Assignment 1 Report

Mahrukh Wahidi 21i-1765

March 16, 2025

Contents

1 Problem 1: Signature Recognition using CNN

This section covers the implementation of a CNN model for signature recognition and comparison with manual feature extraction techniques.

1.1 Methodology

1.1.1 Dataset Description

The dataset used for the CNN model is the Signature Verification Dataset from Kaggle: `https://www.kaggle.com/datasets/robinreni/signature-verification`

Dataset Directory Structure:

```
train/
 002/       # Genuine signatures for ID 002
    002_02.PNG
    (other genuine images)
    002_forg/   # Forged signatures for ID 002
       (forged images)
 003/
    003_forg/
    ... (128 directories)
test/
 049/
    01_049.png   # Genuine signatures for ID 049
    049_forg/
```

```
        02_0210049.PNG  # Forged signatures for ID 049
  050/
     050_forg/
     ... (42 directories)
train_data.csv
test_data.csv
```

Train CSV Data Sample:

```
     genuine_image                   forged_image  label
0   068/09_068.png  068_forg/03_0113068.PNG      1
1   068/09_068.png  068_forg/01_0124068.PNG      1
2   068/09_068.png  068_forg/02_0124068.PNG      1
3   068/09_068.png  068_forg/01_0113068.PNG      1
4   068/09_068.png  068_forg/04_0124068.PNG      1
```

Test CSV Data Sample:

```
     genuine_image                   forged_image  label
0   068/09_068.png  068_forg/03_0113068.PNG      1
1   068/09_068.png  068_forg/01_0124068.PNG      1
2   068/09_068.png  068_forg/02_0124068.PNG      1
3   068/09_068.png  068_forg/01_0113068.PNG      1
4   068/09_068.png  068_forg/04_0124068.PNG      1
```

1.2 Preprocessing Steps

- Headers were added to CSV files to avoid conflicts.
- Images were loaded and resized to 128x128 pixels using OpenCV.
- Images were normalized and reshaped for training/testing.
- Data leakage was checked by flattening images and comparing samples in both datasets.
- MD5 hash values were generated for each image to detect overlaps.

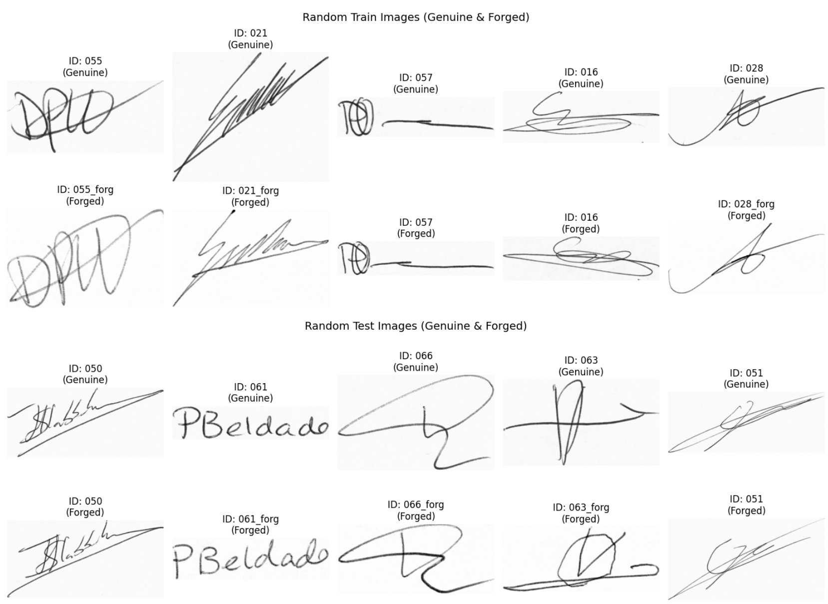

Figure 1: Sample genuine and forged signatures

1.3 Model Architecture

The CNN model architecture used is as follows:

