Handwritten Signatures Recognition using CNN based Siamese Network

Rasel Ahmed Bhuiyan

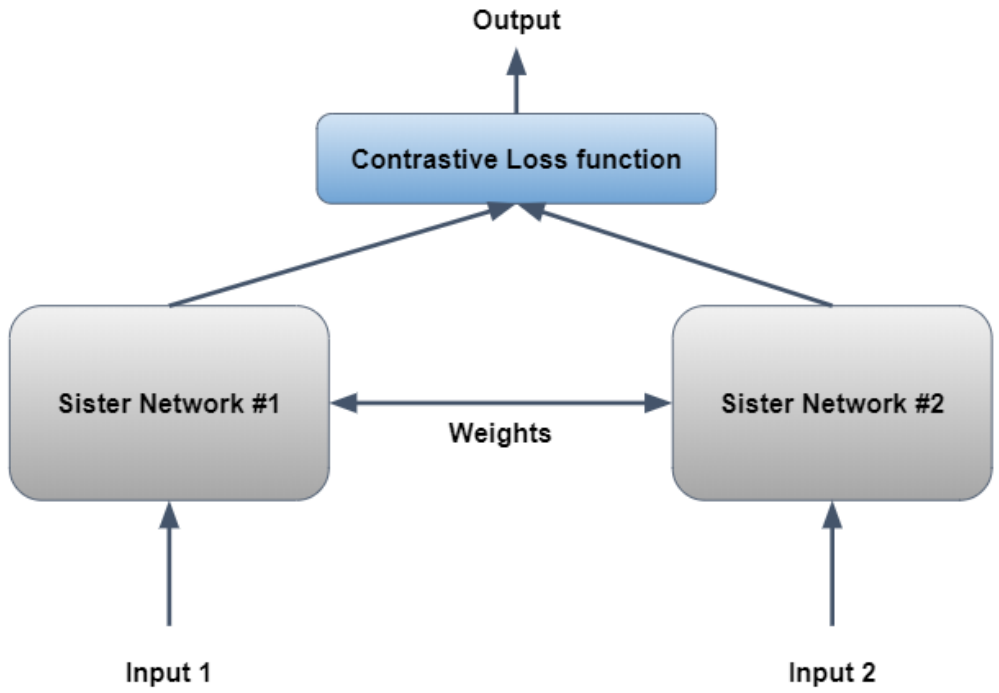
Index

Introduction

- Signature is one of the most popular and commonly accepted biometric hallmarks that has been used since ancient times for verifying different entities related to human beings, viz. documents, forms, bank checks, individuals, etc.
- Signature verification is a critical task and many efforts have been made to remove the uncertainty involved in the manual authentication procedure, which makes signature verification an important research line in the field of machine learning, computer vision, and pattern recognition.

