BiLSTM	90.04	**91.62**	88.98	90.25	17
2.	BiGRU	**90.32**	91.56	**89.53**	**90.50**	18
3.	TCN	90.04	**91.62**	88.98	90.25	17
4.	Transformer	85.86	88.94	83.86	86.26	18
5.	Meta Model	**92.30**	**94.80**	**93.18**	**94.25**	15

4.9 Handwritten text samples









5.Conclusion and recommendation

IAM data sets of 1,15,320 images of handwritten text were used to recognize handwritten text. 78 characters in all, including the image "!"#&'()*+,./0123456789:;?ABCDEFGHIJKLMNO PQRSTUVWXYZabcdefghijklmnopqrstuvwxyz," make up the handwritten text. Following the acquisition of data sets, the sets underwent preprocessing, and CNN architecture is used to extract features from collected data sets. Following the extraction of features, four basic models—BiLstm, BiGru, TCN, and Transformer—were constructed, and their respective performances were assessed.

After that, the performance of the model was assessed and a meta model for the top performing model (Meta Bi-Gru) was built. Among the basic models, BiGRU achieved the highest accuracy and F1 score. The Transformer model had slightly lower performance compared to the recurrent models. The Meta Model based on BiGRU outperformed all individual models in terms of accuracy and F1 score. The choice of model architecture significantly impacts performance, with recurrent models (BiLSTM, BiGRU, TCN) generally performing well. The Transformer model, while still effective, showed comparatively lower accuracy and F1 score. The Meta Model, incorporating BiGRU, demonstrated superior performance, indicating the potential benefits of meta-learning approaches. The number of epochs varied across models, suggesting different convergence rates or training requirements.
Furthermore, meta models can be built for TCN and transformer models, which can further enhance the performance of the HTR.

6.Refrences

[1] 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)*, Nov. 2020.
[2] Ravina Mithe, N. Divekar, and S. Indalkar, "Optical Character Recognition," International Journal of Recent Technology and Engineering (IJRTE), vol. 2, no. 1, Mar. 2013.
[3] S. Das et al., "Hand-Written and Machine-Printed Text Classification in Architecture, Engineering & Construction Documents," Aug. 2018.
[4] N. Gupta and N. Goyal, "Machine Learning Tensor Flow Based Platform for Recognition of Hand Written Text," Jan. 2021, doi: https://doi.org/10.1109/iccci50826.2021.9402622.
[5] X.-Y. Zhang, Y. Bengio, and C.-L. Liu, "Online and offline handwritten Chinese character recognition: A comprehensive study and new benchmark," Pattern Recognition, vol. 61, pp. 348–360, Jan. 2017, doi: https://doi.org/10.1016/j.patcog.2016.08.005.
[6] S. Ahlawat, A. Choudhary, A. Nayyar, S. Singh, and B. Yoon, "Improved Handwritten Digit Recognition Using Convolutional Neural Networks (CNN)," Sensors, vol. 20, no. 12, p. 3344, Jun. 2020, doi: https://doi.org/10.3390/s20123344.
[7] L. Kumari, S. Singh, V. V. S. Rathore, and A. Sharma, "Lexicon and attention based handwritten text recognition system," Machine Graphics and Vision, vol. 31, no. 1/4, pp. 75–92, Dec. 2022, doi: https://doi.org/10.22630/mgv.2022.31.1.4.
[8] M. A. Ganaiea, P. N. Suganthanb, M. Tanveera, and M. Hub, "Ensemble deep learning: A review," Apr. 2021, doi: https://doi.org/10.48550/arXiv.2104.02395.
[9] W. Zhou, M. Lu, and R. Ji, "Meta-SE: A Meta-Learning Framework for Few-Shot Speech Enhancement," IEEE Access, vol. 9, pp. 46068–46078, Jan. 2021, doi: https://doi.org/10.1109/access.2021.3066609.
[10] S. Duth P. and B. Amulya, "Recognition of Hand written and Printed Text of Cursive Writing Utilizing Optical Character Recognition," 2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS), May 2020, doi: https://doi.org/10.1109/iciccs48265.2020.9121080.
[11] R. Parthiban, R. Ezhilarasi, and D. Saravanan, "Optical Character Recognition for English Handwritten Text Using Recurrent Neural Network," 2020 International Conference on System, Computation, Automation and Networking (ICSCAN), Jul. 2020.
[12] J. Bai, Z. Chen, B. Feng, and B. Xu, "Image character recognition using deep convolutional neural network learned from different languages," Oct. 2014, doi: https://doi.org/10.1109/icip.2014.7025518.
[13] M. Sonkusare and N. Sahu, "A Survey on Handwritten Character Recognition (HCR) Techniques for English Alphabets," Advances in Vision Computing: An International Journal, vol. 3, no. 1, pp. 1–

