CSE4214_Paper

by Jaasia Anjum

Authenticating Handwritten Bengali Signatures with Deep Learning

1st Jaasia Anjum
Dept. of Computer Science and Engineering
Ahsanullah University of Science and Technology
Dhaka, Bangladesh
190204051@aust.edu

2nd Naosin Akhter Hia
Dept. of Computer Science and Engineering
Ahsanullah University of Science and Technology
Dhaka, Bangladesh
190204050@aust.edu

3rd Hujifa Akter Shila
Dept. of Computer Science and Engineering
Ahsanullah University of Science and Technology
Dhaka, Bangladesh
190204042@aust.edu

4th Md Ashfaq Hosain Rafi
Dept. of Computer Science and Engineering
Ahsanullah University of Science and Technology
Dhaka, Bangladesh
180104099@aust.edu

Abstract—In the field of information security, biometric systems play an important role. One area of study in biometrics is automatic signature identification and verification. This is important because people commonly use signatures to prove who they are, and some rules and laws accept signatures for this purpose. In this research, we look into how well a computer system can check if a Bengali signature is genuine, even when it's not done in real time (offline). Bengali signatures have a unique style that differs from signatures in Western languages. Although Bengali is the sixth most spoken language in the world, there have been very few attempts to authenticate Bengali signatures. Therefore, the issue of the Bengali signature verification needs to be addressed. This study uses several transfer learning techniques with Convolutional Neural Networks, for the sake of verifying Bengali signatures. To improve the accuracy and prediction of the models, the optimizers and hyper-parameters have been studied thoroughly to discover the ideal combination. A transfer learning approach, such as VGG-19, MobileNetV2, and Xception is used with customized CNN layers. A standard Bengali Dataset taken from Kaggle: BHSig60, is used to train our models which contain 5400 Bengali signatures of 100 people. Also we made a signature dataset of 10 people and added with the already available dataset. The MobileNetV2 model achieved the highest accuracy of 90.10%, conquering the rest of the models and promising an improvement in the authentication of Bengali Signature overall.

I. INTRODUCTION

Signature verification is a vital step in ensuring the authenticity of documents and maintaining security. It plays a key role in confirming a person's identity. This process is essential for trustworthiness, ensuring that documents remain unchanged and reliable. In contemporary practices, manual visual verification of signatures is both tedious and prone to errors. The development of a robust signature verification system is a very crucial task to prevent document misrepresentation in high-profile industries such as business, finance, intelligence, and government. We aim to lead the way in creating a verification system that uses machine learning to verify Bengali signatures. This system not only ensures the accuracy of authentication but also makes sure that the process is faster.

The goal of this work is to make Bengali signatures more easily verified. We want to improve security and reduce the possibility of fraud in significant transactions where individuals are verified using their unique signatures. Nowadays, it's common practice to visually verify signatures, which can be tiring and error-prone. Our goal here is to develop a machine learning-based verification system that can prevent fraud and other criminal activities rapidly and precisely. Furthermore, it is not very popular to use this kind of work for Bengali signature verification these days; therefore, we aim to make it available to as many people as possible.

Bengali signature verification through machine learning is not yet widespread, and we aspire to democratize this technology, making it accessible to a broader audience. Our vision is to extend the benefits of advanced signature verification to as many people as possible, fostering a more secure environment for critical transactions.

We have witnessed the usefulness of using convolutional neural network architectures for signature verification. [1] Along with CNN architectures, studies have shown that using transfer learning can bring better results. [1] In this article, we are employing a step-by-step procedure to authenticate Bengali signatures. We are using an enriched dataset with 5400 signatures from a total of 100 people. We have also created a small dataset of our own, including 216 signatures from four people. We made sure the data was resized to a uniform size for better performance. We have employed a transfer learning method for our model. Then, we used customized convolutional neural network (CNN) architectures (VGG-19, MobileNet, and Xception) to decipher the unique patterns seen in Bengali signatures. To put it another way, we aim to build a method that can reliably and efficiently verify Bengali signatures to increase security, particularly in settings where signatures are crucial.

