Signature Forgery Detection Using Convolutional Neural Networks

BALUSU REVANTH
Computer Science Engineering
ABV-IIITM
Gwalior, India

PATHLAVATH SUDEENDRA
Computer Science Engineering
ABV-IIITM
Gwalior, India

SAI REDDY RAJASHEKAR REDDY
Computer Science Engineering
ABV-IIITM
Gwalior, India

Prof. SHASHIKALA TAPASWI
Computer Science Engineering
ABV-IIITM
Gwalior, India

***Abstract*—Handwritten signature verification is a critical task in various applications, including financial transactions, document authentication, and identity verification. There are two kinds of signature verification: online and offline. Online signatures are obtained in real-time using digital devices like tablets, incorporating dynamic elements to enhance security and enable biometric authentication. Conversely, offline signatures are handwritten on physical mediums like paper and lack dynamic data, yet they continue to be widely accepted for legal and traditional purposes. Offline signatures do not provide real-time verification or biometric features. Deep learning models, such as convolutional neural networks (CNNs), have shown remarkable effectiveness in capturing intricate patterns and features in signature images, enabling accurate differentiation between genuine and forged signatures in offline verification. To further enhance the robustness and generalization capabilities of CNNs, various techniques like data augmentation and regularization are explored. With the utilization of deep learning models, handwritten signature verification systems can provide reliable and trustworthy results, ensuring the authenticity and integrity of signatures in critical applications. In this paper, a CNN model for handwritten signature verification is introduced. To improve the results of the verification process, the model includes data augmentation and regularization approaches.**

***Index Terms*—Signature Verification, Forgery Detection, Deep Learning, Convolutional Neural Networks.**

I. Introduction

Handwritten signature verification is an essential and ubiquitous task in modern society, finding extensive applications in financial transactions, document authentication, and identity verification. The process of signature verification involves confirming the authenticity of a signature, ensuring it matches the genuine signature of the individual purportedly signing the document. To achieve this, there are two primary methods of signature verification: online and offline.

Online signature verification employs digital devices, such as tablets, to capture signatures in real-time. This approach incorporates dynamic elements, such as the speed and pressure of the signature, to enhance security and enable biometric authentication. In contrast, offline signatures are handwritten on physical mediums, like paper, and lack dynamic data. Despite the absence of real-time verification and biometric features, offline signatures continue to be widely accepted for legal and traditional purposes.

In recent years, deep learning models, particularly convolutional neural networks (CNNs), have demonstrated remarkable efficacy in the domain of offline signature verification. CNNs excel in capturing intricate patterns and features present in signature images, making them adept at distinguishing between genuine and forged signatures. Their ability to analyze spatial information and extract discriminative features from signature images significantly contributes to the accuracy of the verification process.

A. Problem/Motivation

Traditional methods of signature verification are often time-consuming, subjective, and prone to errors. Leveraging CNNs for signature forgery detection presents an opportunity to automate and enhance the accuracy of this process. By training the CNN on a large dataset of genuine and forged signatures, the model can learn to automatically extract intricate features and patterns that distinguish authentic signatures from forgeries.

By addressing this challenge through deep learning, the project aims to provide a robust and efficient solution to identify fraudulent signatures, thereby bolstering the security and reliability of digital authentication systems. The potential impact of such a project spans across various industries, including banking, legal, and government sectors, where the ability to reliably detect forged signatures is crucial for preventing financial losses, legal disputes, and maintaining trust in digital transactions.

The primary goal of this project is twofold: to enhance the security of digital transactions by preventing fraudulent activities and to bolster the reliability of identity verification processes in legal and financial domains. By automating the detection of forged signatures, we aspire to save valuable time and resources while fortifying the trust individuals and institutions place in digital signatures.