Layer (Type)	**Output Shape**	**Param #**
conv2d_6 (Conv2D)	(None, 126, 126, 32)	320
max_pooling2d_6 (MaxPooling2D)	(None, 63, 63, 32)	0
conv2d_7 (Conv2D)	(None, 61, 61, 64)	18,496
max_pooling2d_7 (MaxPooling2D)	(None, 30, 30, 64)	0
flatten_3 (Flatten)	(None, 57600)	0
dense_6 (Dense)	(None, 128)	7,372,928
dense_7 (Dense)	(None, 1)	129

Table 1: CNN Model Architecture

1.4 Results

1.4.1 CNN

The classification report for the CNN model is as follows:

```
              precision    recall  f1-score   support

     Genuine       0.91      0.96      0.94       248
      Forged       0.96      0.91      0.93       252

    accuracy                           0.93       500
   macro avg       0.94      0.93      0.93       500
weighted avg       0.94      0.93      0.93       500
```

1.4.2 HOG + Logistic Regression

The classification report for the HOG + Logistic Regression model is as follows:

	precision	recall	f1-score	support
0	0.74	0.70	0.72	248
1	0.72	0.76	0.74	252
accuracy			0.73	500
macro avg	0.73	0.73	0.73	500
weighted avg	0.73	0.73	0.73	500

1.4.3 SIFT + SVM

The classification report for the SIFT + SVM model is as follows:

	precision	recall	f1-score	support
0	0.65	0.54	0.59	248
1	0.61	0.72	0.66	252
accuracy			0.63	500
macro avg	0.63	0.63	0.63	500
weighted avg	0.63	0.63	0.63	500

1.5 Comparison

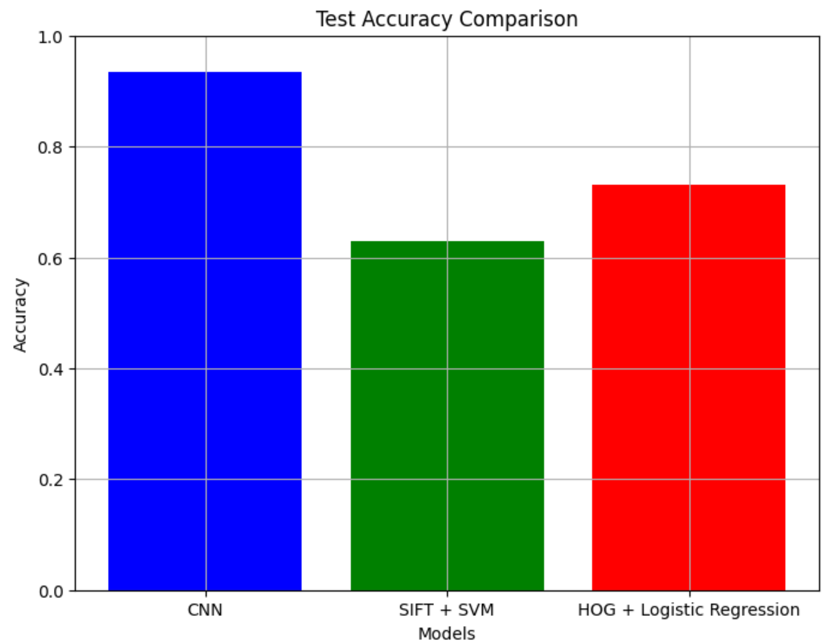

Figure 2: Test Accuracy Comparison of CNN, HOG + Logistic Regression, and SIFT + SVM

1.5.1 Confusion Matrices

1.6 Discussion

The CNN model outperformed the manual feature extraction techniques (HOG + Logistic Regression and SIFT + SVM). To address overfitting:

- Data augmentation techniques (rotation, scaling, flipping) were applied.
- L2 regularization and dropout layers were added.

