International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 11 Issue: 04 | Apr 2024 www.irjet.net p-ISSN: 2395-0072

Signature Forgery Detection Using Deep Learning

P. BHUVANESWARI[1], K. MELVIN CHRISTOPHER[2], V. RISHI MAHESH RAJ[3], M. AJITHKUMAR[4]

[1234] *Dept. of Computer Science and Engineering, Government College of Engineering Srirangam, Tamilnadu, India*

Abstract – *Digital signatures are widely adopted by organizations, both public and private, in recent times due to their legal validity and ease of handling and storage. They find extensive usage in e-commerce websites for customer authentication during deliveries, bank procedures, government organizations, and various other businesses. Governments also utilize digital signatures for contract signing and document verification. However, with advancements in Information Technology (IT), there are both advantages and disadvantages. While digital signatures offer convenience, security, and cost savings, they also pose risks, such as potential forgery or manipulation. To address the risk of signature forgery, researchers are exploring deep learning algorithms like VGG16. These algorithms analyze signature data to differentiate between genuine and fake signatures by learning patterns and features from a dataset. By training and testing these algorithms on diverse signature samples, researchers aim to develop robust systems for detecting and mitigating signature forgery attempts. In summary, digital signatures play a vital role in modern organizational operations, offering benefits like legal validity, convenience, and enhanced security. However, addressing potential risks, such as forgery, requires ongoing research and technological advancements, including the application of deep learning algorithms like VGG16.*

Key Words: Security risks, forgery, manipulation, deep learning algorithms, VGG16.

1. INTRODUCTION

The handwritten signature stands as a critical biometric trait used for identity verification across legal, financial, and administrative domains [1], [2]. Manual authentication processes can be both time-consuming and prone to errors. Recent advancements in deep learning and computer vision have opened up avenues for more accurate and efficient automated signature recognition systems. These systems hold potential applications in sectors such as banking, law enforcement, and governmental organizations. However, despite their promise, they encounter challenges, particularly in accurately detecting forged signatures due to variations in styles, pen pressure, and angles. To tackle this issue, recent research has turned to Convolutional Neural Networks (CNNs), achieving high levels of accuracy in signature recognition, reaching up to 98.8%, and forgery detection, up to 89% [3], [4]. Architectures like GoogLeNet's Inception-v1 and Inception-v3, employing CNN models, have also shown promise, with validation rates of 83% and 75%, respectively [5]. This study utilizes CNNs to enhance the accuracy and reliability of the proposed signature recognition system. The primary objective is to develop a deep learning-based system capable of not only identifying genuine signatures but also detecting forgeries, thereby reducing the need for manual intervention. This approach aims to save time and costs associated with traditional methods. To accomplish these objectives, the study focuses on assembling and preprocessing a comprehensive dataset of signatures [1]. Preprocessing steps include noise removal to facilitate the implementation of a deep learning architecture for signature recognition. Evaluation metrics such as accuracy, precision, recall, and F1 score are utilized, and the system's performance is benchmarked against other state-of-the-art methods. In summary, this study provides valuable insights into model performance, dataset requirements, and potential areas for improvement in the field of signature recognition. The findings underscore the significance of training on diverse datasets and emphasize the capabilities of deep learning approaches.

1.1 RELATED WORK

A. In the field of handwritten signature identification, researchers frequently employed the ResNet architecture, as discussed in the study by Ishikawa et al. (2020). They utilized digital signal processing (DSP) for preprocessing tasks. ResNet architecture proved beneficial in overcoming limitations encountered with Convolutional Neural Networks (CNNs), particularly the vanishing gradient problem. This challenge was effectively addressed by ResNet signature data, highlighting the importance of ensuring data accuracy and consistency in such applications.

