

Unveiling the Future of Signature Verification: Deep Learning Insights

Veena V. G^{*1}, J. R. Jeba²

Submitted: 25/01/2024 Revised: 03/03/2024 Accepted: 11/03/2024

Abstract: A person's signature, which is usually affixed to documents as proof of approval or authority, is a unique handwritten representation of their identity. Signatures serve as a useful means of personal identification and are essential for the validation of transactions, contracts, and official documents. Strong signature verification techniques are necessary to counter the growing threat of forged signatures as the use of signatures in diverse areas increases. Forged signatures and fraudulently created copies significantly affect the security and authenticity of documents. This paper suggests a novel deep learning technique for offline signature verification for ease this concern. This novel method seeks to improve the model's capacity to extract specific details as well as high-level semantic information from signature images. A variety of real and fake signatures are included in the dataset, and thorough preprocessing and augmentation methods are used to guarantee successful model training. With 97.39% accuracy, 96.79% precision, 97% recall, and 97.20% F1-Score, the suggested model performs remarkably well. These findings demonstrate the efficiency of the deep learning-based technique in precisely confirming signatures and identifying suspected forged.

Keywords: Offline Signature Verification, Deep Learning, Genuine or Forged, Dual Path Network, Convolutional Neural Network

1. Introduction

The signature of a person is a handwritten transcription of their entire name or nickname, used as a symbol of identity. A document is typically written as evidence of identity and purpose. It bears the signature of an authorized individual claiming something as evidence. The most widely recognized personal characteristic in both social and legal contexts is Offline Signature Verification. Handwritten signatures have been widely accepted as a practical method of text verification and fraud protection for many years. A handwritten signature is required for an increasing number of transactions, particularly financial, official, and commercial ones. Because automatic verification takes longer to complete, it is challenging to verify every document. Consequently, there has been an exponential growth in the usage of biometrics such as fingerprints, iris scans, signatures, hand shapes, and Deoxyribonucleic Acid (DNA) for the security of financial and commercial documents over time [1]. To verify the authenticity of handwritten signatures, verification of signature is an essential component of identity authentication and document security. The two primary types of signature verification techniques are: online and offline. Using

scanned or digitalized images, static properties, such as the size, shape, and spatial arrangement of signature components, are analyzed in offline signature verification. When digital versions of physical documents are accessible, this technique is frequently used. On the other hand, online signature verification uses specialized equipment, such as tablets with styluses, to record dynamic characteristics of the signing process, such as stroke order, speed, and pressure. Both kinds of signature verification are essential. Both kinds of signature verification are essential for preventing fraud, ensuring the accuracy of legal documents, and promoting safe transactions across a range of industries. The decision between offline and online verification frequently comes down to the particular needs of the application and the state of the technology at hand [2]. Some samples of signatures are given in Figure 1.



Fig.1. Samples of signature

Two genuine signatures from the same signer can never

^{1,2}Department of Computer Applications

¹Research Scholar, Noorul Islam Centre for Higher Education, Kumaracoil, Tamil Nadu, India

²Associate Professor and Head, Noorul Islam Centre for Higher Education, Kumaracoil, Tamil Nadu, India

E-mail Id: ¹ Veena.V.G@outlook.com

² Jeba.J.R@outlook.com

* Corresponding Author Email: Veena.V.G@outlook.com

have the same geometric shape. Consequently, these modifications must be taken into account during the certification process. It is impossible to characterize the signatures in a unique way because of these modifications. This gives others the chance to replicate any signature. Forged signatures refer to signatures that have been copied through fraud [3]. Verification of signatures are made to identify and prevent different kinds of forgeries, guaranteeing the authenticity of signatures on documents. One popular kind is random forging, in which the forger tries to imitate a signature without being aware of the signature style of the real signer. A more careful replication is used in simulated forgeries, when the forger replicates the signer's stroke patterns by studying accessible samples. In contrast, traced forgery uses overlays or transparent sheets to closely resemble an authentic signature. In addition, lifted signatures, freehand attempts, and the use of forging instruments like stamps or tracing devices can all be used to create fake signatures in addition to physical manipulation of authentic signatures, disguised signatures, and multi-signer forgeries provide additional issues for verification systems. Cut-and-paste forgery modifies documents in this way. The detection and distinction of these forgeries is greatly aided by advanced technologies, such as deep learning and machine learning algorithms, which guarantee the durability of signature verification systems in preserving document security and validity [4]. Figure 2 shows some signature samples of genuine and forged.