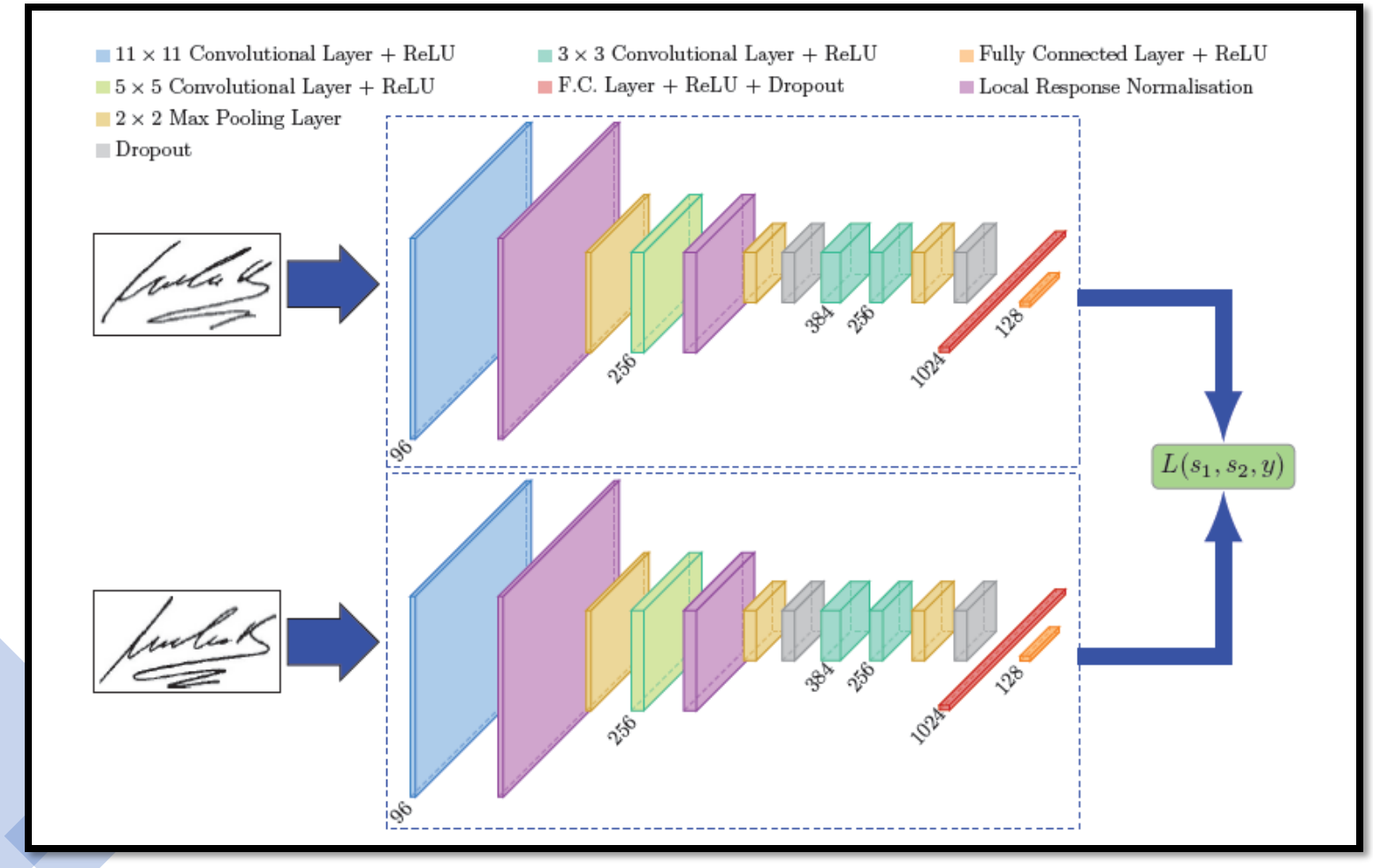
Siamese Network

- Siamese network is a class of neural network architectures that usually contains two identical subnetworks.
- The twin networks have the same configuration with the same parameters and shared weights.
- The parameter updating is mirrored across both the subnetworks.



$$(1-Y)\frac{1}{2}(D_W)^2 + (Y)\frac{1}{2}\{max(0, m - D_W)\}^2$$

Model Architecture



Layer	Size	Parameters
Convolution	$96 \times 11 \times 11$	stride = 1
Local Response Norm.	-	$\alpha = 10^{-4}, \beta = 0.75$ $k = 2, n = 5$
Pooling	$96 \times 3 \times 3$	stride = 2
Convolution	$256 \times 5 \times 5$	stride = 1, pad = 2
Local Response Norm.	-	$\alpha = 10^{-4}, \beta = 0.75$ $k = 2, n = 5$
Pooling + Dropout	$256 \times 3 \times 3$	stride = 2, $p = 0.3$
Convolution	$384 \times 3 \times 3$	stride = 1, pad = 1
Convolution	$256 \times 3 \times 3$	stride = 1, pad = 1
Pooling + Dropout	$256 \times 3 \times 3$	stride = 2, $p = 0.3$
Fully Connected + Dropout	1024	$p = 0.5$
Fully Connected	128	

