Handwriting Recognition: A Deep Learning Approach

Adithya, Architha Rajasekar, Bhuvaneswaran, Harshith Reddy
Department of Artificial intelligence
Amrita vishwa vidyapeetam
Coimbatore, India
cb.sc.u4aie24009@cb.students.amrita.edu

***Abstract*—Handwriting recognition is the task of automatically reading and interpreting handwritten text. Although deep learning has brought significant improvements in this area, offline handwriting recognition—working with static images—still presents many challenges. This paper describes a deep learning model that combines Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), specifically Long Short-Term Memory (LSTM) units, with a special loss function called Connectionist Temporal Classification (CTC). We applied this model to a dataset of handwritten names and evaluated its performance. This paper explains how the system was built, how it was trained and tested, and how each part of the network works. We also discuss difficulties encountered and how the model can be improved in the future.**

***Index Terms*—Handwriting recognition, CNN, LSTM, CTC, deep learning, sequence modeling**

I. Introduction

Handwriting recognition allows computers to read human writing. It is useful in many real-world applications, such as reading scanned forms, sorting mail, or digitizing historical documents.

There are two main types of handwriting recognition:

- **Online**: The system receives data in real time, such as from a stylus or touchscreen. It has access to the writing order and speed.
- **Offline**: The system only sees a static image of the writing, like a photo or scan of paper.

This work focuses on offline handwriting recognition, which is more difficult because we do not have timing information. The main idea is to use deep learning to process an image of handwriting and convert it into text. We use a combination of two types of neural networks: CNNs, which are good at processing images, and RNNs, which are good at understanding sequences.

II. Literature Review

Several significant works have influenced the development of handwriting recognition systems using deep learning:

- **Connectionist Temporal Classification (CTC)** was introduced by Graves et al. [1]. CTC enables training of RNNs for sequence labeling tasks without requiring pre-segmented data. This technique is particularly useful for handwriting and speech recognition.
- **Shi et al.** [2] proposed an end-to-end neural network for image-based sequence recognition, combining CNNs with RNNs and CTC loss. Their architecture demonstrated strong performance in scene text recognition and provided the foundation for many modern handwriting recognition systems.
- **Bluche et al.** [3] extended these ideas by incorporating attention mechanisms with multidimensional LSTM networks, allowing the model to selectively focus on different parts of an input image while decoding text. Their model showed improved results on paragraph-level handwriting recognition.

These works collectively show the effectiveness of deep architectures and sequence modeling techniques in recognizing handwritten or scene text. Our work builds upon these ideas by implementing a CNN-BiLSTM-CTC pipeline for handwritten name recognition.

III. Dataset and Preprocessing

We used a dataset of images, each containing a handwritten name along with its correct text label. Before training the model, we cleaned and prepared the data in the following steps:

1) **Grayscale conversion**: Converted all images to grayscale, removing color and reducing size.
2) **Resizing and normalization**: Resized images to 256 pixels wide and 64 pixels high. Pixel values were normalized to be between 0 and 1.
3) **Label formatting**: Converted all text labels to uppercase and removed any special characters.
4) **Cleaning**: Removed images that were too unclear or had missing labels.

Fig. 1: Example handwritten name images from the dataset

IV. Neural Network Architecture

The model is made up of several layers, each with a specific role:

A. 1. Convolutional Neural Network (CNN)

CNNs are used to extract useful features from the input image. Instead of processing every pixel individually, they find important patterns such as edges, shapes, and textures.

- We used 3 convolutional blocks.
- Each block includes:
  - A convolutional layer to detect features
  - A batch normalization layer to stabilize training
  - A max pooling layer to reduce image size and focus on important features
  - A dropout layer to prevent overfitting

B. 2. Recurrent Neural Network (RNN)

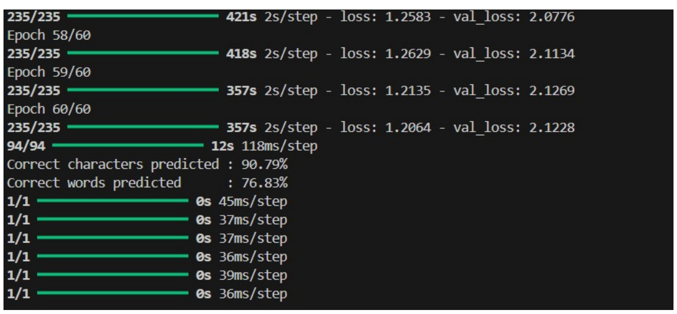
After extracting features with the CNN, we reshape the output so that it can be processed as a sequence.