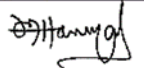
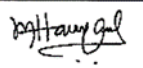
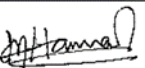

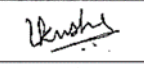

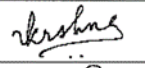
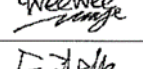
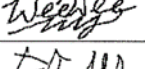

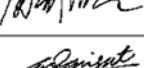
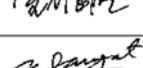
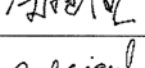

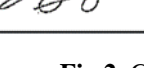
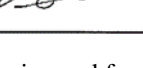
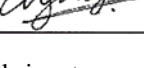

Genuine	Skilled forgery	Unskilled forgery	Random forgery
			
			
			
			
			

Fig.2. Genuine and forged signature samples

Traditional signature verification methods often faced challenges in handling the intricate variations and complexities inherent in signatures. Deep learning techniques have proven highly success rate in addressing these challenges. Deep learning models can automatically identify and extract features that are discriminative from scanned or digitalized signature images in the context of signature verification. These models are skilled at differentiating between genuine and forged signatures because of the hierarchical representations they record, which allow them to read out small variations in stroke patterns, spatial arrangements, and other distinguishing

features. So, a Deep Learning method for offline verification of signature is suggested in the proposed work.

The significant contribution of the suggested work includes:

- A novel model for the verification of signature based on deep learning.
- A novel model that effectively classifies different signature forgeries.
- Comparing the suggested model performance with existing methods.

The remaining portion of the paper are arranged as follows: In Section 2, a summary of literature is provided, highlighting areas that indicate a need for more investigation. In section three, the methodology is explained in depth. The fourth section goes into great detail about the results that the suggested strategy produced. Finally, a summary of the findings is included in Section five, which gives a conclusion to the paper.

2. Related Works

In order to determine the correlation between the pixels of various signatures, Narwade et al. [5] created an offline method for verifying signatures according to shape correspondence. An adaptive weighted combination of Euclidean distance and shape context distance was employed in this method. The SVM classifier was then used to determine whether the signature was authentic by using the computed distances as an input. Here obtained an accuracy of 89.58% by testing the suggested strategy on the GPDS synthetic signature data base.

The offline signatures were analyzed to extract global and local properties, and Sharif et al. [6] provided a structure for offline signature verification. Following a reduction of the extracted features using evolutionary algorithm feature selection approach, the remaining were given to an SVM classifier for validation. Three datasets—CEDAR, MCYT, and GPSS synthetic were used in the studies. The obtained rates of FRR, FAR and AER on the CEDAR dataset surpasses the current methods in machine learning for signature verification.

Masoudnia et al. [7] examined three loss functions for CNNs: hinge loss, Cauchy–Schwarz divergence, and cross-entropy providing offline signature verification with the use of multi representational learning. After that, these losses were incorporated to dynamic multi-loss functions based on the complementing properties. The collection of SVMs was trained using the multi-representation set in the proposed ensemble through several trials, each of which learnt a different representation for every input depending on the identification of signature. Based on the methods for signature verification, the mode of experiments was assessed on three datasets: the GPDS-Synthetic, UT-SIG,

and MCYT. The performance analysis of suggested algorithm was assessed based on accuracy metrics such as FAR, FRR, and AER.

By using images of the handwritings, Shariatmadari et al. [8] suggested a method on hierarchical one-class CNN to learn only real signatures that have distinct feature levels. This study approached signature verification as a problem because forgeries were not available for every user enrolled in an actual application situation. More accurate outcomes can also be obtained by constructing a network architecture hierarchy according to the coarse-to-fine concept in order to create a distinct structure in the image. While higher-level features can distinguish pen stroke quality to forecast forgeries from real signatures, lower-level characteristics enable the network to provide higher-quality visuals at the boundary region, highlighting similarities between genuine signatures. Two Latin databases and two Persian databases were used to test the system that was provided. In contrast to the current approaches the analyses generated by this method for the four signature databases were typically better and more accurate.

A deep CNN technique that uses a single known signature specimen for signature verification and forgery detection was introduced by Kao and Wen [9]. For extracting the desired features from the signature specimen, a method for extracting local features has been implemented. An evaluation of this approach using the ICDAR2011 Sig Comp dataset indicated 92.37% stated accuracy levels. The findings showed that increasing the size of the forged specimens can effectively improve network performance even when working with a single known sample.