B. Objectives

- Designing a robust CNN architecture: The success of any CNN-based forgery detection system hinges on the design of an effective neural network. We will explore various network architectures, optimizing them to achieve the highest possible accuracy in signature classification.
- Dataset curation: To train and evaluate our model, a diverse and comprehensive dataset of genuine and forged signatures will be collected and curated. This dataset will be carefully balanced to ensure unbiased learning and robust generalization.
- Preprocessing and data augmentation: Image preprocessing techniques will be applied to enhance the quality and consistency of the signature images. Additionally, data augmentation methods will be employed to expand the dataset, making the model more robust to variations in signature styles and quality.
- Training and validation: The CNN model will undergo extensive training on the curated dataset, iteratively fine-tuning its parameters to optimize performance. Rigorous validation procedures will be employed to assess the model's ability to generalize on unseen data.
- Evaluation and benchmarking: The forged signature detection system will be evaluated against various metrics and benchmarked against state-of-the-art forgery detection techniques to establish its superiority and efficacy.
- Deployment and integration: Once the model achieves the desired level of accuracy and reliability, we will explore methods to seamlessly integrate it into existing signature verification systems, allowing for real-world deployment and application.

II. Literature Review

A. Background

1) Signature Forgery: Signature forgery is the act of imitating or replicating someone else's signature with the intention of deceiving others. It can be categorized as random forgeries (attempting to imitate a signature without prior knowledge) and skilled forgeries (attempts to imitate a specific signature with knowledge of the original signer's style).

2) Convolutional Neural Networks (CNNs): CNNs (Convolutional Neural Networks) are powerful deep learning models specifically designed for visual data analysis. They employ multiple layers, including convolutional layers, which automatically extract hierarchical features from images, enabling them to tackle intricate recognition tasks. These networks have revolutionized computer vision applications, such as image classification, object detection, and segmentation. By leveraging shared weights and local connections, CNNs efficiently capture spatial patterns within images, making them highly effective in various real-world scenarios. Their ability to learn from data and adapt to different visual domains makes them indispensable tools in the field of artificial intelligence and image processing.

B. Related Works

1) Traditional Methods for Signature Forgery Detection: In the past, signature forgery detection mainly relied on handcrafted features and traditional machine learning techniques. Approaches like Support Vector Machines (SVM), Random Forest, and Hidden Markov Models (HMM) were commonly used. While these methods showed promising results in some scenarios, they often lacked generalization capabilities and struggled with complex forgeries.

2) Deep Learning Approaches for Signature Forgery Detection: With the emergence of deep learning, researchers began exploring CNN-based methods for signature forgery detection due to their ability to automatically learn discriminative features. Some relevant works in this area include:

"Signature Verification using Siamese CNNs" by Bromley et al. (1993):
This early work introduced the concept of Siamese CNNs for signature verification. Siamese networks utilize two parallel CNNs to compare two input signatures and compute a similarity score. This approach proved effective for signature verification tasks.

"Offline Signature Verification using CNNs and Long Short-Term Memory (LSTM)" by Santos et al. (2017):
In this study, a hybrid architecture combining CNNs and LSTM was proposed for offline signature verification. CNNs were used to extract local features from signature images, while LSTM helped capture sequential patterns in signature strokes.

"ForgeryNet: A CNN for Universal Forgery Detection in Handwritten Documents" by Cozzolino et al. (2019):
ForgeryNet is a CNN-based architecture specifically designed for universal forgery detection in various handwritten documents, including signatures. The model leverages adversarial training to improve its robustness against skilled forgeries.

"Signature Forgery Detection using Attention-based CNNs" by Zhang et al. (2021):
This work introduced attention mechanisms into CNNs for signature forgery detection. The attention mechanism allows the network to focus on important regions of the signature, improving performance and interpretability.

III. Methodology

A. Data Acquisition

The dataset gathering process for the Signature Forgery Detection project involves obtaining the CEDAR-Dataset from Kaggle. This dataset contains original and forged signatures, providing a diverse range of samples for training a reliable forgery detection model. The genuine signatures were likely collected with consent from individuals, while the forged signatures were created by experts using various techniques. The dataset's quality and diversity are crucial to ensure the model can detect different forgery types accurately. The dataset

consists of 2640 signature images, categorized as genuine or forged. Genuine signatures are legitimate, while forged signatures are created to imitate others. The dataset contains signatures of different sizes and aspect ratios. By examining the dataset, we can determine the most common image dimensions and consider strategies for resizing or normalizing them to a standard size for better model performance. The images are in the .png format and consist of three channels: red, green, and blue (RGB).

