Report
(Writer_Verification_Challenge)

Abstract

This challenge aims to identify whether handwriting is written by the same person. For this, we have used the Siamese network. This model can be trained on a limited number of samples and provide better accuracy in prediction.

Method

Preprocessing:-

In preprocessing, an image from a writer is taken and then cropped to make it of the size 224*224. And during this process, the patches that do not contain any written letter were also removed, as shown in (Figure 1) below.

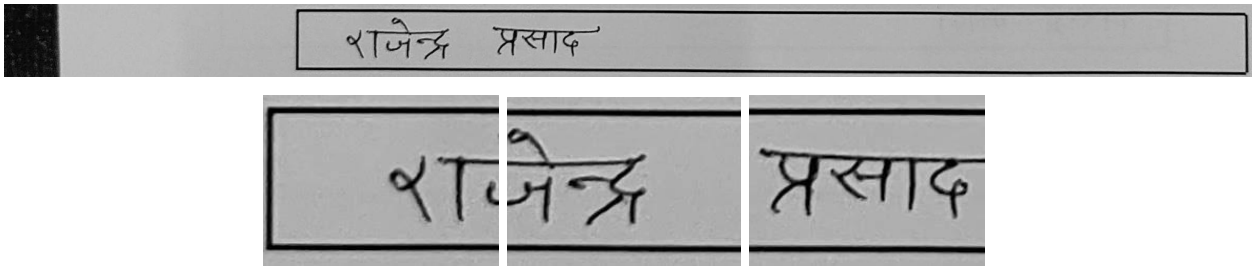


Figure (1)

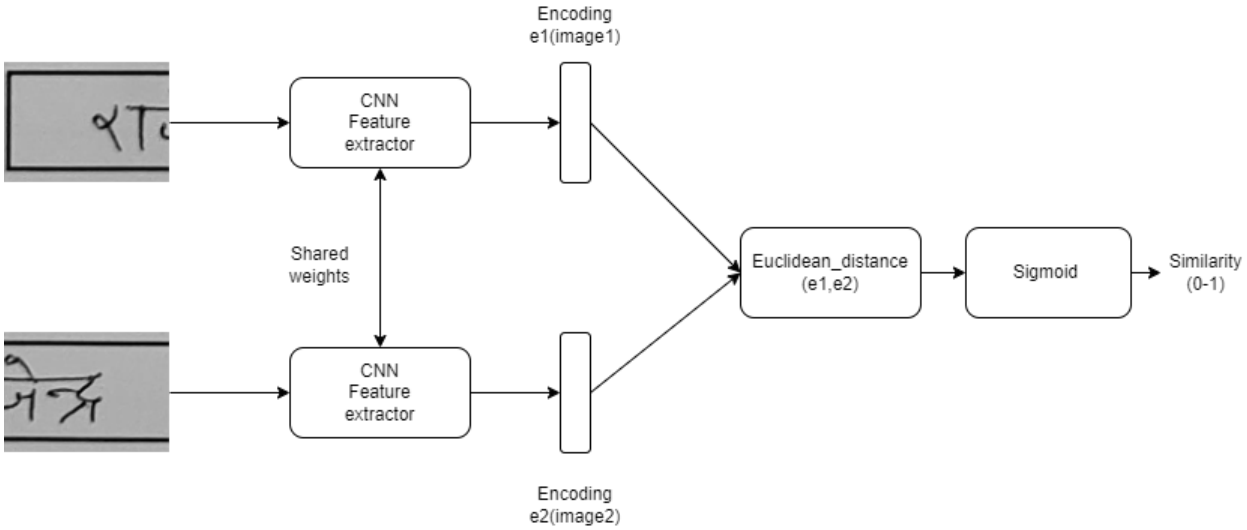
Making pairs of images:-

After making these patches, we make a pair of patches in which each patch is paired with another patch of the same writer, and another pair is made by pairing it with another writer patch. The pair which has the same writer is given a label as 1. And the patch which contains different writer patches is given as 0.

Training:

Siamese network

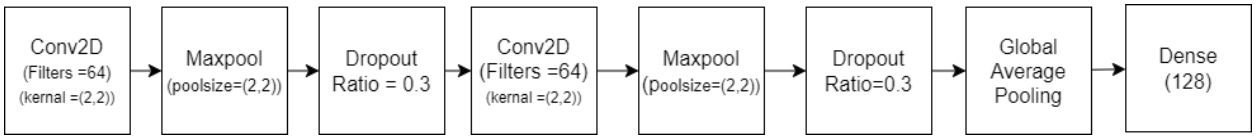
For training the model, we are using the Siamese network. In This, we fed the pair into the feature extractor and got the extracted features. Then we calculated the Euclidean distance between them and then applied a sigmoid layer which gave an output between the range(0 - 1) in which the value near 0 means not similar and near 1 means similar. Figure(2)



Figure(2)