In essence, our project aims to deliver a robust method capable of reliably and efficiently verifying Bengali signatures, significantly enhancing security measures, especially in contexts where signatures play a pivotal role.

II. LITERATURE REVIEW

It is important to have different kinds of data, learn good features, and change how the models are built to make strong signature checkers. The studies by Hafemann et al. [1], Dey et al. [2], and Shayekh et al. [3] CNN-based models in deep learning are good at verifying handwritten signatures done offline. Hafemann et al. [1]'s approach, used CNNs to learn how to tell signatures apart, even if they're by different people. Their way got a really low error rate of 1.72% on GPDS-160, doing way better than what others have done before.

Dey et al. [2]'s SigNet is a special network that's great at checking signatures, no matter who wrote them, even on different sets of data. SigNet gets super high accuracy, even hitting 100% on the CEDAR dataset.

Shayekh et al. [3] changed the VGG-19 design by adding more CNN and fully connected layers. Their model learned from improved and bigger datasets from ICDAR, Kaggle, and CEDAR, reaching high accuracies: 100% on ICDAR, 94.44% on Kaggle, and 88% on CEDAR. Although Bengali is the sixth most spoken language, Bengali signature verification is a research field that has yet to discover much potential. Srikanta et al. [4]'s they focused on checking BHSig60 Bengali signatures using a set limit, and a specific chain with Nearest Neighbour as classifier. That worked well on getting a low error rate of 15.57% is a big step forward in checking Bangla signatures on paper.

Alaei et al. [5]'s innovative method, used a special way to show signatures that worked well for this dataset with fewer training samples. Their approach used interval symbolic representation and a fuzzy similarity measure. It utilizes local binary pattern features to create symbolic data. For each signature class, establish a unique model for individual signatures.

Pal et al. [6]'s method for Indian writing (Bangla and Hindi) using textures and special features like LBP and ULBP. Their Nearest Neighbour checker, especially with ULBP, gets excellent verification rates. This shows that using textures is important when checking signatures, especially in different kinds of writing. They achieve 66.18% accuracy in Bengali signatures in their data set.

Each study brings new ways to do this, showing how different techniques to pick out key parts and check signatures can be effective This could lead to better ways to confirm identities, especially across different languages and writing styles.

III. METHODOLOGY

A. Dataset

In the field of signature verification, there aren't many signature datasets that are freely accessible to the public. The only available Bengali offline signature database,
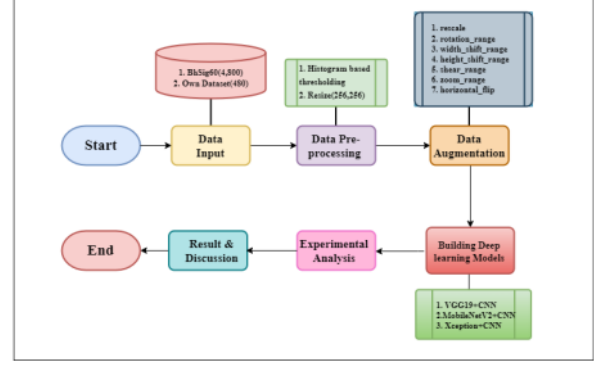

Fig. 1. Methodology

BHSig260(Bengali) [7] will be trained with our model. BHSig260(Bengali) has Bengali signatures of 100 people. To train our model more effectively, we added the signatures of 10 more people. 24 Genuine signatures and 24 forged signatures have been added for each person in both Kaggle dataset and our dataset. There are 2640 genuine signatures and 2640 forged signatures in this dataset of 100+10=110 people. We took the first 20 forged signatures in our training dataset, then the next three for our test dataset, and the rest three for our validation dataset. We did the same with the genuine signatures. The details are shown in this table:

