Technical Report: Signature Verification and Word Completion Using Generative AI Techniques

Wasif Mehboob
Department of Data Science
National University of Computer and Emerging Sciences
Islamabad, Pakistan

***Abstract*—This report presents our approach to two distinct tasks. The first task focuses on signature verification using Convolutional Neural Networks (CNN) and a comparison with a manual feature extraction approach using Histogram of Oriented Gradients (HOG) with an SVM classifier. The CNN model achieved up to 92% accuracy on training data, while the HOG+SVM approach reached around 86%. The second task involves building a word-level LSTM model to perform sentence completion, trained on Shakespeare's plays. Although training the LSTM took a significant amount of time (10 epochs taking around one hour) and achieved roughly 40% accuracy initially, the model produces coherent word suggestions that improve with further training. This report outlines our methodology, experimental results, and discusses challenges encountered.**

***Index Terms*—Signature Verification, CNN, HOG, SVM, Word Completion, LSTM**

I. Introduction

This report covers two distinct tasks:

- **Signature Verification:** A system to classify handwritten signatures based on the directory number labels. The approach uses a CNN for automatic feature extraction and is compared with manual feature extraction using HOG combined with an SVM.
- **Word Completion:** A word-level LSTM model trained on Shakespeare's plays to predict the next word in a sentence. An interactive streamlit based user interface provides real-time word suggestions as the user types.

Both tasks explore different aspects of deep learning—one focusing on image classification and the other on sequence modeling.

II. Methodology

A. Signature Verification

Dataset and Preprocessing: The dataset consists of signature images stored in directories named after the user. Each image is read in grayscale, resized to 128×128 pixels, and normalized by dividing pixel values by 255.0. Labels are extracted from the directory name, and the data is split into training and validation sets.

Model Architecture: The CNN model includes:

- Two convolutional layers (with 32 and 64 filters) followed by max-pooling layers.
- A flatten layer to convert feature maps to a vector.
- A fully connected layer with 128 neurons, with dropout (20%) for regularization.
- A final Dense layer with softmax activation to predict the user label.

This work was conducted using Kaggle's T4 GPU and open-source datasets.

B. Word Completion

Dataset and Preprocessing: The text dataset is comprised of Shakespeare's plays. The text lines are cleaned by converting to lowercase, removing punctuation, and extra whitespace. The corpus is tokenized and sequences of five words (as input) with the subsequent word as the target are generated.

Model Architecture: The LSTM model includes:

- An Embedding layer that converts integer tokens into dense vectors of size 100.
- Two LSTM layers (each with 128 units); the first LSTM returns sequences to pass a complete sequence to the next LSTM.
- A Dense layer with softmax activation to predict the next word from the vocabulary.

For training efficiency, integer labels are used with the sparse categorical cross-entropy loss function.

User Interface: A Streamlit application is developed to allow users to type partial sentences and receive a real-time next-word suggestion by processing the user input with the trained LSTM model.

III. Results

A. Signature Verification

CNN Results: The CNN model was trained for 20 epochs and achieved approximately 92% accuracy with a loss around 20% on training data. On the test set (which was small and not vast), the CNN achieved 100% accuracy.

Manual Feature Extraction Results: The HOG feature extraction combined with an SVM (using a linear kernel) achieved around 86% accuracy with around 30% loss.

B. Word Completion

LSTM Training: Training the LSTM on Shakespeare's plays is computationally heavy. In our case, 10 epochs took nearly one hour on a Kaggle T4 GPU, and after 50 epochs, the model reached approximately 40% accuracy with a loss of around 40%.

Example Completions:

- Input: "to be or not" $\rightarrow$ Suggested next word: "to"
- Input: "all the world's a" $\rightarrow$ Suggested next word: "stage"

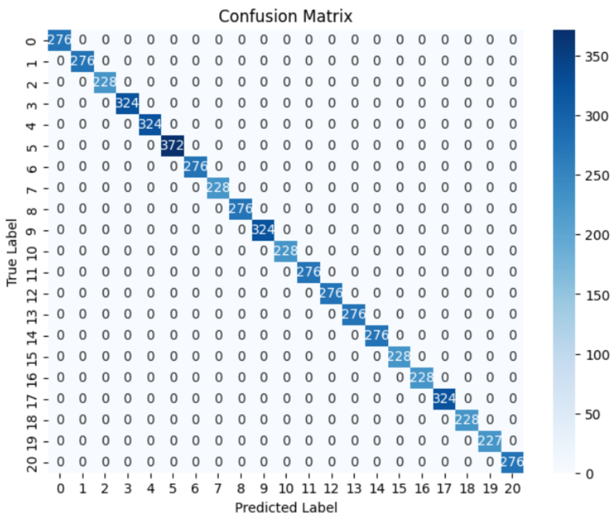

Fig. 1. Training and validation performance of the CNN model for signature verification.

```
Epoch 20/20
581/581 ━━━━━━━━━━━━━━━━━━━━ 7s 13ms/step - accuracy: 0.9263 - loss: 0.2070
```

Fig. 2. Training and loss of the CNN model for training.

```
Manual Feature Extraction with HOG and SVM: Evaluation
Accuracy: 0.861
Precision: 0.854
Recall: 0.86
F1 Score: 0.854
```

Fig. 3. Manual Feature Extraction with HOG and SVM Evaluation.

IV. Discussion

The signature verification experiments demonstrated that the CNN approach significantly outperforms manual feature extraction (HOG+SVM) on the available data, though the test set was limited. For the word completion task, the LSTM model shows promise despite initially low accuracy. The 40% accuracy must be considered in the context of a large vocabulary; even random guessing would yield low numbers. However, as training continues, the model's suggestions become more coherent and contextually relevant. The main challenges encountered were the extensive training time for the LSTM and the difficulty of optimizing next-word prediction for such a diverse text corpus.

V. Conclusion

This report shows that CNNs are highly effective for signature verification, outperforming traditional manual feature extraction methods. For word completion, although training is time-consuming and initial performance is modest, the LSTM model generates increasingly coherent suggestions over time. Future work may involve further hyperparameter tuning, data augmentation, and exploring alternative architectures to improve the prediction accuracy in the word completion task.

VI. Prompts

The following assignment prompts were used:

- Develop a signature verification system using CNNs and compare it with manual feature extraction (HOG+SVM).
- Build a word-level LSTM model trained on Shakespeare's plays to predict the next word in a sentence.
- Create an interactive user interface that dynamically provides next-word suggestions as a user types.
- Evaluate the impact of different hyperparameter settings on the model predictions.

VII. References

References

[1] IEEE Conference Template, Available: https://www.overleaf.com/latex/templates/ieee-conference-template/grfzhhncsfqn.

[2] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[3] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, 2015.

[4] Signature Verification Dataset: Available on Kaggle: https://www.kaggle.com/datasets/robinreni/signature-verification-dataset.

[5] Shakespeare-plays Dataset: Available on Kaggle: https://www.kaggle.com/datasets/kingburrito666/shakespeare-plays.