B. Rateria and Agarwal (2018) introduced a novel approach in their paper on handwritten signature authentication. They combined a traditional Convolutional Neural Network (CNN) with a Siamese neural network to authenticate handwritten signatures. Two configurations were employed for detecting handwritten signatures in their study. The first configuration acted as a feature extractor, crucial for discerning the authenticity of a signature. The second configuration functioned as a classifier, utilizing a Siamese neural network. This innovative setup involved the use of twin identical networks within the Siamese architecture, representing a pioneering effort in utilizing dual networks to extract features and distinguish between authentic and forged signatures.

C. In 2021, Ghosh explored the utilization of Recurrent Neural Networks (RNNs) for authentic signature detection, leveraging various deep learning techniques for image classification. The study involved the extraction of local features from handwritten signatures, followed by the generation of feature maps used in two types of RNN models: Long Short-Term Memory (LSTM) and Bidirectional Long Short-Term Memory (BiLSTM). The research demonstrated that RNNs outperformed other state-of-the-art models, typically based on Convolutional Neural Networks (CNNs), underscoring the efficacy of RNNs in this domain.

2. PROPOSED SYSTEM

Our proposed system aims to develop a robust and efficient solution for fake signature detection using advanced techniques. It leverages the strengths of pretrained visual geometry group (VGG) for classify the signature is real or fake.

2.1 Data Gathering:

For the Signature verification, data was gathered from the Kaggle website. The signs are in English and contain both real and fake handwritten signatures. The Model will be trained on 2 classes, with another 2 classes set aside for testing. It's public source dataset for handwritten signature authentication stems from the Kaggle website and complies with all General Data Protection Regulation (GDPR).

2.2 Data-Pre-processing:

A. Resizing:

Resizing all images to a fixed size (224x224 pixels) is crucial in the VGG Net neural architecture to ensure consistency in input dimensions. When images are resized to fit into the model's correct dimensions, there is less distortion and deformation in the image. This preservation of the image's original proportions ensures that all features are captured accurately, which is particularly critical for tasks such as handwritten signature detection.

B. Grayscale conversion:

The next step involves converting the images to grayscale. This process simplifies the images by removing color information, which can aid in reducing computational complexity and potentially improve model performance, particularly in cases where color is not a significant factor for signature verification. The image preprocessing for signature verification begins with organizing the forged and genuine signatures data into training and testing sets. The dataset is randomly divided, with 80% allocated to the training set and the remaining 20% to the testing set.

Fig.1. The image contains Signature to Grayscale image

3. Implementation of VGG16 Model:

VGG16 is a convolutional neural network configuration used in various profound learning image classification whose architecture is shown in Figure 2. VGG16 is trained with Imagenet dataset which has 1000 classes with 10 million images. By applying transfer learning method to VGG16 pretrained model to verify the Signature of 60 different users is achieved. This model consists of 16 layers with 3x3 size filters which uses sequential model which means that all the layers are connected in sequence. At the end it has two fully connected layer followed by Output layer with Softmax activation having 60 outputs, each output activation represents one user signature. All the hidden layers used RELU activation function. When applying preprocessed images as input to the model, it produced 99% as accuracy whereas produced only 76% for unprocessed input images.

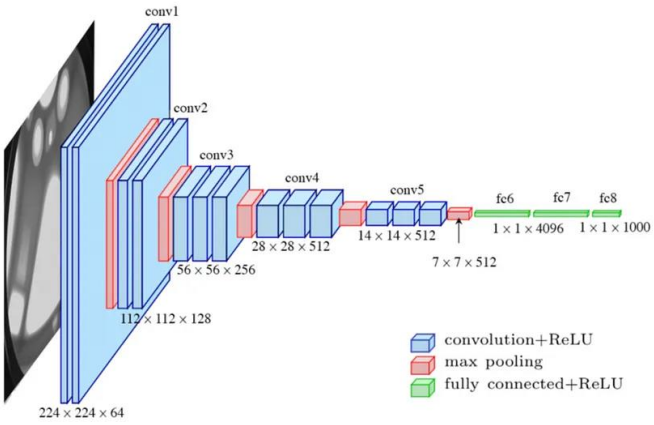

Fig 2. Architecture of VGG16 Model

3.1 Convolutional layers:

In the VGGNet neural network, the convolutional layer is crucial for extracting features and reducing the dimensionality of handwritten signature images. It uses filters to capture important characteristics from the input images, transforming them into feature vectors. Conv2D layers allow movement in two directions, facilitating this process. The case study employs 3x3 filters, starting with 64 filters in the first layers and progressively increasing.