- We use two layers of Bidirectional LSTM.
- LSTMs help the network remember information over time.
- Bidirectional LSTMs read the sequence from both left-to-right and right-to-left, giving more context.

C. 3. Output Layer and CTC Loss

The final dense layer maps each time step in the sequence to a probability distribution over all characters (A-Z, etc.).

We use Connectionist Temporal Classification (CTC) loss to train the model. CTC allows the model to learn alignments between input images and output text without manually marking where each letter appears in the image.



```
235/235 ━━━━━━━━━━━━━━━━━━━━ 421s 2s/step - loss: 1.2583 - val_loss: 2.0776
Epoch 58/60
235/235 ━━━━━━━━━━━━━━━━━━━━ 418s 2s/step - loss: 1.2629 - val_loss: 2.1134
Epoch 59/60
235/235 ━━━━━━━━━━━━━━━━━━━━ 357s 2s/step - loss: 1.2135 - val_loss: 2.1269
Epoch 60/60
235/235 ━━━━━━━━━━━━━━━━━━━━ 357s 2s/step - loss: 1.2064 - val_loss: 2.1228
94/94 ━━━━━━━━━━━━━━━━━━━━ 12s 118ms/step
Correct characters predicted : 90.79%
Correct words predicted      : 76.83%
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 37ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 37ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 36ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 39ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 36ms/step
```

Fig. 2: Diagram of the accuracy

V. Training, Validation, and Testing

A. Training Process

We split the dataset into three parts:

- **Training set (30,000)**: Used to teach the model.
- **Validation set (3000)**: Used to monitor model performance during training.
- **Test set** : Used only at the end to evaluate final accuracy.

Training settings:

- Optimizer: Adam (efficient for deep learning)
- Learning rate: 0.0001
- Batch size: 128 images at a time
- Epochs: 60 full passes over the training set
- Mixed precision and XLA enabled for faster training on GPU

B. Validation and Testing

During and after training, we measured:

- **Character-level accuracy**: Percentage of characters predicted correctly
- **Word-level accuracy**: Percentage of full names predicted without any error

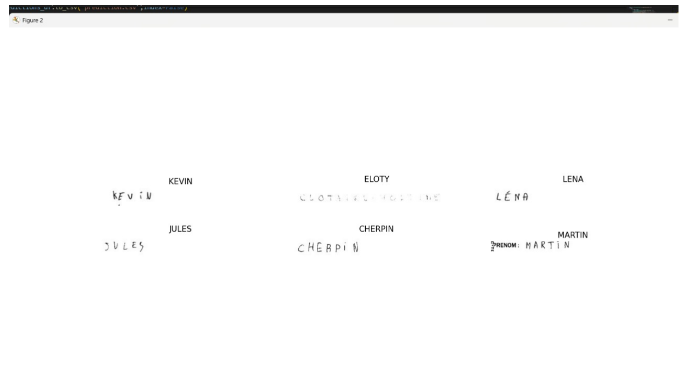


Fig. 3: Examples of correct predictions made by the model

VI. Results and Discussion

The model achieved the following results:

- Character accuracy: 90.79%
- Word accuracy: 76.83%

Key findings:

1) CNN-RNN architecture works well for handwriting.
2) Bidirectional LSTMs improve accuracy by using full context.
3) CTC loss avoids the need for manual alignment.
4) Good preprocessing makes a big difference.

VII. Challenges and Future Work

Despite the good performance, some issues remain:

- Different handwriting styles are hard to generalize
- The model works mainly with names and not longer text
- It requires a powerful computer to train

Ways to improve:

- Add attention mechanisms for better focus on characters
- Try transformer-based models
- Use data augmentation to simulate more handwriting styles
- Train with more diverse and larger datasets

VIII. Conclusion

We built a deep learning model that can read handwritten names from images using a combination of CNN and RNN layers with CTC loss. The system performs well and can be useful in many practical applications like form reading and document scanning. With further improvements, it can be adapted to work with longer texts and more complex handwriting.

REFERENCES

[1] Graves, A., et al. (2006). "Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks."
[2] Shi, B., et al. (2015). "An End-to-End Trainable Neural Network for Image-based Sequence Recognition and Its Application to Scene Text Recognition."
[3] Bluche, T., et al. (2017). "Scan, Attend and Read: End-to-End Handwritten Paragraph Recognition with MDLSTM Attention."