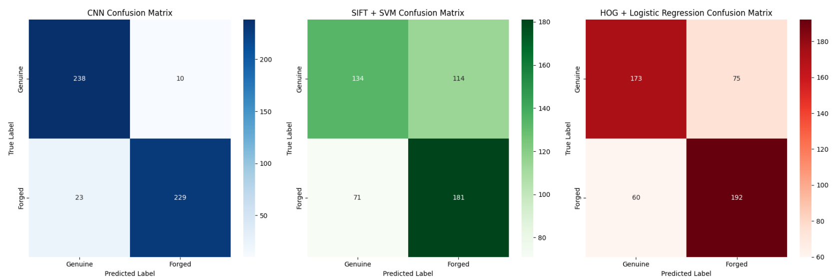

Figure 3: Confusion Matrix for CNN, HOG + Logistic Regression, and SIFT + SVM

2 Problem 2: Sentence Completion using LSTM

This section covers the implementation of a word-level LSTM model for sentence completion using the Shakespeare dataset from Kaggle.

2.1 Methodology

2.1.1 Dataset Description

The dataset includes:

- **Shakespeare_data.csv**: Contains lines from Shakespeare's plays.
- **alllines.txt**: A cleaned version of the text for training.
- **william-shakespeare-black-silhouette.jpg**: Used for word clouds.

2.1.2 Preprocessing Steps

- Missing values in the dataset were handled:
  - `Player` column: Filled with `'Unknown'`.

- `ActSceneLine` column: Filled with `'No Scene Info'`.
- `PlayerLinenumber` column: Filled with `0`.

- The total number of lines in `alllines.txt` was calculated.

2.1.3 Model Architecture

The LSTM model architecture is as follows:

Layer (Type)	**Output Shape**	**Param #**
Embedding	(None, 19, 256)	2,481,408
Bidirectional LSTM	(None, 19, 512)	1,050,624
Batch Normalization	(None, 19, 512)	2,048
Dropout	(None, 19, 512)	0
LSTM	(None, 256)	787,456
Dense	(None, 256)	65,792
Dropout	(None, 256)	0
Dense (Output)	(None, 9693)	2,491,101

Table 2: LSTM Model Architecture

2.2 Results

2.2.1 Model Accuracy

3 Discussion

In this study, the Convolutional Neural Network (CNN) demonstrated satisfactory performance and outperformed manual extraction methods for the signature

Figure 4: Model Accuracy during Training

dataset. However, training the CNN model posed significant challenges, particularly due to overfitting issues. Despite these difficulties, the CNN model ultimately achieved better results compared to traditional manual extraction techniques.

For the second problem, the Long Short-Term Memory (LSTM) model used for sentence generation initially yielded a very low accuracy of 13%. To address this, several improvements were implemented, including the use of the SwiGLU activation function and the incorporation of a Transformer Self-Attention Inspired (TSI) layer. These enhancements significantly increased the model's accuracy to 45%. However, due to time constraints, further optimization and refinement of the model could not be pursued. Future work could focus on exploring additional techniques to improve the accuracy and robustness of the sentence generation model.

4 Conclusion

In conclusion, this study demonstrated the effectiveness of Convolutional Neural Networks (CNNs) for signature verification, outperforming traditional manual extraction methods despite challenges such as overfitting during training. For sentence generation, the initial LSTM model achieved low accuracy (13%), but significant improvements were made by incorporating the SwiGLU activation function and a Transformer Self-Attention Inspired (TSI) layer, increasing accuracy to 45%. While these results are promising, further optimization is required to enhance performance. This work highlights the potential of advanced neural network architectures for complex tasks, but also underscores the need for additional research and development to address existing limitations.

5 Prompts

No specific prompts were used in this study.

6 References

1. ChatGPT OpenAI. (2023). ChatGPT. Retrieved from [https://openai.com/chatgpt](https://openai.com/c

2. DeepSeek DeepSeek. (2023). DeepSeek Official Website. Retrieved from https://www.deepseek.com

3. Kaggle Kaggle. (2023). Kaggle: Your Machine Learning and Data Science Community. Retrieved from https://www.kaggle.com