Element-wise operations with these filters extract key features. The Add function combines layers, and after convolution, the output is flattened for further processing. The network consists of 13 convolutional layers and 3 fully connected layers. Filter sizes vary across layers, with 64-bit filters initially, followed by 128-bit and 256-bit filters, and finally 512-bit filters in subsequent layers. Figure 3 outlines the layers used in the VGG16 model.

Layer (type)	Output Shape	Param #
input_7 (InputLayer)	[(None, 224, 224, 3)]	0
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36928
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73856
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147584
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295168
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590080
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590080
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1180160
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2359808
block4_conv3 (Conv2D)	(None, 28, 28, 512)	2359808
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2359808
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0
sequential_6 (Sequential)	(None, 2)	6423298

Fig 3. The layers in VGG16 model

3.2 Activation layers:

The Activation layer introduces nonlinearity into the output of neural network neurons. In this case study, the Rectified Linear Unit (ReLu) activation function is utilized. ReLu returns zero for negative inputs and the input itself for positive inputs. This function is chosen to linearly modify image data, aiming to enhance linear separability. Additionally, ReLu helps address the vanishing gradient problem present in other activation functions. The equation describing the ReLu activation function is:

$f(x)=\max(0,x)$ where $f(x)$ represents the output of the ReLu function and x is the input to the function.

3.3 Pooling layers:

Pooling layers play a vital role in reducing the size of images while preserving essential features in handwritten signature images. They also aid in minimizing the learning parameters of the network and preventing overfitting while retaining important properties. Unlike convolutional layers, which maintain detailed information such as coordinates, pooling layers ensure that changes resulting from operations like resizing, shuffling, or rotation do not significantly affect the feature map. In this case study, maximum pooling was used, extracting the maximum value from each patch in the feature map. Specifically, four max-pooling layers were employed in the authentication of handwritten signatures.

3.4 Dense layer (Fully connected layer):

In this case study, the fully connected layer, also known as a dense layer, establishes connections with all preceding levels in the neural network. Its activation function involves key parameters such as weights and biases. Both Softmax and Relu activation functions are employed. Softmax is chosen for its suitability in scenarios with multiple class labels, as seen in this research. The input format for a Dense layer remains consistent at (batch size, input dimension), while the output format is (batch size, units). A total of 3 fully connected layers are utilized in this case study.

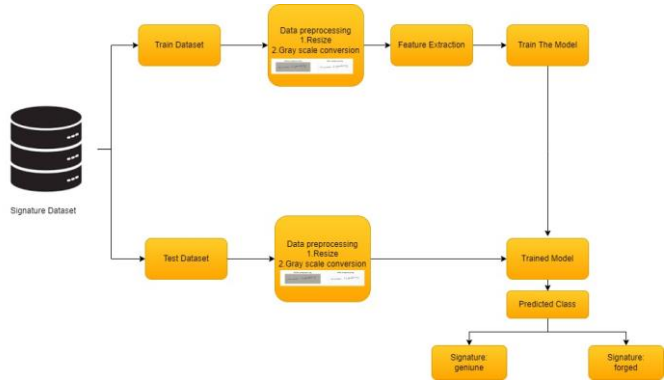

Fig 4. Architecture Diagram.

The dataset comprises 2689 images, of which 2050 were allocated for training the model and 604 for testing the VGGNet model. Thus, in percentage terms, 80% of the images were utilized for training and 20% for testing. Below are the hyperparameters employed for optimizing the model.

