E-ISSN 2581 – 7957
P-ISSN 2277 – 3916
CVR Journal of Science and Technology, Volume 26, June 2024
DOI: 10.32377/cvrjst2617

A Study on Handwritten Text Recognition Classification using Diverse Deep Learning Techniques and Computation of CTC Loss

Ratnam Dodda[1], S Balakrishna Reddy[2], Azmera Chandu Naik[3], Venugopal Gaddam[4]
[1]Sr. Assistant Professor, CVR College of Engineering / Dept. of CSE(AI&ML), Hyderabad, India.
Email: ratnam.dodda@gmail.com
[2]Assistant professor, CVR College of Engineering/ Department of CSE(DS), Hyderabad, India.
Email: sama.balakrishnareddy@gmail.com
[3]Sr. Assistant Professor, CVR College of Engineering/ Dept. of CSE(AI&ML), Hyderabad, India.
Email: azmerachandunaik@cvr.ac.in
[4]Associate Professor, KL University/ Dept. of CSE, Guntur, India.
Email: venugopal.gaddam@gmail.com

Abstract: **Handwriting recognition encompasses the conversion of hand-written text images into digital text, where input images yield predicted textual output. Optical Character Recognition (OCR) technology has conventionally fulfilled this role. With surging mobile phone usage, leveraging text detection via mobile cameras gains significance in fields like medical script processing and exam evaluation. To enhance image quality, noise reduction techniques like binarization and thresholding are applied. Image processing entails letter segmentation and extraction. In this paper, we propose a neural network classifier model amalgamating Convolution Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks. Leveraging an existing feature dataset, the model is trained. Input images are evaluated through the neural network, yielding recognized words in a text document format. Ultimately, the predicted output is derived via Connectionist Temporal Classification (CTC)- based loss computation.**

Keywords: **Handwritten Recognition, Deep Learning Techniques, Optical Character Recognition.**

I. Introduction

Handwriting recognition is a valuable technique used to convert handwritten text, captured as images from mobile phones, into machine-readable text. While Optical Character Recognition (OCR) scanners are typically employed for this purpose, the prevalence of mobile phones has expanded the scope of text detection from mobile cameras [1]. This has led to numerous practical applications, including medical script processing and exam script evaluation. However, mobile camera images tend to be noisier compared to OCR-scanned images. To mitigate this image processing techniques like binarization and thresholding are applied for noise reduction. Subsequently, the handwritten characters are segmented and isolated from the image. Features, such as binary codes, are extracted from these characters [2]. To classify these characters, a neural network classifier, often built on a Long Short-Term Memory (LSTM) network, is trained using a character dataset. This neural network is then utilized to analyze input images and produce a text document containing the recognized words [3]. Given that input sources are image-based, noise levels are typically higher than those found in systems using scanned images. Consequently, noise reduction methods, such as low-intensity pixel removal, are employed to enhance the efficiency of the process [4].

A. Recurrent neural networks (RNN)

Recurrent neural networks (RNNs) belong to a category of artificial neural networks specifically designed for handling sequential data, making them well-suited for tasks that rely on memory and sequential dependencies. They find extensive applications in various domains such as speech recognition, language translation, natural language processing (NLP), and more. RNNs are categorized as supervised learning algorithms [5].

In the architecture of an RNN, a crucial feature is its ability to preserve memory. This is achieved by feeding the output of the current time step back into the network as input for the subsequent time step. This iterative process can be repeated as needed, making it adaptable to problems of varying complexity. Once the output for the final time step is computed, it is compared to the actual target output. Any discrepancies, or errors, between the predicted and actual values are then propagated backward through the network. This back-propagation process is instrumental in training the RNN, enabling it to learn and improve its performance over time [6].