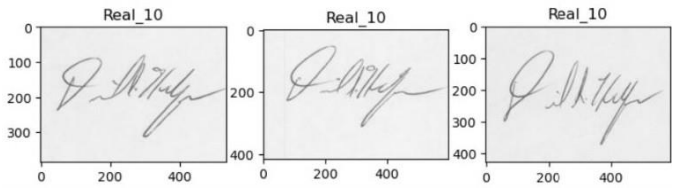

Fig. 1. Example of Original Signature

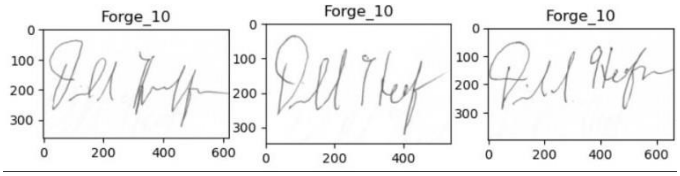

Fig. 2. Example of Forged Signature

B. Data Preprocessing

Data preprocessing for the signature forgery detection project involves several important steps.

1) Data resizing: The images are resized to dimensions of 224x224. This resizing ensures that all images have a consistent size, which is essential for machine learning models that require fixed-size inputs. By standardizing the dimensions, we eliminate potential issues that may arise from varying image sizes.

2) RGB to Greyscale: .The RGB images are converted to grayscale. This conversion reduces the colour channels from three (red, green, and blue) to one (gray). Since signature forgery detection relies more on shape and texture rather than colour information, converting to grayscale simplifies the data while retaining the important features needed for accurate detection.

C. Data Augmentation

Data augmentation techniques are applied specifically to the training data. Data augmentation involves applying various transformations to artificially increase the size and diversity of the training dataset. For the signature forgery detection project, the training data is augmented with techniques such as rotation, shear, zoom, and horizontal flip. These augmentation techniques are applied only to the training data, not the validation or testing data. By augmenting the training data, the model is exposed to a wider range of samples, making it more robust and less prone to overfitting. It learns to generalize well and adapt to different variations and angles that may occur in real-world signature forgery scenarios.

D. Model Architecture

The CNN architecture plays a crucial role in the model's performance. Designing an appropriate network architecture involves selecting the number and type of layers, their connectivity, and their sizes. Typically, a CNN model for signature forgery detection consists of multiple convolutional layers to extract relevant features from the input images. These convolutional layers are often followed by pooling layers to reduce spatial dimensions and introduce translation invariance. Additional layers like batch normalization and dropout can be included to improve model generalization and prevent overfitting. Fully connected layers are responsible for the final classification. The specific architecture of the CNN can vary based on the complexity of the problem and available computational resources.

1) Convolutional Layer: A convolutional layer is a fundamental building block of Convolutional Neural Networks (CNNs), which are primarily used for image recognition tasks but can also be applied to other types of data. The convolutional layer performs convolutional operations on the input data. The key idea behind this layer is to detect local patterns or features in the input data by sliding a small filter (also known as a kernel) over the input and computing dot products between the filter and local regions of the input. The output of this operation is called a feature map, and it represents the presence of specific features in the input data.

2) Batch Normalization: Batch Normalization is a technique used to improve the training and performance of deep neural networks. It addresses the problem of internal covariate shift, which occurs when the distribution of each layer's input changes during training, making it harder to train the network. Batch normalization normalizes the input of each layer by standardizing it to have zero mean and unit variance. This helps to stabilize and accelerate the training process, allows for higher learning rates, and can act as a regularizer, reducing the need for other regularization techniques like dropout.

3) Maxpooling: Maxpooling is a downsampling operation typically used in CNNs to reduce the spatial dimensions of the feature maps while retaining the most important information. It works by dividing the feature map into non-overlapping regions and selecting the maximum value from each region. This helps to reduce the computational complexity of the network, make it less sensitive to small variations in input, and introduce a degree of translation invariance since the maximum value represents the most prominent feature in each region.

