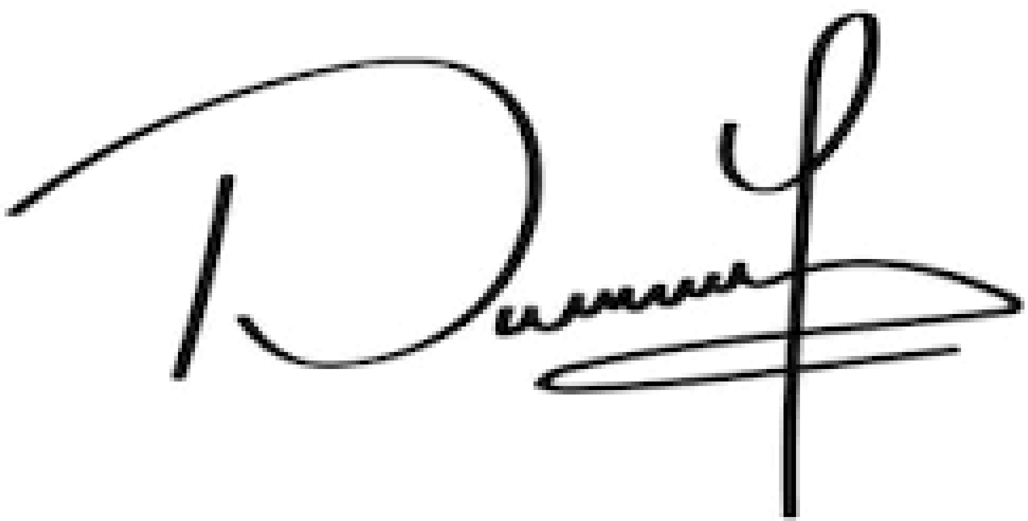
Handwritten Signature Verification



Made by

Salma Sherif 202200622

Mariam Yasser 202200886

Last updated:12/13/2024

Introduction to Signature Verification

Imagine signing your name on a document. That simple act holds immense power your signature is a representation of your identity and consent. But what happens when someone tries to mimic it? This is where signature verification comes into play a fascinating blend of human behavior, mathematics, and machine learning.

The Role of Technology

This challenge opens the door for automated signature verification systems, designed to detect forgery with precision and speed. These systems use advanced algorithms and machine learning models to compare signatures against known examples. They don't just look for surface level similarities but analyze patterns, pressure points, and overall style.

METHODOLOGY

DATASETS USED

Name	Made by	Folder Format	Size
handwritten signatures	Divyansh Rai	A folder containing 4 datasets each one has forged and real folders	Forged: 360 Real: 360
Handwritten signature verification	Zack Pashkin	A folder containing two folders one for forged and one for real images	Forged: 2983 Real: 3188
Signature-Forgery-Dataset	Saurab Shrestha	Two folders containing images. The ones starting with forge or forgeries or real were used	Forged: 3684 Real: 2497

Dataset challenges:

- The data showed an imbalance between datasets which made it hard to generalize
- The data had different features to be able to generate a pattern out of them combined
- Some datasets had an imbalance between real and forged images
- Loading data from multiple folders one inside the other

Models used:

1. Bidirectional LSTM
   - Why use: Captures temporal dependencies in sequential patterns of handwritten signatures for robust verification [1].
2. Siamese with contrastive loss
   - Why use: Compare signature pairs by learning a distance metric.
3. Pre trained ResNet (Resnet50)
   - Why use: Extracts high-level features from signature images, leveraging transfer learning for efficient and accurate classification.

Used methods to generalize

- Train the model on one dataset and use it to predict on the others. However, this approach resulted in poor accuracy.
- Train the model on one dataset, update its weights, and then fine-tune the pre-trained model on other datasets by unfreezing the layers and retraining. Unfortunately, this also yielded unsatisfactory accuracy.
- Combine subsets from each dataset to create a concatenated training dataset, train the model on this combined set, and evaluate it on the full three datasets.

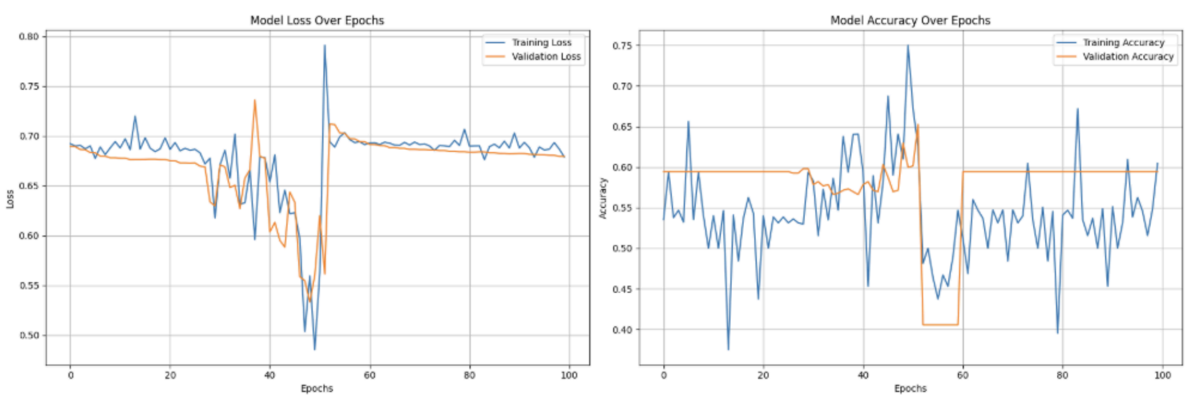
Challenge in Generalization for Signature Verification