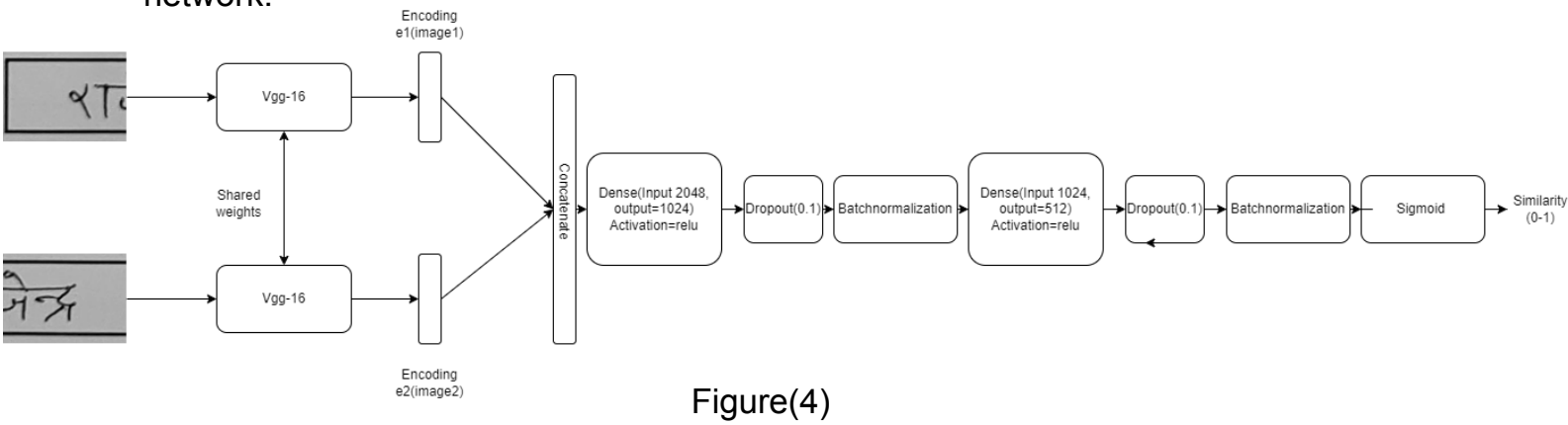
Feature extractor:

Feature extractors can be any cnn model like Vgg16, Resnet, or some custom cnn. Cnn used for the semi-final model can be seen in Figure (3)



Figure(3)

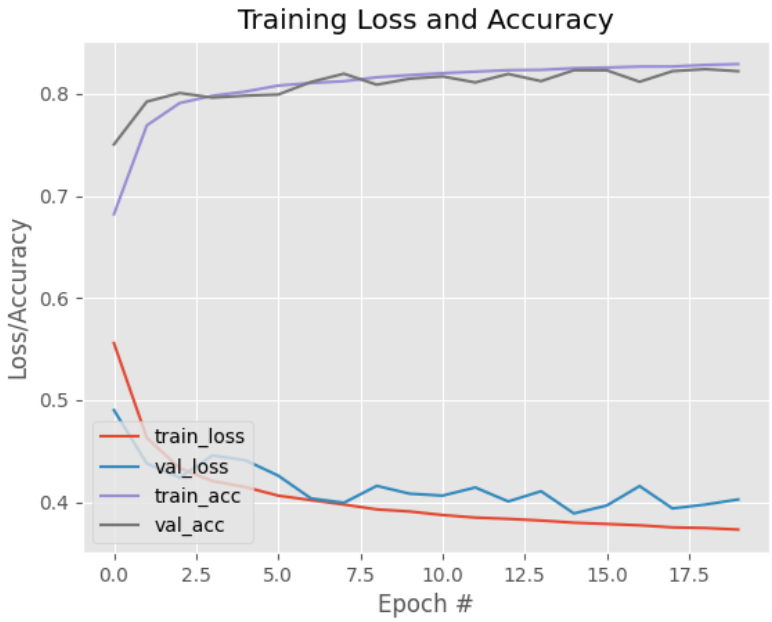
Model used for final-submission was Vgg-16 as Feature extractor with neural net as a comparator which was used in place of euclidean distance which can learn more complex patterns in the writing. Figure (4) shows the design of the neural network.



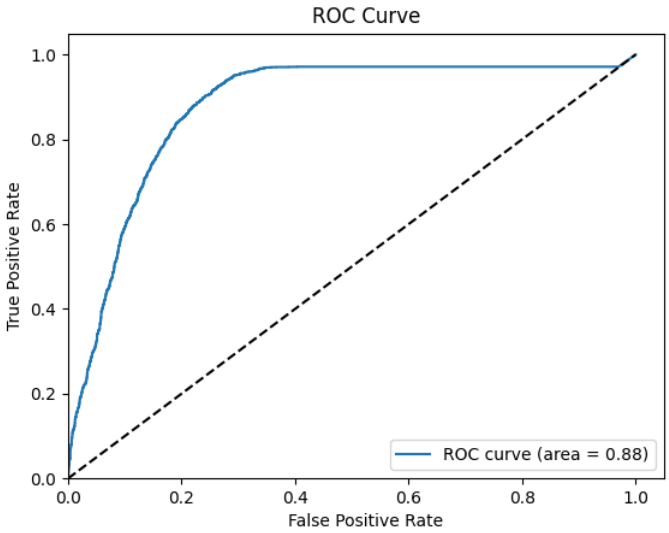
Figure(4)

Training

Various hyperparameters were varied to see improvements in the AUC score on the validation set and training time till convergence of the AUC. USed adam as the optimizer and loss function as Binary cross entropy.
The model was trained for 50 epochs
For implementation of optimizer,model,losses,etc Tensorflow library was used.



Training curve



ROC curve on validation dataset

References

1) https://arxiv.org/abs/1606.06472
2) https://stackoverflow.com/questions/33949831/how-to-remove-all-lines-and-borders-in-an-image-while-keeping-text-programmatica
3) https://pyimagesearch.com/2020/11/30/siamese-networks-with-keras-tensorflow-and-deep-learning/