Forged			Genuine		
24			24		
Train	Test	Validation	Train	Test	Validation
20	2	2	20	2	2

TABLE I
EACH PERSON DATA DIVISION

Forged			Genuine		
2400			2400		
Train	Test	Validation	Train	Test	Validation
2000	200	200	2000	200	200

TABLE II
WHOLE DATASET DIVISION

B. Pre-processing and Data Augmentation

The Kaggle dataset BHSig60 (Bengali) had previously been pre-processed using a histogram-based thresholding technique. After scaling the data, each image's dimensions were resized to 256x256x3. Next, we normalized our data. The training set has undergone flips, zooms, and rotations to enhance model training and increase the amount of data.

We added ten people's signatures, 24 forged signatures, and 24 genuine signatures, and we used histogram-based thresholding techniques to process our images. We took the threshold value of 120. Similarly, we resize the data to 256x256x3.

Genuine Signatures	Forged Signatures

TABLE III
COMPARISON OF GENUINE AND FORGED SIGNATURES

Before Pre-processing	After Pre-processing

TABLE IV
COMPARISON OF BEFORE AND AFTER PRE-PROCESSING

C. Deep Learning Models

1) VGG-19: We have used VGG-19 as our base model, which has been trained with weights from the imagenet dataset. The layers of this base model are subsequently frozen to preserve the learned features during the upcoming training process. The model architecture extends beyond the VGG19 base by incorporating additional convolutional and fully connected layers. Three convolutional layers with 256, 128, and 64 filters have been used, each with a 3*3 MaxPooling kernel size. ReLU was used as an activation function for these layers. The epoch size we used is 20. There are three densely connected layers with 512, 256, and 128 neurons. The final layer employs a Sigmoid activation function for binary classification. The model is compiled, trained, saved, and then loaded for evaluation. Evaluation metrics such as precision, recall, and a confusion matrix are calculated for the binary classification task on a test dataset.

2) MobileNetV2: We implemented an advanced convolutional neural network (CNN) architecture by leveraging the MobileNetV2 model, pre-trained on the ImageNet dataset. The base model is loaded with an input shape of (256, 256, 3), and its top layers, including fully connected layers, are omitted. To preserve the learned features, all layers of the pre-trained MobileNetV2 model are frozen during the training process. Our custom CNN design involves initializing a sequential model, into which the MobileNetV2 base model is seamlessly integrated. Further refinement is achieved by introducing an additional 3 convolutional layers (256, 128, 64) with progressively decreasing filter sizes, accompanied by strategic max-pooling operations to enhance feature extraction. The output undergoes flattening, followed by the three dense layers, which

have 512, 256, and 128 neurons with diminishing sizes to capture intricate patterns in the data. The ultimate layer consists of a single neuron employing a sigmoid activation function, which aligns with the requirements of binary classification tasks. The epoch size we used is 20. This enhanced CNN architecture aims to improve image classification performance by combining the knowledge transfer from a pre-trained model with fine-tuned convolutional layers.

3) Xception: We have employed the Xception model. This model utilizes transfer learning by using the pre-trained Xception model as a feature extractor. Here, a new sequential model is established, incorporating the pre-trained Xception model as the initial layer. Subsequent layers include three Convolutional (Conv2D) 32, 64, and 128 filter layers with 3*3 MaxPooling2D interspersed, promoting feature extraction. A Global Average Pooling 2D layer consolidates spatial information, followed by a dense layer with 256 neurons with ReLU activation and a dropout layer for regularization. The final layer employs a dense layer with a sigmoid activation function, suitable for binary classification tasks. The epoch size we used is 20. All layers of the pre-trained Xception model are set to non-trainable to retain learned features. For training configuration, the model is compiled using the Adam optimizer and binary cross-entropy loss. This setup prepares the model for training on specific data, leveraging the pre-trained Xception features to enhance its abilities in binary image classification.