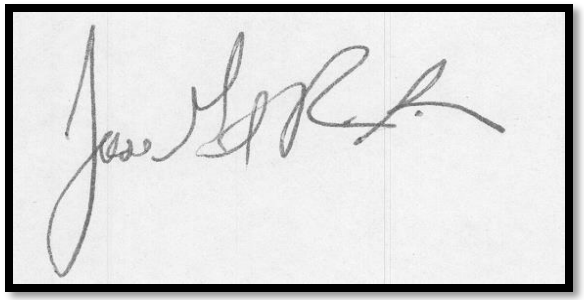
Dataset description

- In order to evaluate my signature verification deep learning model inspired by the Siamese Network, I have considered two widely used benchmark databases.
  - CEDAR Signature Dataset
  - BHSig260 Signature Dataset

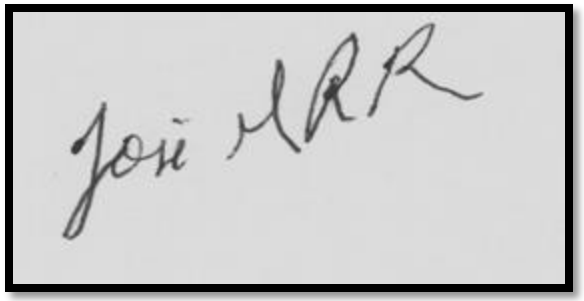
CEDAR Signature Dataset

- 55 Subjects
- Each subject has 24 genuine and 24 forgery signatures
- Training set: 18,170 pairs (9085 genuine and 9085 forgery pairs)
- Test set: 12,140 pairs (6070 genuine and 6070 forgery pairs)

Genuine

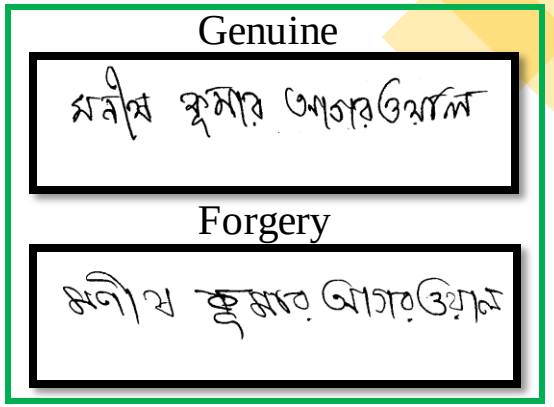


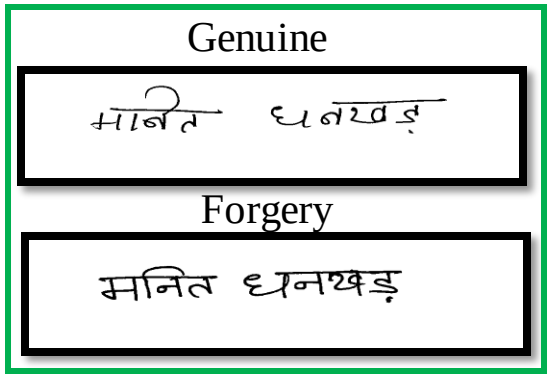
Forgery



BHSig260 Signature Dataset

- The BHSig260 signature dataset contains two sub-datasets namely Bengali and Hindi of 260 subjects.
- Bengali:
  - 100 subjects
  - Each subject has 24 genuine and 30 forgery signatures
  - Training set: 33,120 pairs (16,560 genuine and 16,560 forgery pairs)
  - Test set: 22,080 pairs (11,040 genuine and 11,040 forgery pairs)
- Hindi:
  - 160 subjects
  - Each subject has 24 genuine and 30 forgery signatures
  - Training set: 52,992 pairs (26,496 genuine and 26,496 forgery pairs)
  - Test set: 35,328 pairs (17,644 genuine and 17,644 forgery pairs)