12, Mar. 2016, doi: https://doi.org/10.5121/avc.2016.3101.
[14] P. Nayak and S. Chandwani, “Improved Offline Optical Handwritten Character Recognition: A Comprehensive Review using Tensorflow,” International Journal of Engineering Research & Technology (IJERT), vol. 10, no. 11, Nov. 2021.
[15] K. Sharma, S. Kona, Anshul Jangwal, M Aarthy, Prayline Rajabai C, and Deepika Rani Sona, “Handwritten Digits and Optical Characters Recognition,” International Journal on Recent and Innovation Trends in Computing and Communication, vol. 11, no. 4, pp. 20–24, May 2023, doi: https://doi.org/10.17762/ijritcc.v11i4.6376.
[16] A. F. de Sousa Neto, B. L. D. Bezerra, A. H. Toselli, and E. B. Lima, “HTR-Flor: A Deep Learning System for Offline Handwritten Text Recognition,” 2020 33rd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), Nov. 2020, doi: https://doi.org/10.1109/sibgrapi51738.2020.00016.
[17] L. B. Kara and T. F. Stahovich, “An image-based, trainable symbol recognizer for hand-drawn sketches,” Computers & Graphics, vol. 29, no. 4, pp. 501–517, Aug. 2005, doi: https://doi.org/10.1016/j.cag.2005.05.004.
[18] J. C. Aradillas Jaramillo, J. J. Murillo-Fuentes, and P. M. Olmos, “Boosting Handwriting Text Recognition in Small Databases with Transfer Learning,” IEEE Xplore, Aug. 01, 2018. https://ieeexplore.ieee.org/document/8583799 (accessed Apr. 08, 2023).
[19] “Handwritten Character Recognition using Deep Learning,” International Journal of Recent Technology and Engineering, vol. 8, no. 6, pp. 5815–5819, Mar. 2020, doi: https://doi.org/10.35940/ijrte.f8608.038620.
[20] Zuo Huahong, J. Tang, and P. Han, “A New Type Method of Adhesive Handwritten Digit Recognition Based on Improved Faster RCNN,” Oct. 2020, doi: https://doi.org/10.1109/icsip49896.2020.9339270.
[21] U.-V. . Marti and H. Bunke, “The IAM-database: an English sentence database for offline handwriting recognition,” International Journal on Document Analysis and Recognition, vol. 5, no. 1, pp. 39–46, Nov. 2002, doi: https://doi.org/10.1007/s100320200071.
[22] Reymond Mesuga and Brian James Bayanay, “A Deep Transfer Learning Approach on Identifying Glitch Wave-form in Gravitational Wave Data,” arXiv (Cornell University), Jul. 2021, doi: https://doi.org/10.48550/arxiv.2107.01863.
[23] Z.-H. Wang, G.-J. Horng, T.-H. Hsu, A. Aripriharta, and G.-J. Jong, “Heart sound signal recovery based on time series signal prediction using a recurrent neural network in the long short-term memory model,” The Journal of Supercomputing, vol. 76, no. 11, pp. 8373–8390, Dec. 2019, doi: https://doi.org/10.1007/s11227-019-03096-x.
[24] H. Saleh, S. Mostafa, L. A. Gabralla, A. O. Aseeri, and S. El-Sappagh, “Enhanced Arabic Sentiment Analysis Using a Novel Stacking Ensemble of Hybrid and Deep Learning Models,” Applied Sciences, vol. 12, no. 18, p. 8967, Sep. 2022, doi: https://doi.org/10.3390/app12188967.
[25] N. Prabakaran, R. Kannadasan, A. Krishnamoorthy, and Vijay Kakani, “A Bidirectional LSTM approach for written script auto evaluation using keywords-based pattern matching,” Natural Language Processing Journal, vol. 5, pp. 100033–100033, Dec. 2023, doi: https://doi.org/10.1016/j.nlp.2023.100033.
[26] S. Furrer, Y. E. Erginbas, and M. Kayaalp, “Meta-Learner LSTM for Few Shots Learning,” 2017.
[27] Fırat Kızılırmak and Berrin Yanıkoğlu, “CNN-BiLSTM model for English Handwriting Recognition: Comprehensive Evaluation on the IAM Dataset,” Research Square (Research Square), Nov. 2022, doi: https://doi.org/10.21203/rs.3.rs-2274499/v1.
[28] S. Ravi and H. Larochelle, “Optimization as a Model for Few-Shot Learning,” International

Conference on Learning Representations, Apr. 2017.
[29] S. Bai, J. Zico Kolter, and V. Koltun, “An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling,” *arXiv (Cornell University)*, Mar. 2018.
[30] L. Yang, I. Koprinska, and M. Rana, “Temporal Convolutional Attention Neural Networks for Time Series Forecasting,” Jul. 2021,
doi: https://doi.org/10.1109/ijcnn52387.2021.9534351.
[31] Y. Lin, I. Koprinska, and M. Rana, “Temporal Convolutional Neural Networks for Solar Power Forecasting,” Jul. 2020, doi: https://doi.org/10.1109/ijcnn48605.2020.9206991.
[32] S. R. Choi and M. Lee, “Transformer Architecture and Attention Mechanisms in Genome Data Analysis: A Comprehensive Review,” Biology, vol. 12, no. 7, p. 1033, Jul. 2023, doi: https://doi.org/10.3390/biology12071033.