A writer-independent handwritten signature verification model, Inverse Discriminative Networks (IDN) was suggested by Wei et al. [10]. Each of the four network streams in the suggested model has two pairs of signature samples. Two set of signature samples are provided: a reference pair and a test pair of convolutional feature extraction from signatures. The inverted grey reference sample of signature and the test signature sample are included in the first pair, which focuses on signature strokes.

Poddar et al. [11] introduced a deep learning technique for identifying authentic signatures and detecting forgeries. Following the detection of the signatures with the CNN and Crest-Trough methods, the SURF and Harris algorithms are utilized to detect signature forgeries. The range of accuracy reports for forgery detection and signature identification is 85–89% and 90–94%, respectively.

Convolutional Auto encoder (CAE) was utilized by Vorugunti et al. [12] to obtain features from online signatures, which were subsequently coupled with manually created feature. The input of the Depth-wise Separable CNN was this hybrid set of features. Compared to conventional

CNNs, DWSCNN used fewer training samples and parameters, which results in a lighter Online Signature Verification (OSV) framework. The obtained findings showed that the suggested framework performs better than the most advanced OSV techniques. For the first time, a mixed combination of features and few shots learning was taken into consideration in the suggested model. Gumusbas and Yildirim [13] assessed the effectiveness of a capsule network in the identification and confirmation of offline signatures. The proposed method used CEDAR database and achieved accuracy rates that were higher than those obtained using CNN, with results for 64×64 and 32×32 input resolutions reaching 91.8% and 92.6%, respectively.

Maergner et al. [14] offered a method to integrate a structural approach employing a statistical method that utilizes deep triplet networks based on graph edit distance for achieving significantly better signature verification. The definition of the Multiple Classifier System (MCS) was the combination of neural network-based and graph-based dissimilarity. On four different datasets, the suggested MCS system performs better than the separate GED and CNN systems, illustrating the complimentary qualities of structural and statistical models. The MCS method provides competitive results overcoming challenges like dataset-specific thresholds, suggesting its potential for signature verification across various datasets.

Jiaxin Lu et al. [15] included studies on the identification of authentic signatures by combining dynamic and static data through the utilization of deep learning and machine learning. To find more intelligible features for accurate signature identification, the aim of this work was to combine the dynamic elements of digital writing with the constant elements of conventional pen and paper writing. This research enhanced the classification accuracy of signature identification by concentrating on feature extraction and integrating the benefits of both static and dynamic features. The classification accuracy had increased than the previous studies on machine learning but number of words contained in each signature is very small which make the identification difficult.

P. Kiran et al. [16] suggested using a back propagation neural network architecture and image processing techniques to recognize offline signatures. Pre-processing signatures can involve image processing techniques such as filtering, RGB2Gray conversion, thresholding, altering, and canny edge detection followed by image scaling to reduce processing time. To obtain processed picture features, a back propagation neural network system having a fixed number of neurons and hidden layers was employed. Similar preparation procedures were used for feature extraction from data set images. Better recognition rates were attained depending upon the number of hidden layers and neurons.

Wee How Khoh et al. [17] aimed to determine whether transfer learning might be employed to categorize a signature based on hand gestures. Each depth image was analyzed by the algorithm to identify and segment the hand region. Next, from a variety of images, the significant spatial and temporal aspects were created. To classify the recently observed image features, the previously trained model data was transferred into the model again. Furthermore, examined the adaptability the proposed technique towards common types of forgeries like skilled and random. Additionally, the suggested strategy demonstrated its adaptability to various forgery assaults by obtaining low error rates.

An automatic technique based on optimal features selection and multi-level features fusion was suggested by Faiza Eba Batool et al. [18] for OSV. Eight geometric characteristics and 22 Gray Level Co-occurrences Matrix were computed for this purpose using pre-processing signature samples. An alternative approach based on high-priority index feature (HPIF) fuses these features. Skewness-kurtosis controlled PCA (SKcPCA), a skewness-kurtosis based features selection method, was also suggested. It selects the finest characteristics to be used in the final categorization of genuine and forged signatures. The suggested system was validated using MCYT, GPDS simulated, and CEDAR datasets, providing improvements in FAR and FRR when compared to current techniques.