IV. EXPERIMENTAL ANALYSIS

At first, we approached without transfer learning and with customized CNN architecture only. We tested various combinations of convolutional layers with our dataset. Two of the CNN architectures gave better results than the others. The first one, comprising 32 and 64 neurons in the convolutional layer and 128 neurons in the dense layer, gave 71% accuracy. The second one, comprising 256,128,64 neurons in convolutional layers, 512,256,128 neurons in dense layers, and a 0.5 dropout rate, gave an accuracy of 78% which was the best one so far.

With this model, we approached the transfer learning procedure. We used VGG-19, MobileNetV2, and Xception as our pre-trained models. VGG-19, MobileNetV2, and Xception achieved much satisfactory accuracy rate. Although their accuracy rate was pretty high, these models fluctuated in precision and recall scores. By the confusion matrix of these, we saw that, for Xception and MobileNetV2 the False Positive rate was higher, and for the VGG-19 model, the False Negative rate was higher. Both are not satisfactory for our model.

We are working on the Precision and Recall rates of the three models and hope to create a balance among them to identify a forged or a genuine signature.

V. RESULT AND DISCUSSIONS

We tried out several different models. Our main models are the customized CNN architecture pre-trained with VGG-19, MobileNetV2, and Xception. We employed the criteria of recall score, accuracy, and precision.

VGG-19

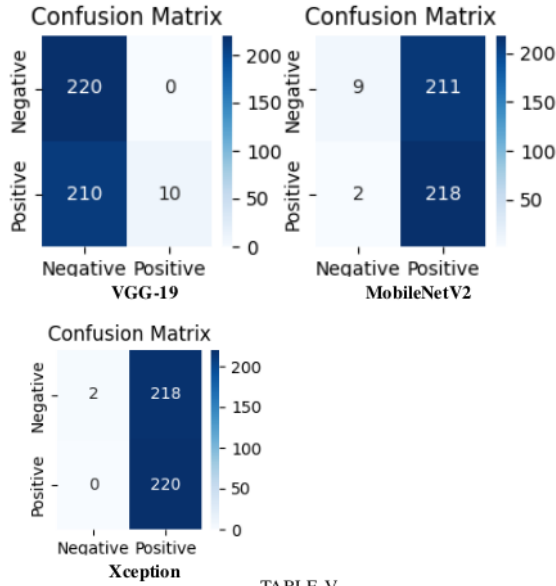**MobileNetV2**

Xception

TABLE V
CONFUSION MATRIXES OF DIFFERENT MODELS

Precision, Recall, and Accuracy can be defined as follows:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \quad (1)$$

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \quad (2)$$

$$\text{Accuracy} = \frac{\text{Correct Predictions}}{\text{Total Predictions}} \quad (3)$$

The summary of the results is shown below in the table: In

Models	Accuracy	Precision	Recall
VGG-19	89.32	100	4.5
MobileNetV2	90.10	50.81	99.09
Xception	84.89	50.22	100

TABLE VI
RESULT COMPARISON

our model, MobileNetV2 demonstrated superior performance with the highest accuracy at 90.10%, outperforming VGG19 at 89.32% and Xception at 84.81%. However, precision and recall scores exhibited significant variations. VGG-19 achieved the highest precision at 100%, while Xception scored 50.22%, and MobileNetV2 scored 50.81%. Although Xception secured the highest recall score at 100%, MobileNetV2 followed closely with 99%, while VGG19 lagged at only 4.5%. In summary, considering overall performance, MobileNetV2 emerges as the top-performing model.

VI. CONCLUSION AND FUTURE WORKS

As a result of our thorough study of the models, we have determined that, out of VGG-19, MobileNetV2, and Xception,

MobileNetV2 is the most accurate. Although there were occasional swings in the precision and recall ratings, MobileNetV2 performed significantly better overall. Still, there remains a light dissatisfaction about the model on how they fluctuated in precision and recall rates. To remedy that, we are working on a model that will compare the signatures pixel by pixel between any given two pictures. After comparing the signatures, it will generate a score that will determine whether the signature is genuine or forged. We are currently working on that model in the hope of a better outcome. This makes us optimistic about our contribution to improving the verification of Bengali signatures.