B. Convolutional Neural Networks (CNN)

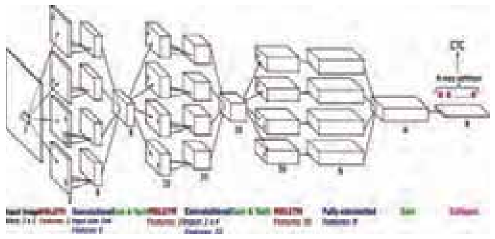


Figure 1. Convolutional Neural Network Framework

The proposed design is as follows, the input image is taken as a 2x2 block at once and given as an input to the first LSTM and convolution layer where the activation functions used are tanh. Finally, before the output is predicted a soft

Received on 03-02-2024, Revised on 04-03-2024, Accepted on 03-04-2024.

max activation function is applied [7]. Convolutional neural networks are a class of artificial neural networks utilized primarily for pixel processing and image recognition. The architecture consists of three layers, firstly a convolutional layer followed by a pooling layer, and finally a Fully Connected (FC) layer, with increasing complexity [8]. The convolutional layer serves as the fundamental building block within a Convolutional Neural Network (CNN), responsible for the bulk of data processing and calculations. Depending on the complexity of the task at hand, multiple convolutional layers can be stacked on top of one another. In this layer, a pivotal operation known as convolution takes place. Here, a kernel or filter traverses the receptive field of the input image, systematically inspecting for the presence of features. Through multiple iterations, the kernel traverses the entire image, calculating the dot product between the input pixels and the filter during each pass. The cumulative result of these dot products yields a feature map or convolved feature, which encapsulates important spatial information [8]. Importantly, the image is transformed into numerical values within this layer, enabling the CNN to interpret the image and discern relevant patterns. Following the convolutional layers, the pooling layer comes into play, facilitating dimensionality reduction. While this process trims down data, it comes at the cost of some loss of information. The pooling layer can perform two distinct types of pooling, namely max pooling and average pooling [9]. The final piece of the puzzle resides in the Fully Connected (FC) layer, where image classification takes center stage based on the features extracted in the preceding layers. The term "fully connected" implies that every input or node from one layer is linked to each activation unit or node in the subsequent layer. However, it's worth noting that not all layers within a CNN are fully connected. This strategic design choice helps prevent the network from becoming unnecessarily dense, reducing potential losses, maintaining output quality, and ensuring computational efficiency [10].

II. Methodology

The handwriting recognition model proposed in this paper takes input in the form of an image and then converts it into digital text. This model consists of a CNN-LSTM architecture. The loss used in the end is the CTC (Connectionist Temporal Classification) loss [11].

Importing Data and Dependencies We import the data using the Pandas Data frame. The data needed is only the images of the words and the word.txt file. We then place the downloaded files inside the data directory [12].

Data Pre-processing and preparing the images for training the input images are initially loaded in grayscale format and then undergo a series of transformations. First, they are reshaped to adhere to a standardized size with a width of 128

pixels and a height of 32 pixels. If the original image dimensions exceed these specifications (i.e., the width is greater than 128 or the height is greater than 32), cropping is applied to adjust the image to the required size while retaining the most significant content. On the other hand, iftheoriginalimagedimensionsaresmallerthan128pixelsinwidt hor32pixels in height, the image is padded with white pixels to reach the prescribed dimensions. Following these adjustments, the image is rotated in a clockwise direction to achieve the desired shape, typically specified as (x, y). Lastly, to facilitate consistent processing, the image is normalized, scaling its pixel values to fall within the range of [0, 1]. This normalization simplifies subsequent computations and ensures uniformity across all input images [13].

A.Label Encoding for CTC Loss

The primary objective function employed for minimizing the loss is the Connectionist Temporal Categorical loss function, commonly referred to as CTC. In contrast to other loss functions that optimize a single objective, the CTC loss uniquely addresses a dual optimization task. It is specifically designed to optimize not only the accurate prediction of class labels within a sequence but also the optimal alignment and length of the predicted sequence. This dual optimization is particularly advantageous when dealing with input images of varying nature [14]. To facilitate the training process, the image labels in the data set need to be converted into numerical representations that correspond to each character found in the training set. The character set, often referred to as the "Alphabet," encompasses the letters A to Z and includes three special characters: hyphen ('-'), apostrophe ('), and space. This transformation is essential for enabling the network to learn and make predictions effectively during training and inference [15].