Adam Optimizers:

In deep neural networks, ADAgrad and RMSProp serve as alternatives to stochastic gradient descent. These optimization techniques effectively address noise present in patches of handwritten signature images. Backpropagation is employed to facilitate the learning process of the model, adjusting its weights using learning parameters (alpha).

Callbacks:

Callbacks are utilized effectively in this case study to enhance the training process of a VGG16 Model. Their implementation serves several crucial purposes, including smoothing the learning curve and accelerating the training process significantly. Additionally, callbacks contribute to preventing overfitting in the model. These benefits highlight

the importance of employing callbacks in neural networks. Early stopping is a technique implemented to prevent overfitting by specifying a large number of epochs in the neural network training process and halting the training when there is no improvement in the model's performance. In this study, early stopping is based on the validation loss parameter. If the validation loss fails to improve over consecutive epochs, training of the VGG16 model is terminated. ReduceLR On Plateau is a method used to adjust the learning rate of the VGGNet neural network when the accuracy of the model is not increasing.

4. RESULTS:

The proposed model significantly enhances the performance of the handwritten signature verification system. It is structured upon the VGG16 architecture and employs a pre-trained base model with the last few layers made trainable. During training, the model achieves impressive accuracy scores of 99.78% for training and 99.75% for validation, with a test accuracy of 98.96%. The model demonstrates a false acceptance rate (FAR) of 0.0, indicating precise identification of genuine signatures, and a false rejection rate (FRR) of 2.77%, showcasing effective detection of impostor signatures. Its precision stands at 98.9%, while the F1 score attains 0.986, reflecting the model's strong overall performance

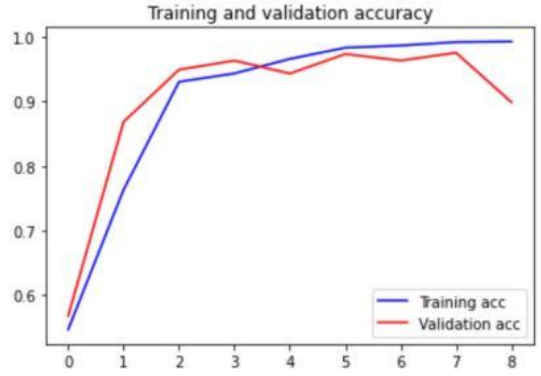

Fig.5. Training and Validation Accuracy of Proposed Model.

The outcomes showcase the model's capability in accurately categorizing and validating handwritten signatures. Its minimal false acceptance rate and acceptable false rejection rate bolster its reliability and credibility. The model exhibits considerable potential for real world utilization scenarios where precise signature verification is paramount. This signifies a notable progression and harbors the potential to fortify the dependability and resilience of signature verification systems. Fig. 4 illustrates the training and validation accuracy, while Fig. 5 showcases the training and validation loss, providing insights into the model's learning progress.

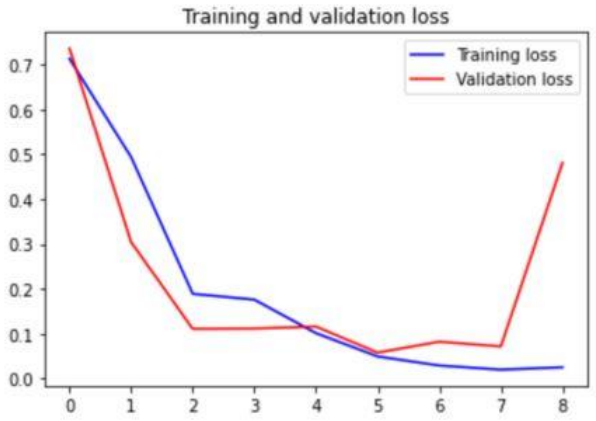

Fig.6. Training and Validation Loss of Proposed Model.