4) Dropout: Dropout is a regularization technique used to prevent overfitting in neural networks. During training, dropout randomly sets a fraction of the neurons' outputs to zero with a certain probability (dropout rate). This effectively removes those neurons from the network for that particular forward and backward pass. By doing this, dropout prevents the network from relying too much on any particular set of neurons, forcing

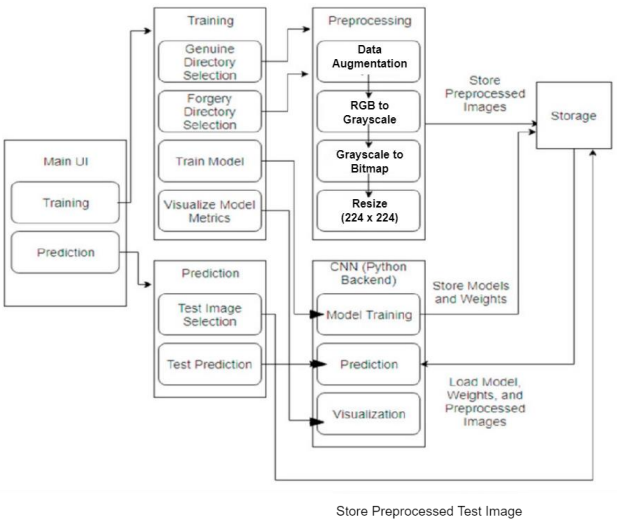

Fig. 3. Architecture Diagram.

it to learn more robust and general features. During inference (testing), dropout is usually turned off, and the entire network is used for making predictions.

5) Flatten: Flatten is an operation that converts multidimensional data (e.g., a 2D image or a 3D volume) into a 1D vector. In CNNs, the convolutional and max-pooling layers produce 3D feature maps, and before passing the data to fully connected (dense) layers, it needs to be flattened into a 1D vector.

6) Dense (Fully Connected) Layer: A dense layer is a standard layer in a neural network where each neuron is connected to every neuron in the previous layer. In other words, all the neurons in a dense layer receive input from all the output neurons of the previous layer. These layers are typically used at the end of a CNN or other types of networks to map the extracted features to the final output (e.g., classification scores).

Operation	Formula
Convolution	$z^l = h^{l-1} * W^l$
Max Pooling	$h^l{}_{xy} = \max_{i=0..s, j=0..s} h^{l-1}(x+i)(y+j)$
Fully-connected layer	$z_l = W_l * h_{l-1}$
ReLu(Rectifier)	$\mathrm{ReLU}(z_i) = \max(0, z_i)$
Softmax	$\mathrm{softmax}(z_i) = e^{z_i} / \sum_j e^{z_j}$

Fig. 4. List of formulas for the operations in the CNN

E. Early stopping and Learning Rate Reduction

Early stopping and learning rate reduction are crucial techniques in signature forgery detection to improve model performance and efficiency. Early stopping is employed to prevent