B.Model Building

In our architectural design, tailored for input images with dimensions of 32 pixels, 32 pixels in height, and 128 pixels in width, we've in incorporated a sequence of seven convolutional layers. Among these, six employ a kernel size of (3,3), while the final one employs a kernel size of (2,2). We progressively increase the number of filters in a layer-by-layer fashion, ranging from 64 to 512 filters. To capture and distill relevant features, we intersperse max-pooling layers with a size of (2,2). Additionally, we introduce two max-pooling layers with a size of (2,1) to focus on extracting features with larger widths, which is particularly advantageous for accurately predicting long text sequences [16]. To expedite the training process and enhance stability, we incorporate batch normalization layers after the completion of the fifth and sixth convolutional layers. To ensure seamless compatibility with the subsequent LSTM layer, we employ a lambda function to preprocess the output from the convolutional layers. Following this, our network utilizes two Bidirectional LSTM layers, each comprising 128 units. These RNN layers yield an output with dimensions (batch size,31,79), where 79 represents the total number of output classes, encompassing both characters and blank spaces [17].

TABLE1.
MODEL DESCRIPTION

Layer(type)	Output shape	Param
input1(Input Layer)	(None,32,128,1)	0
conv2d(Conv2D)	(None,32,128,64)	640
maxpooling2d	(None,16,64,64)	0
conv2d1(Conv2D)	(None,16,64,128)	73856
Max pooling 2d1(MaxPooling2D)	(None,8,32,128)	0
conv2d2(Conv2D)	(None,8,32,256)	295168
conv2d3(Conv2D	(None,8,32,256)	590080
Max pooling 2d2(MaxPooling2D)	(None,4,32,256)	0
conv2d4(Conv2D)	(None,4,32,512)	1180160
batch normalization	(None,4,32,512)	2048
conv2d5(Conv2D)	(None,4,32,512)	2359808
batch normalization1	(None,4,32,512)	2048
Max pooling 2d3(MaxPooling2D)	(None,2,32,512)	0
conv2d6(Conv2D)	(None,1,31,512)	1049088
Lambda (Lambda)	(None,31,512)	0
Bidirectional (Bidirectional)	(None,31,512)	1574912
bidirectional1(Bidirectional)	(None,31,512)	1574912
Dense (Dense)	(None,31,79)	40527

Defining Loss Function

$$L(\theta) = \frac{1}{N}\sum_{i=1}^{N} L(f(x_i;\theta), y_i)$$

- $L(\theta)$ represents the loss function.
- N represents the total number of samples.
- $f(x_i;\theta)$ represents the model prediction for the ith sample with parameters θ.
- y_i represents the true label for the i th sample.
- L represents the loss function applied to the predicted value and the true label.

In this step we use the Connectionist Temporal Categorical CTC Loss function which calculates the loss between a continuous series and a target sequence. Here we use the CTC loss function over other loss functions as the CTC loss can optimize the length of the sequence that is predicted as well as the classes of the predicted sequence [18].

C. Training Model

Here we use batch size 5 and run 25 epochs. We use the Stochastic Gradient Descent (SGD) optimizer which iteratively optimizes an objective function with appropriate smoothness properties. We use the SGD optimizer as it reduces excessive computational burden thereby achieving faster iterations for a lower convergence rate.

D. Decoding Outputs from Prediction

Performance evaluation is conducted using the Levante in distance metric. To assess the quality of recognition, we employ the Jaro-Winkler algorithm, which gauges the similarity between the text captured by our system and the ground truth or actual text. This algorithm is instrumental in quantifying the accuracy of our text recognition process. This is also allowing us to get a quantitative understanding of how good our model is.