5. EXPERIMENTAL EVALUATION

5.1 Environment specifications:

The experimental configuration employed in the study. The research work takes place on an AMD Ryzen 7 CPU. Furthermore, the machine has 16GB of RAM and an Nvidia graphics card. The models are constructed using Python and are executed using deep learning frameworks such as Kera's and TensorFlow.

5.2 Dataset:

For the Signature verification, data was gathered from the Kaggle website. The signs are in English and contain both real and fake handwritten signatures. The Model will be trained on 2 classes, with another 2 classes set aside for testing. It's public source dataset for handwritten signature authentication stems from the Kaggle website and complies with all General Data Protection Regulation (GDPR).

6. CONCLUSIONS

The conclusion of this study underscores advancements in handwritten signature recognition, aiming to develop a deep learning-based system capable of discerning genuine from forged signatures. The project involved the collection and preprocessing of a dataset, implementation of the VGG16 model with transfer learning for signature recognition, and assessment of the system's performance through diverse metrics. The primary findings underscore the significance of training on larger and more diverse datasets to enhance robustness and generalization capabilities. The models were developed and trained on a merged dataset. Compared to Gupta Y et al.'s VGG16 model, these models exhibited superior accuracy, achieving 98.96% accuracy on the collected dataset Future endeavors should explore the utilization of local machine setups, broaden the

dataset to encompass a wider array of signature styles, and augment the dataset to bolster the model's robustness. Furthermore, the development of a user-friendly interface for the system would augment its accessibility and usability. Addressing these areas of enhancement would optimize the system's efficacy, adaptability, and user engagement, thereby propelling advancements in automated signature recognition technology across various applications.

7. REFERENCES

[1] Bharkav Rajyagor, Rajinish Rakhlia Hand Written character reconition using Deep Learning, International journal of recent technology and engineering (IJRTE) ISSN:2277-3878.

[2] Cherri Ishikawa; Jeff Allen U.Marasigan Cloud-based signature validation using CNN inception-Resnet architecture, IEEE 12th International conference on Humanoid, 2020.

[3] F. Noor, A. E. Mohamed, F. A. Ahmed, and S. K. Taha, "Offline handwritten signa ture recognition using convolutional neural network approach," in 2020International Conference on Computing, Networking, Telecommunications & Engineering Sciences Applications (CoNTESA), pp. 51–57, IEEE, 2020.

[4] J. A. Lopes, B. Baptista, N. Lavado, and M. Mendes, "Offline handwritten signature verification using deep neural networks," Energies, vol. 15, no. 20, p. 7611, 2022.

[5] J. Poddar, V. Parikh, and S. K. Bharti, "Offline signature recognition and forgery detection using deep learning," Procedia Computer Science, vol. 170, pp. 610–617, 2020.

[6] O. Tarek and A. Atia, "Forensic handwritten signature identification using deep learning," in 2022 IEEE 9th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT), pp. 185–190, IEEE, 2022.

[7] P.William Implementation of Hand Written based Signature Verification Technology using Deep Learning, International conference on intelligent engineering and management,2023 IEEE.

[8] S.Bonde, P. Narwade, and R. Sawant, "Offline signature verification using convo lutional neural network," in 2020 6th International Conference on Signal Processing and Communication (ICSC), pp. 119–127, IEEE, 2020.

[9] T. Venkat Narayana Rao, R. Balasubramanian, and K. S. Seshan, Real-Time Handwritten Signature Verification using CNN and Siamese Network, International Conference on Computing, Communication, and Intelligent Systems (ICCCIS), IEEE-2019.

[10] Y. Gupta, S. Kulkarni, and P. Jain, "Handwritten signature verification using trans fer learning and data augmentation," in Proceedings of International Conference on Intelligent Cyber-Physical Systems: ICPS 2021, pp. 233–245, Springer, 2022.