The review focuses on significant developments in signature verification approaches, ranging from image processing to deep learning. But there is a clear research gap regarding the limited investigation of real-world settings, as most studies concentrate on artificial datasets and controlled

environments. The lack of inherent variability in writing styles, such as differences in rhythm, intensity, and personal preferences, makes it difficult for existing signature verification techniques to be applied. Another significant deficiency is the lack of focus on the interpretability and explainability of complicated models especially with neural network and kernel-based systems. It takes a lot of work to choose the right kernel, increase training complexity, and extend to multi-class settings. The overall accuracy of classification is reduced when majority class labels are incorrectly labelled due to imbalanced datasets. Retraining neural networks to account for variations in the quantity of signature classes is a time-consuming and computationally costly process. These approaches perform inadequately with small sample sizes because they don't have a precise strategy for taking data uncertainties into account and force the creation of new data. Memory needs are a problem that affect system performance, particularly for big training datasets. Moreover, partial occlusion, clutter susceptibility, and the creation of large feature vectors increase storage costs and localization errors, which reduces the overall efficacy of signature verification.

3. Materials and Methods

In offline signature verification, the authenticity and explainability is crucial. Most existing offline signature verification technique approaches image processing to deep learning techniques. So, in this paper for improving the information flow and learning in deep networks, a dual path network architecture is developed. Figure 3 shows the schematic block diagram illustrating the proposed methodology.

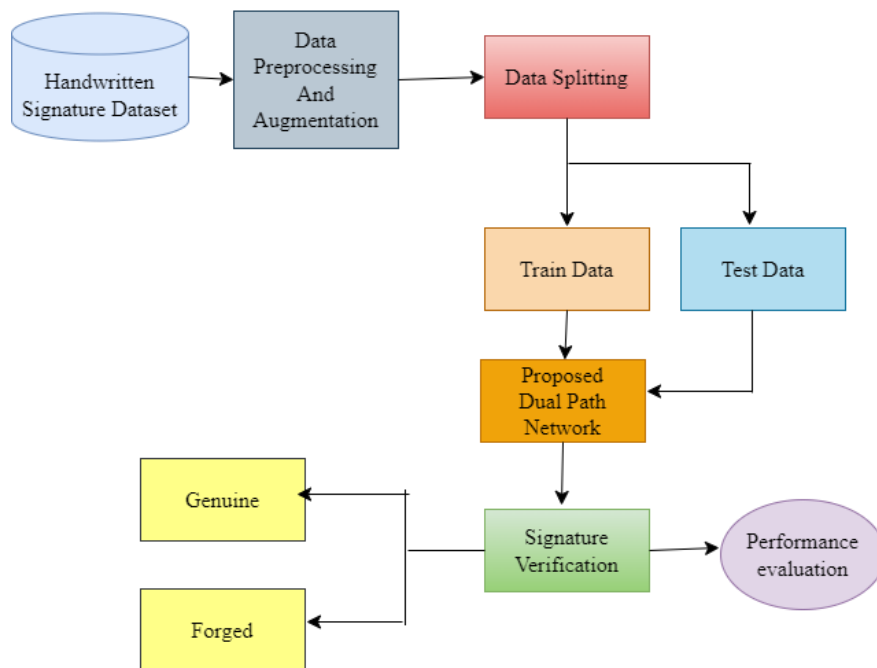


Fig.3. Schematic Block diagram representation of Proposed Methodology

3.1 Handwritten Signature Dataset

The dataset of handwritten signature has been collected from the Kaggle Repository, <https://www.kaggle.com/datasets/sinjini1999/sigcomp->

[signature-verification](#). The dataset consists with both genuine and forged signature images which have to be verified. Some sample images in the dataset are showed in Figure 4.