REFERENCES

[1] Luiz G Hafemann, Robert Sabourin, and Luiz S Oliveira. Learning features for offline handwritten signature verification using deep convolutional neural networks. *Pattern Recognition*, 70:163–176, 2017.

[2] Sounak Dey, Anjan Dutta, J Ignacio Toledo, Suman K Ghosh, Josep Lladós, and Umapada Pal. Signet: Convolutional siamese network for writer independent offline signature verification. *arXiv preprint arXiv:1707.02131*, 2017.

[3] Shayekh Mohiuddin Ahmed Navid, Shamima Haque Priya, Nabiul Hoque Khandakar, Zannatul Ferdous, and Akm Bahalul Haque. Signature verification using convolutional neural network. In *2019 IEEE International Conference on Robotics, Automation, Artificial-intelligence and Internet-of-Things (RAAICON)*, pages 35–39. IEEE, 2019.

[4] Srikanta Pal, Alireza Alaei, Umapada Pal, and Michael Blumenstein. Off-line bangla signature verification: an empirical study. In *The 2013 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. IEEE, 2013.

[5] Alireza Alaei, Srikanta Pal, Umapada Pal, and Michael Blumenstein. An efficient signature verification method based on an interval symbolic representation and a fuzzy similarity measure. *IEEE Transactions on Information Forensics and Security*, 12(10):2360–2372, 2017.

[6] Srikanta Pal, Alireza Alaei, Umapada Pal, and Michael Blumenstein. Performance of an off-line signature verification method based on texture features on a large indic-script signature dataset. In *2016 12th IAPR workshop on document analysis systems (DAS)*, pages 72–77. IEEE, 2016.

[7] Ishanikathuria. Handwritten signature datasets. https://www.kaggle.com/datasets/ishanikathuria/handwritten-signature-datasets.

CSE4214_Paper

| 6 | researchr.org
Internet Source | 1% |

| 7 | export.arxiv.org
Internet Source | 1% |

| 8 | Submitted to Institute of Technology, Nirma University
Student Paper | 1% |

| 9 | Submitted to The University of Wolverhampton
Student Paper | 1% |

| 10 | Submitted to Yeditepe University
Student Paper | <1% |

| 11 | www.mdpi.com
Internet Source | <1% |

| 12 | "International Conference on Innovative Computing and Communications", Springer Science and Business Media LLC, 2020
Publication | <1% |

| 13 | Ishu Priya, Nisha Chaurasia. "Estimating the Effectiveness of CNN and Fuzzy Logic for Signature Verification", 2023 Third International Conference on Secure Cyber Computing and Communication (ICSCCC), 2023
Publication | <1% |

14   Yusnur Muhtar, Mahpirat Muhammat, Nurbiya Yadikar, Alimjan Aysa, Kurban Ubul. "FC-ResNet: A Multilingual Handwritten Signature Verification Model Using an Improved ResNet with CBAM", Applied Sciences, 2023
Publication

<1 %

15   dokumen.pub
Internet Source

<1 %

16   Alona Levy, Ben Nassi, Yuval Elovici, Erez Shmueli. "Handwritten Signature Verification Using Wrist-Worn Devices", Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, 2018
Publication

<1 %

17   Jennifer B. Rattray, Ryan J. Lowhorn, Ryan Walden, Pedro Márquez-Zacarías et al. " Machine learning identification of strains from colony image data ", Cold Spring Harbor Laboratory, 2023
Publication

<1 %

18   www.ncbi.nlm.nih.gov
Internet Source

<1 %

Exclude quotes   Off   Exclude matches   Off
Exclude bibliography   Off