Layer(type)	Output Shape	Param#
conv2d (Conv2D)	(None, 222, 222, 32)	320
batch$_n$*ormalization*(*BatchNormalization*)	(None, 222, 222, 32)	128
max$_p$*ooling2d*(*MaxPooling2D*)	(None, 111, 111, 32)	0
dropout (Dropout)	(None, 111, 111, 32)	0
conv2d$_1$(*Conv2D*)	(None, 109, 109, 64)	18496
batch$_n$*ormalization*$_1$(*BatchNormalization*)	(None, 109, 109, 64)	256
max$_p$*ooling2d*$_1$(*MaxPooling2D*)	(None, 54, 54, 64)	0
dropout$_1$(*Dropout*)	(None, 54, 54, 64)	0
conv2d$_2$(*Conv2D*)	(None, 52, 52, 128)	73856
batch$_n$*ormalization*$_2$(*BatchNormalization*)	(None, 52, 52, 128)	512
max$_p$*ooling2d*$_2$(*MaxPooling2D*)	(None, 26, 26, 128)	0
dropout$_2$(*Dropout*)	(None, 26, 26, 128)	0
conv2d$_3$(*Conv2D*)	(None, 24, 24, 256)	295168
batch$_n$*ormalization*$_3$(*BatchNormalization*)	(None, 24, 24, 256)	1024
max$_p$*ooling2d*$_3$(*MaxPooling2D*)	(None, 12, 12, 256)	0
dropout$_3$(*Dropout*)	(None, 12, 12, 256)	0
conv2d$_4$(*Conv2D*)	(None, 10, 10, 256)	590080
batch$_n$*ormalization*$_4$(*BatchNormalization*)	(None, 10, 10, 256)	1024
max$_p$*ooling2d*$_4$(*MaxPooling2D*)	(None, 5, 5, 256)	0
dropout$_4$(*Dropout*)	(None, 5, 5, 256)	0
conv2d$_5$(*Conv2D*)	(None, 3, 3, 512)	1180160
batch$_n$*ormalization*$_5$(*BatchNormalization*)	(None, 3, 3, 512)	2048
max$_p$*ooling2d*$_5$(*MaxPooling2D*)	(None, 1, 1, 512)	0
dropout$_5$(*Dropout*)	(None, 1, 1, 512)	0
flatten (Flatten)	(None, 512)	0
dense (Dense)	(None, 256)	131328
batch$_n$*ormalization*$_6$(*BatchNormalization*)	(None, 256)	1024
dropout$_6$(*Dropout*)	(None, 256)	0
dense$_1$(*Dense*)	(None, 2)	514

Fig. 5. Model Architechure

overfitting and enhance generalization. It involves monitoring the model's performance on a validation set during training. If the performance does not improve for a certain number of epochs (controlled by a patience value), the training process is stopped. Early stopping prevents the model from memorizing the training data and ensures it learns useful patterns.

Learning rate reduction is used to optimize the training process and model convergence. The learning rate determines the step size taken during parameter updates. By dynamically adjusting the learning rate, typically when validation performance plateaus or the loss function stops decreasing, the model can fine-tune its parameters more effectively. This adjustment aids in reaching better local or global minima during optimization.

IV. Results

A. Working

B. Performance Comparision

To assess the performance of the CNN model, two additional variants are compared: the CNN model with data augmentation and the CNN model with data augmentation and early stopping with learning rate reduction.

1) CNN Model: The baseline CNN model achieved a respectable level of accuracy during training. However, as is typical with deep learning models, it exhibited a risk of overfitting on the training data, leading to a comparatively lower accuracy on the validation dataset. The model's loss function showed a decreasing trend during the initial epochs, but the validation loss indicated potential overfitting, as it started to increase after a certain point.

2) CNN Model with Data Augmentation: To mitigate overfitting and improve generalization, data augmentation techniques were incorporated during training. By augmenting the training dataset with various transformations, such as rotation, flipping, and scaling, the model gained exposure to a more diverse range of signature variations. Consequently, the CNN model with data augmentation exhibited a noticeable improvement in accuracy on the validation set. The loss function demonstrated a steadier convergence, indicating better generalization capabilities.
3) CNN Model with Data Augmentation and Early Stopping with Learning Rate Reduction: To further enhance the model's performance and prevent overfitting, early stopping and learning rate reduction techniques were introduced. Early stopping allowed the model to halt training when the validation loss stopped improving, preventing excessive training and potential overfitting. Additionally, learning rate reduction dynamically adjusted the learning rate during training, fine-tuning the model's optimization process.

The CNN model with data augmentation and early stopping with learning rate reduction showcased the most promising results. Its accuracy on both the training and validation datasets improved significantly compared to the baseline CNN model. Moreover, the model's loss function demonstrated a smoother convergence, indicating a robust learning process.