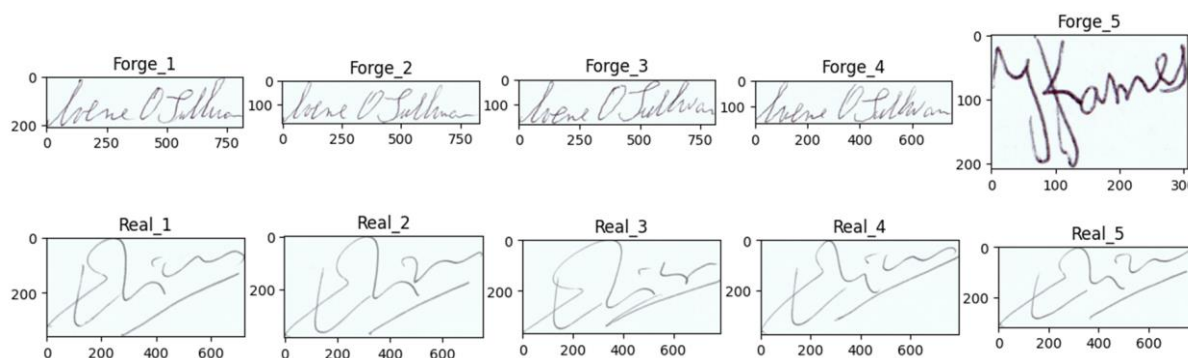


Fig.4. Sample Images in the Dataset

3.2 Data Pre-processing and Augmentation

The most important phase in the workflow for data analysis is data pre-processing. In order to prepare raw data for analysis or model training, it must be cleaned and transformed. The effectiveness and consistency of the outcomes from data-driven tasks can be greatly impacted by proper data pre-processing. It often involves filtering, normalisation etc. Filtering is the process of adding or eliminating specific characteristics or patterns from data. Normalization in pre-processing refers to the transformation or scaling of data to bring within a specific range. The normal range in which data is to be normalised is typically between 0 and 1. Data augmentation is a method that uses pre-existing data to create modified copies of a dataset, thereby artificially expanding the training set. It includes either creating new data sets through deep learning or making slight modifications to the dataset. The data augmentation techniques using here are rotation, flipping, shearing, zooming, and filling [19].

3.3 Proposed Model Architecture

3.3.1 Convolutional Neural Network

CNN is a feed-forward deep neural network (DNN) that used for visual imagery analysis. It functions similarly to how people perceive things. CNNs that use different multi-layer perceptron algorithms reduce the amount of pre-processing that is necessary for the incoming data. CNNs are made up of neurons that have biases and weights that can be learned. After receiving inputs and performing a dot product, each neuron has the option to add non-linearity. A single differentiable score function is created between the class scores on one end and the raw image pixels on the other throughout the network. CNNs furthermore feature a loss function on the final (fully-connected) layer, such as Softmax. The CNN layer differs significantly from other neural networks in that each unit is a two-dimensional filter, or high-dimensional filter, convolved with the layer's input rather than via general matrix multiplication [20]. Millions of neurons are arranged in multiple hierarchical levels within a typical CNN. The basic convolutional network architecture is given in Figure 5.

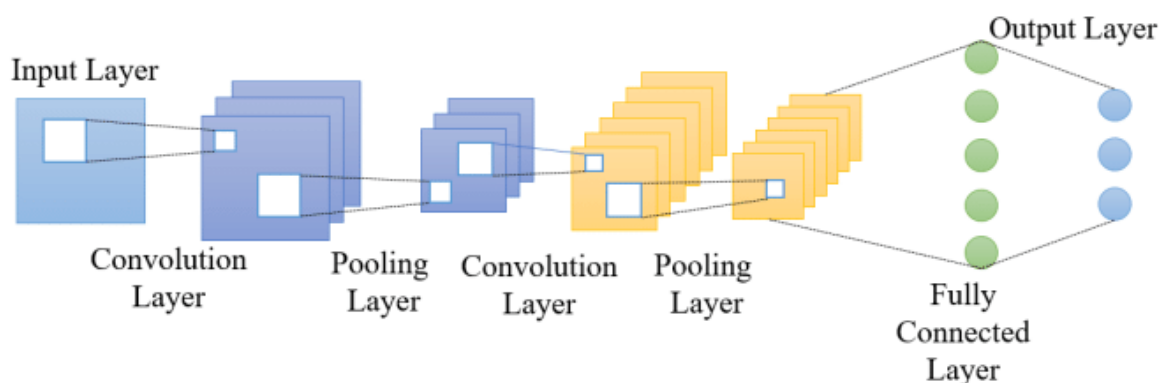


Fig.5. Basic architecture of CNN

CNN designs with three fundamental layers: convolutional, fully-connected (FC) and pooling. The convolutional layers are responsible for feature extraction through convolution operations, pooling layers for down sampling and feature

selection, activation functions to introduce non-linearities and enhance model expressiveness, and fully connected layers for making final predictions based on the extracted features. In certain cases, pooling layers are partially

connected and for decreasing the size of the input images. CNN's fully connected layer, or output, primarily functions as a classifier. The learnability of the networks is determined by the hidden layers found in the FC and convolution layers. The number of layers in a CNN correlates with its depth, and the deeper the layer, the higher the degree of characteristics it retrieves. Since hidden layer neurons connect to neurons in the layer below, it is easier to resize images of higher quality. Visual stimuli boost the input layer neurons in CNN processing. The convolution layer's primary function is feature extraction from images, which are then used to drive computations into hidden layers and retrieve the findings through the output layer. Activation functions frequently help transport important and vital information between hidden layers so that the subsequent layers can use it. The basic convolution operation is given by Equation 1.