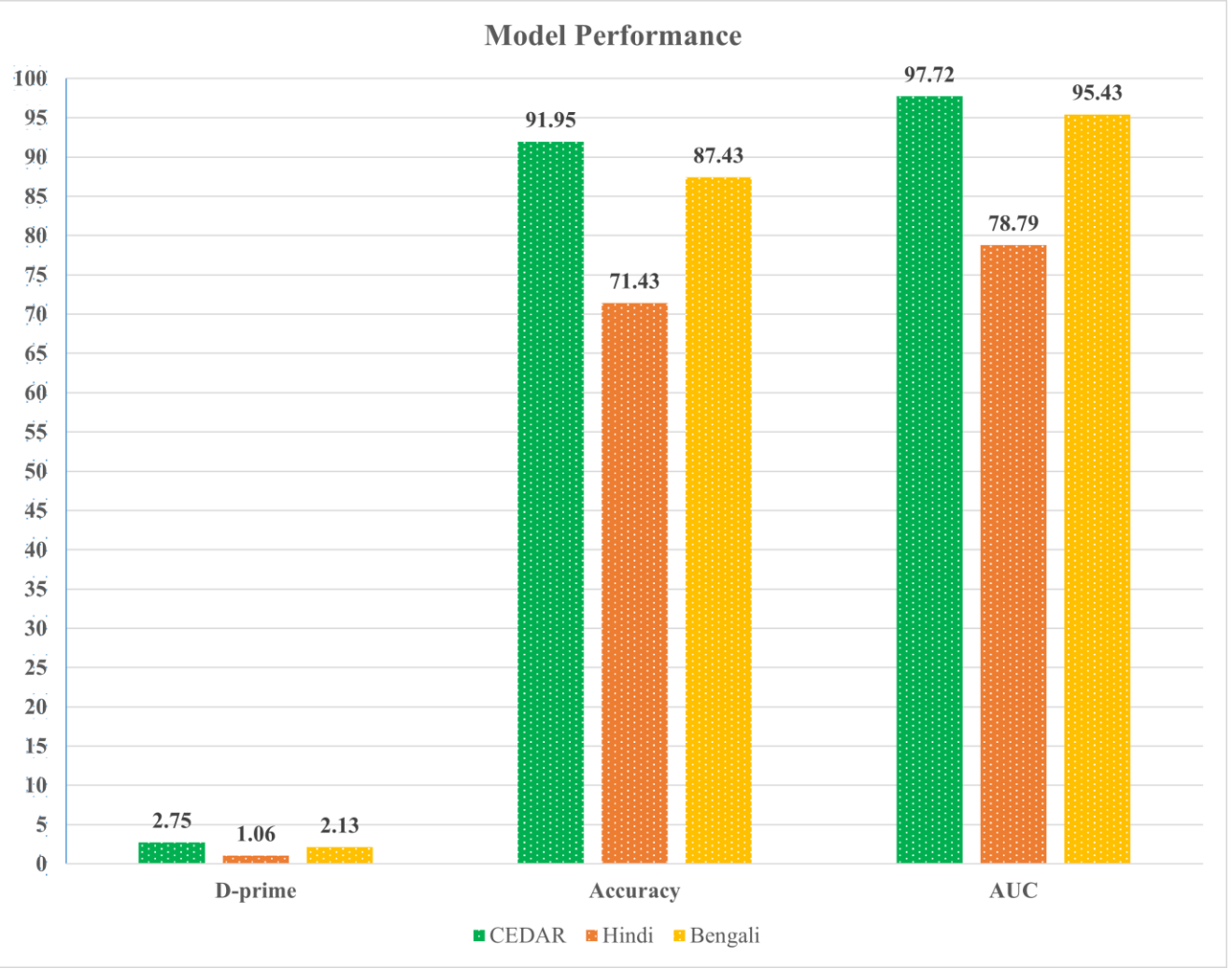


Data preprocessing

- Neural network typically needs images of the same sizes.
- However, for the signature images, I have considered different sizes ranging from (153 X 258) to (819 X 1137).
- Resized all the images to a fixed size of 155 X 220
- Convert into Grayscale
- To reduce the complexity of the pixel values, I have normalized them by a mean of 0.5 and a standard deviation of 0.5. ((x-mean)/std)

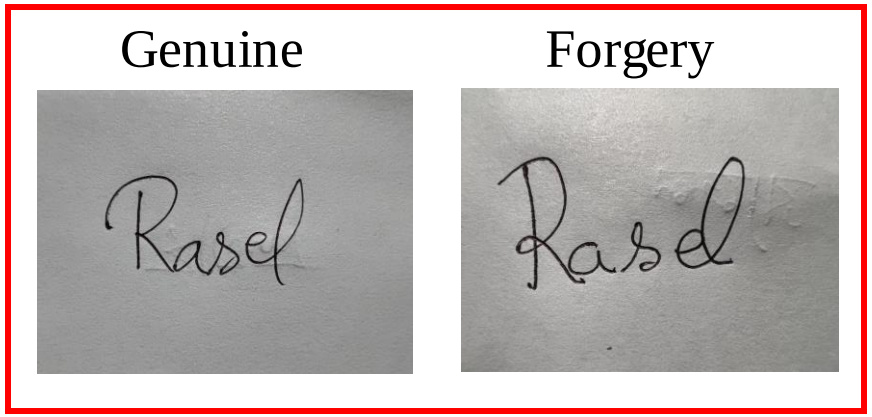
Experimental Results

- Trained the model using the ADAM optimizer.
- Number of epochs 20.
- The mini-batch size is equal to 8.
- Initial learning rate equal to 1e – 6.