The primary challenge in generalization, particularly in the domain of signature verification, is testing on unseen data. The main issue arises when the model is unable to determine whether a signature is forged or genuine if it has not encountered or trained on any samples of that specific signature during training.

Conclusion for generalized

For Bidirectional LSTM:-

- train_accuracy: 0.7031 - train_loss: 0.5079
- val_accuracy: 0.6808 - val_loss: 0.4751
- test_accuracy: 0.6702 - test_loss: 0.4988
- The Bidirectional LSTM model achieved a training accuracy of 0.7031 with a training loss of 0.5079, indicating moderate learning on the training data. The validation accuracy of 0.6808 and validation loss of 0.4751 reflects a comparable performance on unseen validation data, suggesting minimal overfitting.
- It has done a moderate job in generalization.



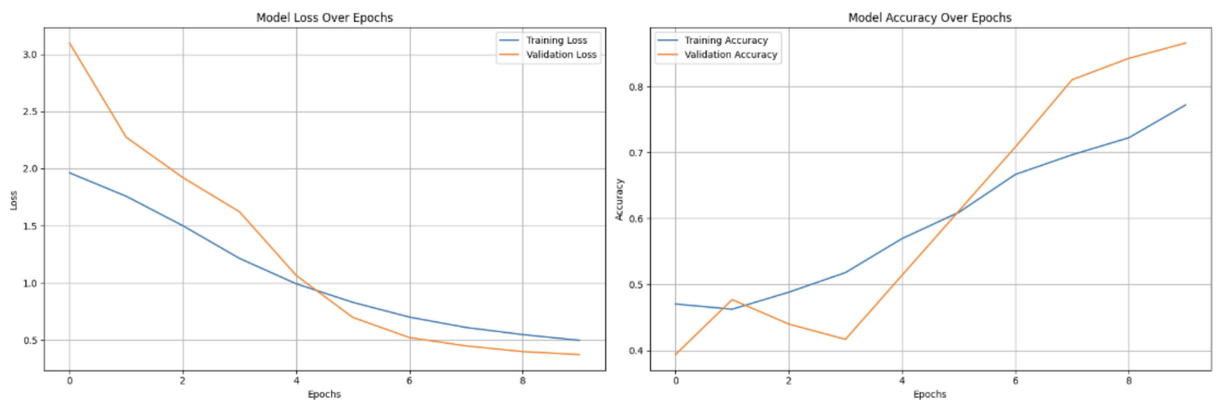
-

Conclusion for individual training

For ResNet50:-

—— Dataset1: used the ResNet trained on second data as it is already trained on some handwritten features

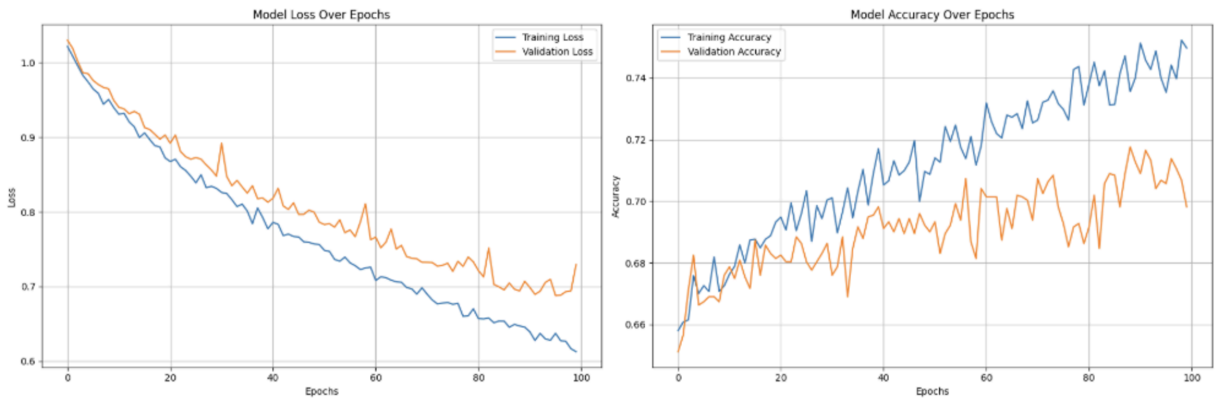
- Train_accuracy: 0.9067 - train_loss: 0.3713
- test_accuracy: 0.8657 - test_loss: 0.3270
- The model demonstrates strong performance with a high training accuracy of 90.67% and a test accuracy of 86.57%, indicating effective learning The slightly lower test accuracy compared to training suggests the risk of overfitting.
- Challenges: small data size