$$y[m, n] = \sum_{i=0}^{k-1} \sum_{j=0}^{k-1} x[m+1, n+j] \cdot w[i, j] \quad (1)$$

where $w[m, n]$ represents the filter values, $x[m, n]$ represents the input data word, and $y[m, n]$ represents one data word in the output. A pixel and its surrounding pixels are multiplied by a tiny filter matrix known as a kernel in image convolution, and the resultant output central pixel is then added up. A CNN's convolutional layers process the input image using a convolution operation before sending the outcome to the following layer. From the input image, this layer pulls different features for additional analysis. A convolution layer uses a number of distinct kernels to follow various tasks like sharpening, edge detection, and blurring [21]. There exist some hyper parameters involving like kernel size, stride, zero padding etc. The Figure 6 illustrates the pooling procedure.

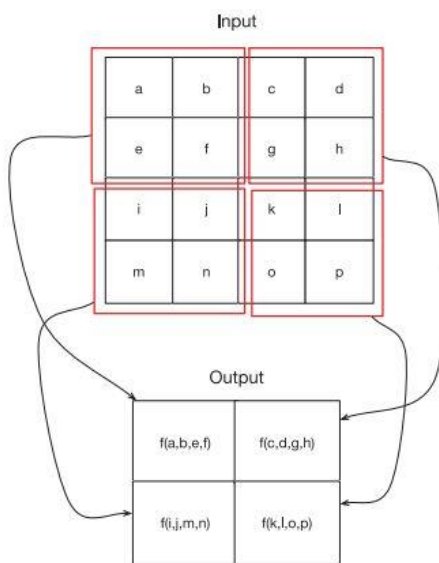


Fig.6. Pooling or subsampling

The CNN's pooling or down sampling layer is responsible of progressively decreasing the dimension of the feature maps generated by the convolution layer. The convolution layer's depth is unaffected by the pooling layer. By eliminating unnecessary geographic information, the pooling layer reduces computational expenses and helps avoid over fitting. Local or global pooling layers in convolutional networks combine the output of neurons from the preceding layer into a single neuron for the subsequent layer. The two most popular methods for pooling layers are max pooling and average pooling [22].

Feature map of the input is denoted as I , the size of the pooling window, also called pool size as $k \times k$, and the resulting pooled feature map as P . The Equation 2 gives max pooling operation.

$$P(i, j) = \max_{m=0}^{k-1} \max_{n=0}^{k-1} I(i.s + m, j.s + n) \quad (2)$$

where:

$P(i, j)$ is the value of pooled feature map at position (i, j)

$I(i.s + m, j.s + n)$ is the value of input feature map at position $(i.s + m, j.s + n)$

$k \times k$ is the dimension of the pooling window

S is the stride (the step size of the pooling operation)

At each position (i, j) of the pooled feature map, the maximum value within the corresponding $k \times k$ region of the input feature map is selected. By calculating the maximum value inside each pooling window, max pooling efficiently down samples the feature map while keeping the most notable features. Max pooling is more frequently utilized in practice for CNNs than average pooling, which evaluates the average value inside each pooling window. In order to add non-linearity to neural networks and enable them to detect non-linear properties in the input data, activation functions are performed on the input.

3.3.2 Proposed Dual Path network

The images which are pre-processed and augmented are fed to the proposed Dual Path Network model. A convolutional neural network that exhibits an internal connection route topology is called a Dual Path Network (DPN). The goal of this architecture is to improve gradient flow during training and prevent the vanishing gradient issue. The network can simultaneously capture high-level and low-level characteristics due to the dual path design, which improves its capacity to learn hierarchical representations. DPNs have shown enhanced performance across a range of computer vision applications, proving their efficiency in managing the difficulties associated with deep learning architectures. The Dual Path Network architecture that has been suggested is aimed at obtaining high-level semantic information and fine-grained details from the incoming data. The network

consists of two separate branches, each of which goes through a series of pooling and convolutional layers to process the input data. In order to extract features at diverse scales, these branches have distinct convolutional kernel sizes (3x3 and 7x7 for branch1 and branch2, respectively).