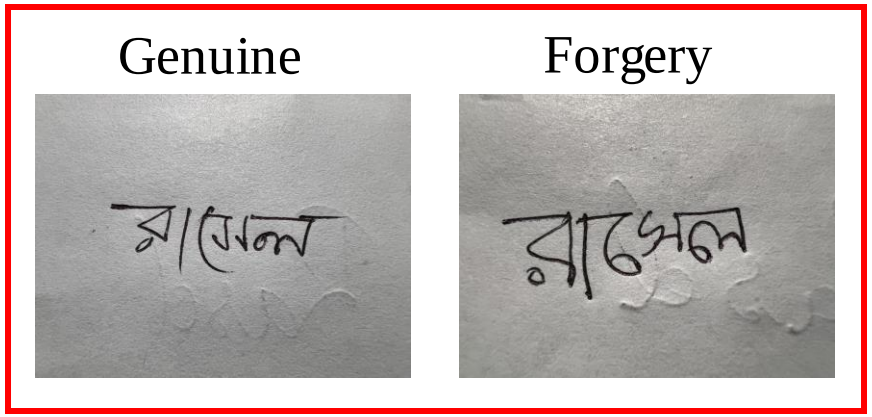


Model Robustness

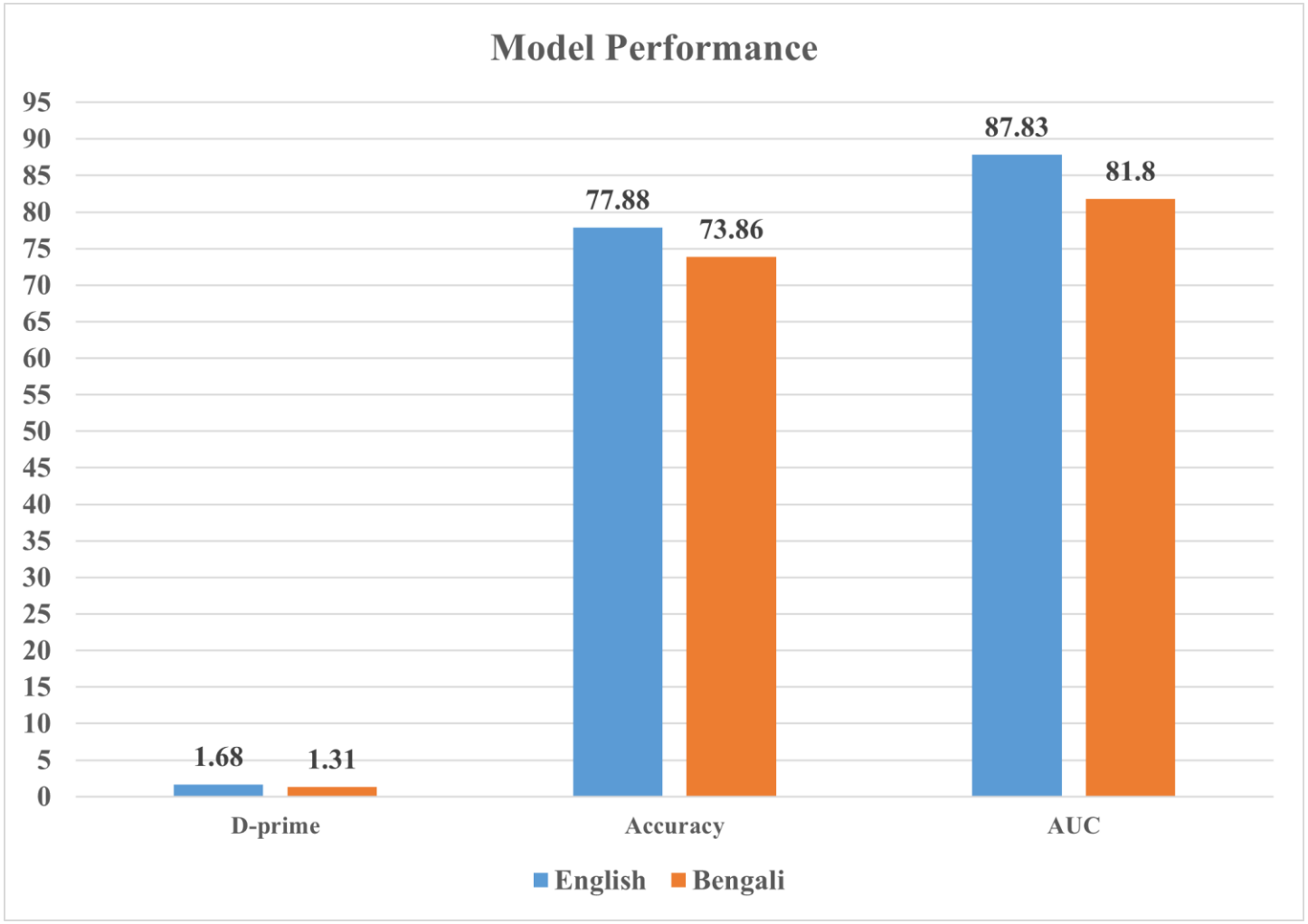
- English:
  - 11 genuine and 11 forgery signatures
  - 200 pairs



- Bengali
  - 11 genuine and 11 forgery signatures
  - 200 pairs



Performance of Created Test Set



Future Work

- From the experimental results, clearly showed that the designed model didn’t perform well for all the datasets and there might be overfitting or underfitting.
- To mitigate this problem and enhance the model performance, I will change the model architecture or apply different learning rates and batch sizes.
- Moreover, will try other optimizers.

Conclusion

- Created a CNN-based Siamese Network.
- Evaluated the model on two benchmark signature datasets.
- Checked the robustness of the model using the created test set.