CNN and LSTM-Based Approaches for Signature Recognition and Word Completion

Awais Malik
dept. Computer Science
FAST National University of Computer and Emerging Sciences
Islamabad, Pakistan
muhamadawaismalik@gmail.com

***Abstract*—This report discusses two tasks: signature recognition using a Convolutional Neural Network (CNN) and sentence completion using a Long Short-Term Memory (LSTM) model. The CNN model was designed to classify handwritten signatures extracted from noisy images. Challenges with image segmentation and noise were encountered during preprocessing, which affected model accuracy. The LSTM model was trained on Shakespeare's plays to predict the next word in a sentence.**

***Index Terms*—CNN, LSTM, Signature Recognition, Word Prediction, Feature Extraction**

I. Introduction

This report covers two tasks: developing a CNN model for signature recognition and building an LSTM model for sentence completion. The aim was to classify handwritten signatures accurately and generate coherent sentence completions based on text input.

II. Methodology

Task 1: Signature Recognition using CNN
Dataset: The data consists of images containing multiple rows of handwritten signatures, each associated with a person ID.
Preprocessing: Grayscale conversion and Gaussian blurring were applied to the images to reduce noise. Each image was segmented into rows to extract the person IDs and corresponding signatures. Signature images were resized for CNN input, and labels were encoded for classification.
Model: A CNN model was constructed with two convolutional layers, max pooling, and dense layers, followed by a softmax classifier for recognizing the person IDs.
Task 2: Sentence Completion using LSTM
Dataset: The text dataset was derived from Shakespeare's plays, using sentences for word prediction.
Preprocessing: The text was tokenized, converted to lowercase, and padded to ensure uniform sequence lengths.
Model: A word-level LSTM model was created using embedding layers and bidirectional LSTMs to predict the next word in a sequence.

III. Results

Task 1: Signature Recognition
Performance: The CNN model's accuracy was limited due to noisy data during preprocessing. The inconsistency in the quality of the extracted signature images affected the model's ability to correctly classify signatures. Despite denoising and segmentation efforts, the model struggled with clarity and precision in the recognition task.
Evaluation: Overall, the CNN achieved relatively low accuracy, highlighting the need for improved preprocessing techniques to handle noisy images more effectively.

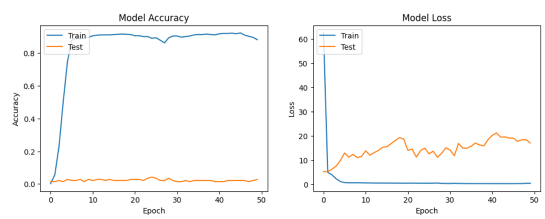

Fig. 1. CNN's accuracy and loss with 15 Epochs.

Task 2: Sentence Completion
Performance: The LSTM model improved over time with more training, successfully completing common sentence patterns. However, it encountered difficulties with more complex and rare word sequences, affecting its overall predictive accuracy. At the end of the training, the model achieved an accuracy of **0.2550** with a loss of **2.5036**. The model generated several phrases, including:

- I am a king that
- Find we a time for her
- No more the thirsty entrance her
- Nor more shall trenching war well
- Nor bruise her flowerets with that

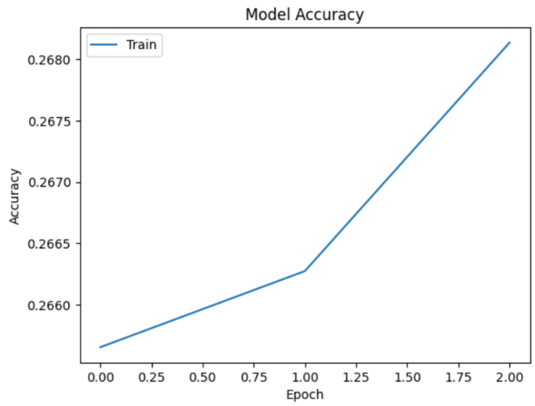

Fig. 2. Example of signature recognition results with 2 Epochs.

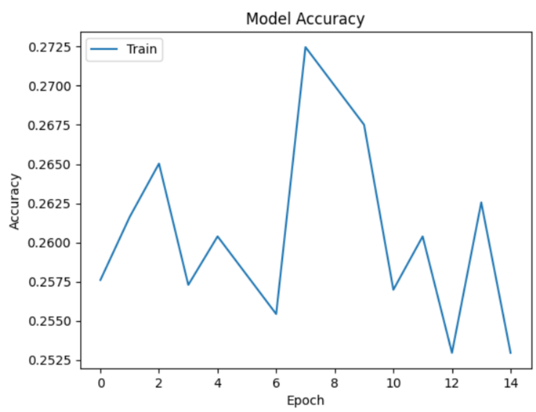

Fig. 3. Example of signature recognition results with 15 Epochs.

IV. Discussion

Signature Recognition

A major challenge in the signature recognition task was the preprocessing of data. Many of the images were noisy or unclear, making it difficult to extract clean, segmented signatures. This resulted in poor input quality for the CNN, which in turn led to lower accuracy. While basic denoising techniques were applied, further preprocessing steps, such as more advanced filtering or manual intervention, could help improve data quality and, consequently, model performance.

Sentence Completion

The LSTM model showed promise, especially with simpler sequences, but it struggled with rare word combinations. Although the model's accuracy improved with training, its handling of complex sentences remained inconsistent. This task could benefit from further hyperparameter tuning and longer training periods to enhance sentence coherence and word prediction accuracy.

V. Conclusion

Both the Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) models showed some improvement over time, but each faced its own challenges.

For the CNN model, the main issue was the noisy data, which made preprocessing difficult. This noise affected the clarity of the images, leading to struggles in accurately recognizing signatures. It became clear that effective data cleaning is essential; even small problems in the data can lead to significant drops in accuracy.

The LSTM model, on the other hand, had trouble predicting complex word sequences. Even though LSTMs are good at remembering long-term information, the unique style and structure of Shakespeare's writing made it hard for the model to generate meaningful word predictions. This experience showed that the model needs careful tuning to better understand the training data.

References

1. tarekhikal. "Text Generation with Tensorflow — NLP — RNN." Kaggle.com, Kaggle, 18 Sept. 2024, www.kaggle.com/code/tarekhikal/text-generation-with-tensorflow-nlp-rnn. Accessed 29 Sept. 2024.

2. "Tutorials — TensorFlow Core." TensorFlow, www.tensorflow.org/tutorials.

3. Unfold Data Science. "LSTM next Word Prediction in Python — LSTM Python TensorFlow — LSTM Python Keras — LSTM Python." YouTube, 24 July 2023, www.youtube.com/watch?v=jTjLf0ytSw8. Accessed 29 Sept. 2024.