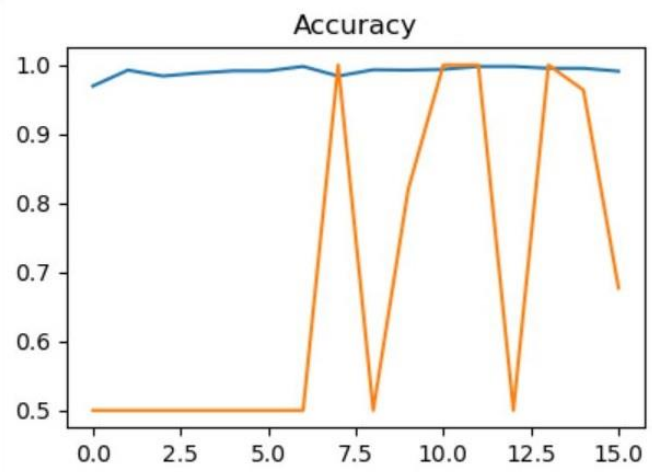

Fig. 6. Accuracy

We achieved a test accuracy of 99.4% with the train, validation, test split of ratio 70%, 15%, 15%.

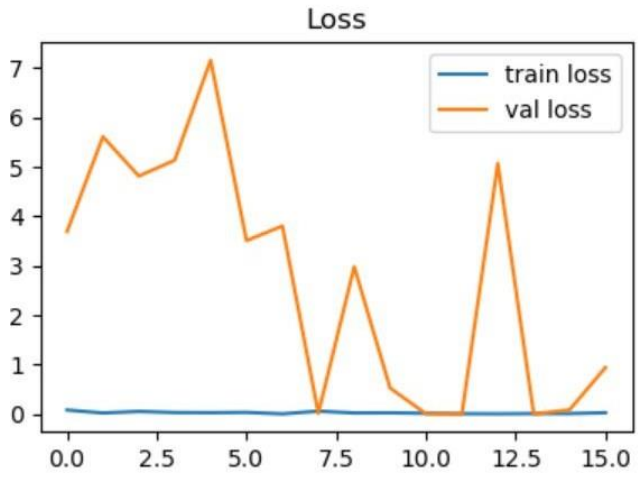

Fig. 7. Loss

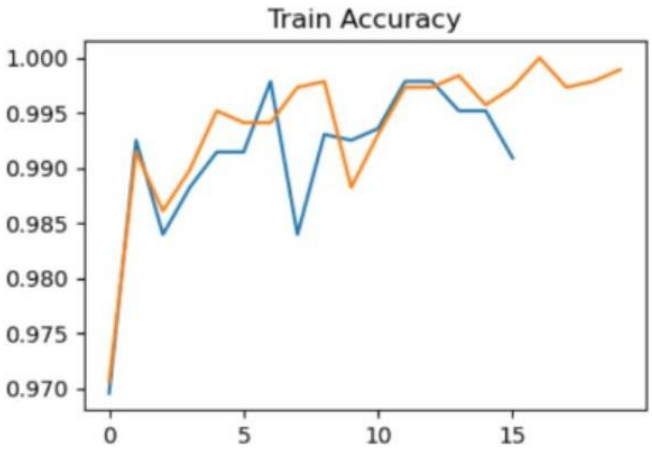

Fig. 8. Train Accuracy for both models

V. Conclusion

A. Limitations

- Limited Datasets: One of the primary challenges in training CNNs for signature forgery detection is the availability of large-scale, diverse, and annotated datasets. Collecting a comprehensive dataset of genuine signatures, as well as various types of forgeries, can be time-consuming and costly. The limited dataset may lead to overfitting, where the model may not generalize well to unseen or complex forgery patterns.
- Variations in Signature Styles: Signatures can exhibit significant variations even for the same individual, depending on factors such as writing instruments, writing surface, mood, and age. Moreover, signatures can differ substantially between different individuals. The high intra-class and inter-class variations make it challenging for CNNs to capture the essential discriminative features effectively.