The purpose of DPNs is to improve learning and information flow in deep networks through the use of neural network architecture [23]. This image model block in

convolutional neural networks shares common traits and facilitates the investigation of new features using dual path architectures [24]. DPN is intended to improve learning and information flow in deep networks. Using a "main path" for conventional feature learning and a "dual path" for additional information flow via a more complex pathway, DPNs introduce a dual path structure. The proposed model architecture is given in Figure 7.

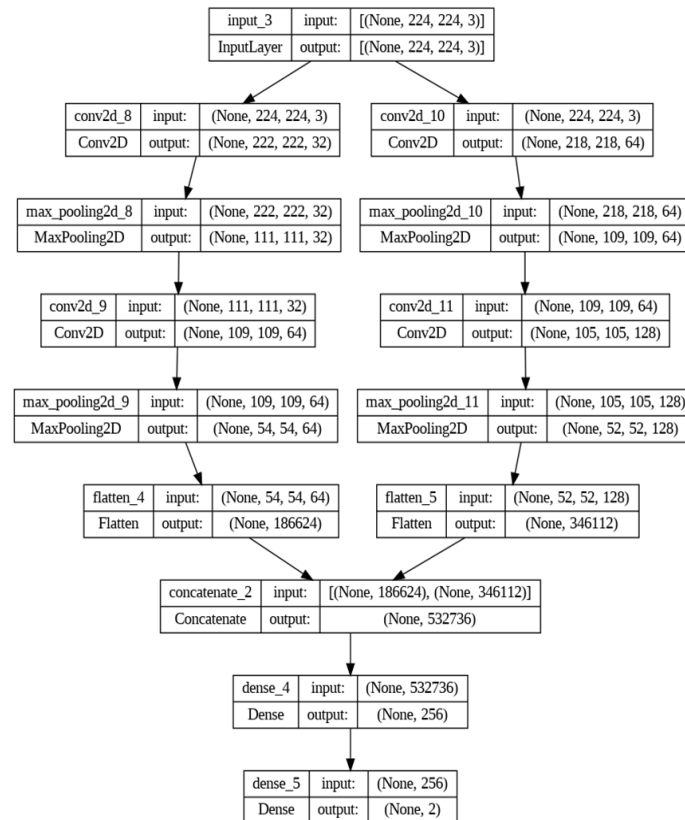


Fig.7. Proposed Model Architecture

The network can effectively identify local and global patterns in the input images by using different kernel sizes. Each branch's outputs are flattened and connected using the concatenate layer after feature extraction, resulting in a fused representation that incorporates the many features that each branch has learned. This combination of data is essential to improving the model's comprehension of complicated trends and variances in the input data. Following a shared fully connected layer having 256

neurons and a ReLu activation function, the concatenated features are then fed via an output layer with activation function of Softmax for classification. The Dual Path Network architecture that has been suggested seeks to enhance the model's representation learning skills for signature forgery classification using advantage of multi-scale feature extraction. The algorithm of proposed method is given below.

Algorithm.1. Pseudo code of the proposed methodology

Input: Handwritten Signature Images

Output: Forged Signature Classification

Begin

- Dataset collection
 - Preprocessing and data augmentation: Cleaning, organizing, and converting raw data into a format suitable for analysis.
-

- ❖ Applying filtering, normalization
- ❖ Rotation, flipping, shearing, zooming and filling can be used for data augmentation
- Data Splitting: Splitting of data to training and testing set
- Dual Path Network:
 - ❖ Main path for traditional feature learning and dual path for additional information flow
 - ❖ Initialize with different convolutional kernel size: 3x3 and 7x7 for branch1 and branch2, respectively
 - ❖ Convolutional and pooling layer for processing input data
 - ❖ After convolution, applying activation function (e.g. ReLu)
 - ❖ Extracting features from different convolutional layers
 - ❖ Concatenate or sum features for fusion
- Dense Layer with Softmax Classifier:
 - ❖ Flattened the fused features for input to the softmax classifier
 - ❖ Apply softmax function for the classification
 - ❖ Select the class with the greatest probability to be given the predicted label.

End

3.4 Hardware and Software Setup

The model was developed and trained on Google Collaboratory, where the entire process was completed using Python and Tensorflow. The Adam optimizer, binary cross entropy loss function, and accuracy as the evaluation metric are used to create the model. For training, a batch size of 128 samples per iteration is used, and the procedure is carried out over ten epochs. The table representing the hyper parameters and its values are given in Table 1.