E. Architecture

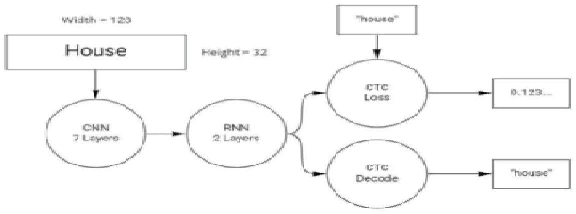
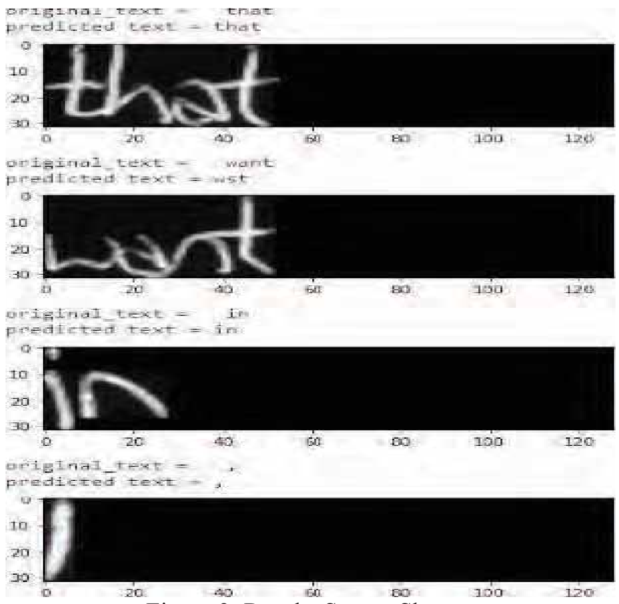


Figure 2. System Architecture

The system architecture represents the underlying architecture of the neural network used to represent images. First, the image (after preprocessing into 128x32) is passed into the 7 CNN layers where the feature extraction and transformation take place. After this step the 2 RNN or LSTM layers are used to predict the text from the images. The CTC decode is the method used to predict the text whereas the CTC Loss is the method used to calculate the loss of the predicted text. The technology used for the construction of this project is a combination of neural networks. First, we encounter convolutional neural networks (7 layers) which are used to process the input image. They change the dimensionality of the image while retaining the data to be able to pass this input to the next layer. Next, we use two bi-directional LSTM networks where the text on the image is recognized. Finally, the transcription layer translates the per-frame predictions into the final label sequence. For calculating the loss, we are using the CT Closes function which is very apt for handwriting recognition as it calculates the loss based on character scores [8].

III. RESULTS

The final model can identify English words with an accuracy of up to 63.1



```
original_text =   that
predicted text = that
original_text =   want
predicted text = wst
original_text =   in
predicted text = in
original_text =   ,
predicted text = ,
```

Figure 3. Results Screen Shots

As shown in the screenshot, our model can predict words and symbols. The word “that” is predicted as that, and “want” is predicted as wst - this inaccuracy can be attributed to the fact that “a” and “n” in the image are very closely written and the model is unable to distinguish the letter as it is not provided with any context, “in” is predicted a sin, and the “,” symbol is predicted as, Learning curves plot the loss of training and validation of a sample of training examples by adding new training examples incrementally. Learning curves help us determine whether adding more training examples would improve the validation score (score on unseen data). If the model is overfitting, adding more training examples can improve the model’s performance on unseen data. Similarly, if the model is undersized, adding training examples will not help. The first learning curve is the loss graph that is plotted along each epoch or cycle for both the training and validation data. It is observed that as the epoch increases, over time the loss decreases. This Cond. learning curve is the accuracy graph that is plotted along each epoch or cycle for both the training and validation data. It is observed that as the epoch increases, over time the accuracy increases and eventually flattens out. So, in our model with each epoch, the loss decreases and accuracy increases.

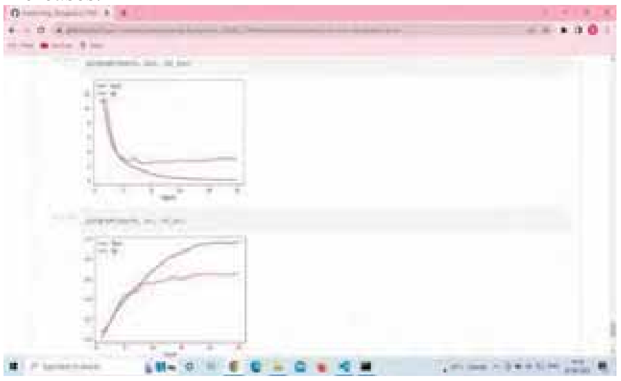


Figure 4. Loss and Accuracy Graphs

IV. Conclusion

The current model demonstrates proficient prediction capabilities for words, numbers, and symbols. However, the potential for enhancing its capabilities lies in the incorporation of line segmentation. By integrating line segmentation, the model can be expanded to encompass complete paragraph text recognition. This advancement opens doors to achieving efficient recognition of handwriting across various languages, transcending the confines of English. Modifying and extending the existing CRNN+LSTM+CTC architecture serves as the foundation for creating a versatile system capable of both segmenting and recognizing handwritten text, thereby facilitating its widespread applicability.