—— Dataset2: used keras ResNet

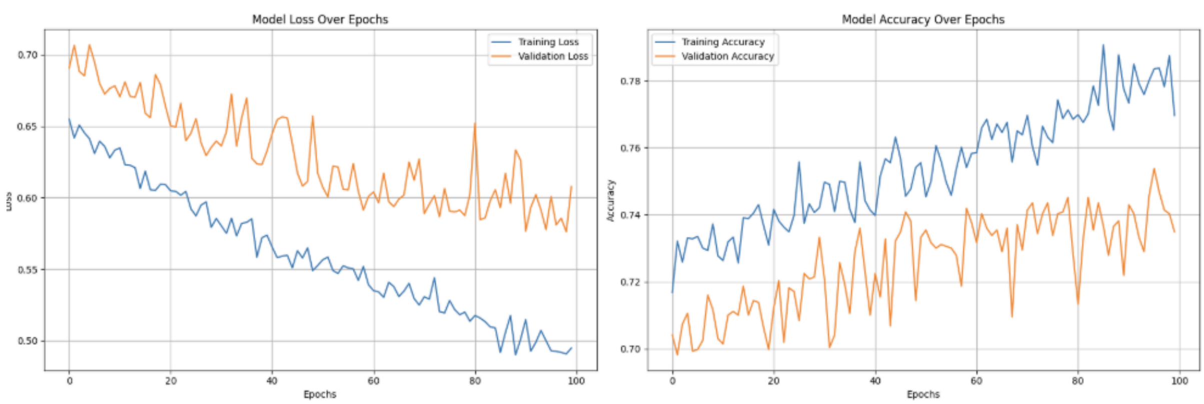
- Train_accuracy: 0.7707 - train_loss: 0.5211
- test_accuracy: 0.7451 - test_loss: 0.5875
- This was reached then tried to increase the accuracy but it turned to overfit even after making the image augmentation more complex
- The model achieves a training accuracy of 77.07% and a test accuracy of 74.51%, indicating decent learning but slight overfitting.

First 100 epochs



-

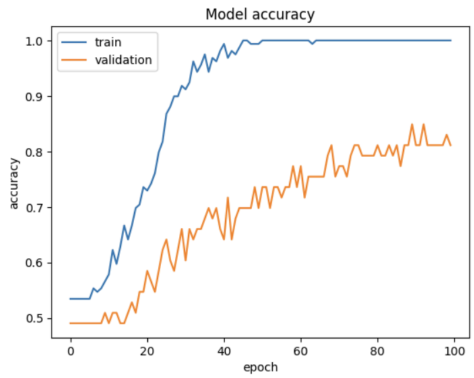
- Second 100 epochs

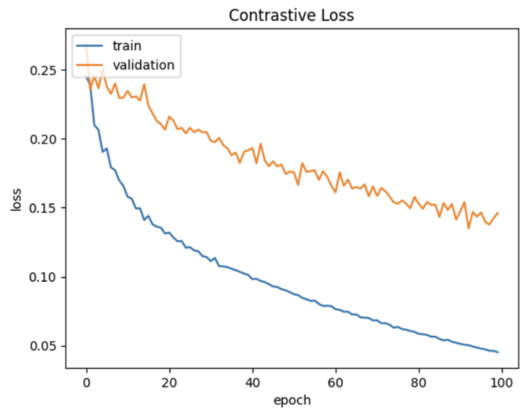


For [Siamese notebook] Dataset 1:-

- train_accuracy: 1.0000 - train_loss: 0.0516
- val_accuracy: 0.8113 - val_loss: 0.1460
- test_accuracy: 0.9074 - test_loss: 0.1236
- The Siamese model shows strong performance with an impressive training accuracy of 1.0000 and a low training loss of 0.0516.However, the validation accuracy of 0.8113 and validation loss of 0.1460 suggest a slight overfitting.
- When evaluated on the test set, the model achieved a test accuracy of 0.9074 with

a test loss of 0.1236. These results suggest that while the model is effective in distinguishing between genuine and forged signatures, further tuning or regularization techniques may improve its ability to generalize across different datasets.





Datasets

1. https://www.kaggle.com/datasets/divyanshrai/handwritten-signatures
2. https://www.kaggle.com/datasets/tienen/handwritten-signature-verification
3. https://www.kaggle.com/datasets/saurabstha5/signature-forgery-dataset

References

[1] Biometric signature verification using recurrent neural networks. (n.d.). IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/abstract/document/8270043