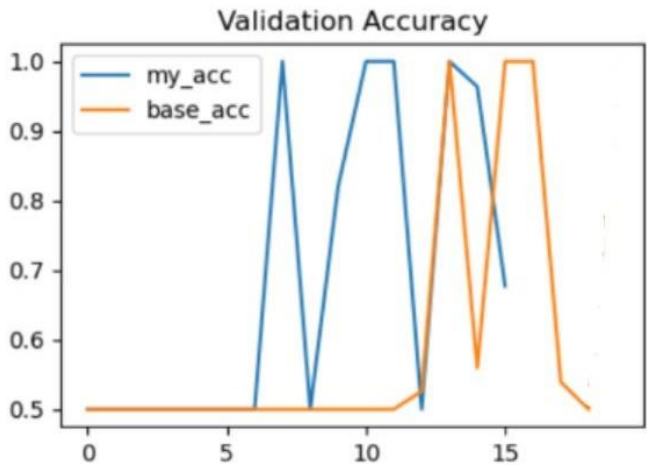

Fig. 9. Validation Accuracy for both models

- Skilled Forgeries: Detecting skilled forgeries is particularly challenging because forgers attempt to mimic the genuine signer's style closely. Skilled forgeries are often so convincing that even human experts may struggle to differentiate them from genuine signatures. CNNs must be able to identify subtle differences in stroke patterns, pressure, and shape to detect such forgeries accurately.
- Online vs. Offline Signatures: Signature forgery detection can be performed on both offline (scanned or photographed) and online (captured using a digital stylus) signatures. Online signatures carry temporal information about the signing process, whereas offline signatures lack this temporal context. Combining both types of signatures to create a robust CNN model is a non-trivial task

B. Future Scope

Improved Deep Learning Architectures: Continued research and development in deep learning architectures, specifically tailored for signature forgery detection, can lead to more effective models. Exploring novel CNN architectures, attention mechanisms, and transformer-based models may improve the accuracy and robustness of the detection process.

GANs for Data Augmentation: Generative Adversarial Networks (GANs) can be employed to augment the signature dataset. GANs can generate realistic synthetic signatures, both genuine and forged, to address the issue of limited training data. This approach can improve the model's ability to detect various types of forgeries.

Online Signature Verification: Extend the forgery detection to online signature verification, which involves capturing dynamic information during the signing process. Online signature verification can provide additional behavioral cues, making the forgery detection process more reliable.

Multi-Modal Fusion: Combining information from different sources, such as online and offline signatures, along with additional biometric modalities like keystroke dynamics or pressure sensitivity, can create a more robust and comprehensive forgery detection system.

REFERENCES

[1] Hafemann, L.G., Sabourin, R. and Oliveira, L.S., 2017. Learning features for offline handwritten signature verification using deep convolutional neural networks. Pattern Recognition, 70, pp.163-176.

[2] Li, C., Lin, F., Wang, Z., Yu, G., Yuan, L. and Wang, H., 2019, September. DeepHSV: User-independent offline signature verification using two-channel CNN. In 2019 International Conference on Document Analysis and Recognition (ICDAR) (pp. 166-171). IEEE.

[3] Mohapatra, R.K., Shaswat, K. and Kedia, S., 2019, November. Offline handwritten signature verification using CNN inspired by inception V1 architecture. In 2019 Fifth International Conference on Image Information Processing (ICIIP) (pp. 263-267). IEEE.

[4] Parcham, E., Ilbeygi, M. and Amini, M., 2021. CBCapsNet: A novel writer-independent offline signature verification model using a CNN-based architecture and capsule neural networks. Expert Systems with Applications, 185, p.115649.

[5] Tayeb, S., Pirouz, M., Cozzens, B., Huang, R., Jay, M., Khembunjong, K., Paliskara, S., Zhan, F., Zhang, M., Zhan, J. and Latifi, S., 2017, December. Toward data quality analytics in signature verification using a convolutional neural network. In 2017 IEEE international conference on big data (Big Data) (pp. 2644-2651). IEEE.

[6] Hafemann, L.G., Sabourin, R. and Oliveira, L.S., 2016, December. Analyzing features learned for offline signature verification using deep CNNs. In 2016 23rd international conference on pattern recognition (ICPR) (pp. 2989-2994). IEEE.

[7] Ishikawa, C., Marasigan, J.A.U. and Caya, M.V.C., 2020, December. Cloud-based Signature Validation Using CNN Inception-ResNet Architecture. In 2020 IEEE 12th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM) (pp. 1-6). IEEE.