References

[1] B. Balci, D. Saadati, and D. Shiferaw, “Handwritten text recognition using deep learning,” CS231n: Convolutional Neural Networks for Visual Recognition, Stan- ford University, Course Project Report, Spring, pp. 752–759, 2017.

[2] C. C. Tappert, C. Y. Suen, and T. Wakahara, “The state of the art in online handwriting recognition,” IEEE Transactions on pattern analysis and machine intelligence,vol.12,no.8,pp.787–808,1990.

[3] J.Brownlee,“Sequence classification with lstm recurrent neural networks in python with keras,” Deep Learning for Natural Language Processing, Machine Learning Mastery, 2016.

[4] L. Malinski, K. Radlak, and B. Smolka, “Is large improvement in the efficiency of impulsive noise removal in color images till possible?,”Plosone,vol. 16,no.6, p.e0253117,2021.

[5] R. DiPietro and G. D. Hager, “Deep learning: Rnns and lstm,” in Handbook of medical image computing and computer assisted Intervention, pp. 503–519, Elsevier, 2020. Sutskever, Training recurrent neural networks. University of Toronto Toronto, ON, Canada, 2013.

[6] S.Iqbal,A.N.Qureshi,J.Li,andT.Mahmood, “On the analyses of medical images using traditional machine learning techniques and convolutional neural networks,” Archives of Computational Methods in Engineering,vol.30,no.5,pp.3173–3233, 2023.

[7] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, “Convolutional neural networks:an overview and application in radiology,”Insights into imaging, vol. 9, pp.611–629,2018.

[8] J.Brownlee,“Agentleintroductiontopoolinglayersforco nvolutionalneuralnet- works,”Machine Learning Mastery,vol.22,2019.

[9] S. Patel and A. Patel, “Object detection with convolutional neural networks,” Machine Learning for Predictive Analysis: Proceedings of ICTIS 2020, pp. 529– 539, 2021.

[10] Abdallah, M. Hamada, and D. Nurseitov, “Attention-based fully gated cnn-bru for Russian handwritten text,” Journal of Imaging, vol. 6, no. 12, p. 141, 2020.

[11] S. Khalid, S. Wu, A. Alam, and I. Ullah, “Real-time feedback query expansion technique for supporting scholarly search using citation network analysis,” Journal of Information Science, vol. 47, no. 1, pp. 3–15, 2021.

[12] V. V. Mainkar, J. A. Katkar, A. B. Upade, and P. R. Pednekar, “Handwritten character recognition Conference on Electronics and Sustainable Communication Systems (ICESC), pp. 599–602, IEEE, 2020.

[13] Kim, S., Hori, T., & Watanabe, S. (2017). “Joint CTC-attention based end-to-end speech recognition using multi-task learning”. In 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 4835-4839).

[14] N.H.Feldman, T.L. Griffiths, S . Goldwater, and J.L.Morgan,“ Role for the developing lexicon in phonetic category acquisition., ”Psychological Review, vol. 120, no. 4, p. 751, 2013.

[15] R. Ramya, A. Anandh, K. Muthulakshmi, and S. Venkatesh, "Gender recognition from facial images using multi-channel deep learning framework," in Machine Learning for Biometrics, pp. 105–128, Elsevier, 2022.

[16] Xie, S., Girshick, R., Dollár, P., Tu, Z., & He, K. (2017). "Aggregated Residual Transformations for Deep Neural Networks". In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)

[17] W. W. Cohen, P. Ravikumar, S. E. Fienberg, et al., "A comparison of string distance metrics for name-matching tasks.," in IIWeb, vol. 3, pp. 73–78, 2003.

[18] M. Soh, "Learning cnn- lstm architectures for image caption generation," Dept. Comput. Sci., Stanford Univ., Stanford, CA, USA, Tech.Rep, vol.1,2016.