Table.1. Hyper parameter Specifications

Hyper parameter	Values
Optimizer	Adam
No. of epochs	10
Loss Function	Binary Cross entropy
Batch Size	128

4. Result and Discussion

4.1 Performance Evaluation

Accuracy, recall, precision, and F1-Score were used for the evaluation of the suggested Dual path network. The accuracy of the model is determined by taking the ratio of accurate predictions to the total number of predictions. Accuracy can be mathematically expressed as in Equation 3.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (3)$$

The precision is measured as the ratio of all samples correctly classified (or incorrectly labeled as positive) to all samples correctly classed as positive. The accuracy measures how precisely the model classifies a sample as positive. It can be expressed as in Equation 4.

$$\text{precision} = \frac{TP}{TP+FP} \quad (4)$$

The ratio of correctly categorized positive samples to the total number of positive samples is used to compute the recall. Recall evaluates how the model can identify positive samples effectively. The value of recall will be high if more positive samples are detected. The mathematical expression is given in Equation 5.

$$\text{Recall} = \frac{TP}{TP+FN} \quad (5)$$

The harmonic mean of recall and precision determines an F1 score. The F1 score combines precision and recall into a single metric for evaluation purposes in binary and multi-class classification to improve comprehension of model performance. F1 score is mathematically expressed as in Equation 6.

$$\text{F1 score} = \frac{2*\text{precision}*\text{recall}}{\text{precision}+\text{recall}} \quad (6)$$

Table 2 presents the performance analysis result of the proposed approach for offline signature verification in terms of recall, f1 score, accuracy, and precision.

Table.2. Classification Report of Proposed Method

Performance Parameters	Result Obtained (%)
Accuracy	97.39
Precision	96.79
Recall	97.00
F1- Score	97.20

From Table 2 it is clear that with an accuracy of 97.39%, the model's performance indicators show great overall efficiency. With a precision score of 96.79%, the model effectively minimizes false positives by correctly

identifying positive cases within its predictions. With a 97% recall rate, the model minimizes false negatives by capturing a large percentage of true positive cases. A well-balanced trade-off between recall and precision is shown by the F1-score, which takes both factors into account and comes out at 97.20%. All of these findings point to the model's stability and dependability, as well as its high degree of accuracy and careful handling of both false positives and negatives while accurately recognizing positive cases.

A model's performance throughout training iterations or epochs is visually represented by an accuracy plot, which indicates how correctly the model predicts outcomes. A loss plot displays the model's error or loss function as it decreases across training iterations, showing how the model's parameters are optimized for improved performance and convergence. Here accuracy plot and loss plot are illustrated in Figure 8 and Figure 9.

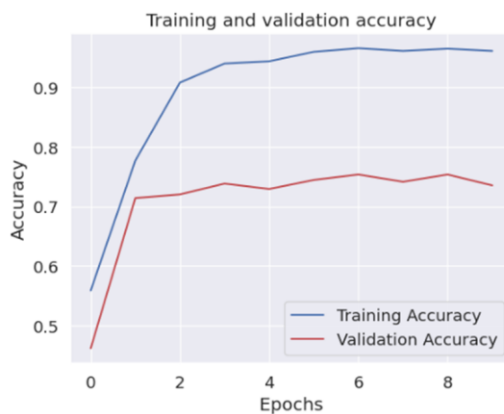


Fig.8. Accuracy Plot of proposed method

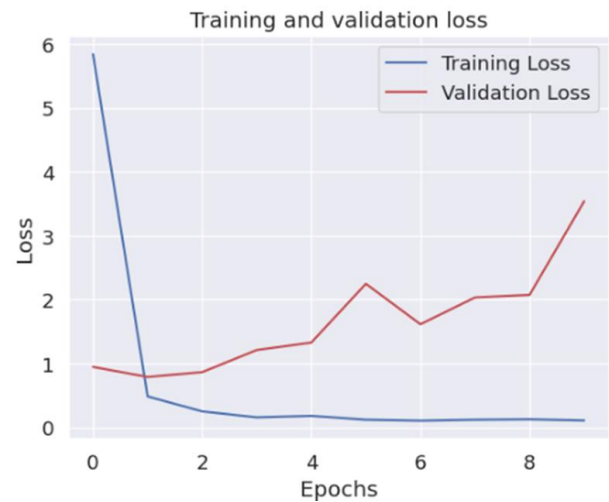


Fig.9. Loss Plot of proposed method

A confusion matrix shows accuracy of classification model. The total number of false positives, true positives, false negatives and true negatives is shown. The confusion matrix of the proposed model is given in Figure 10.

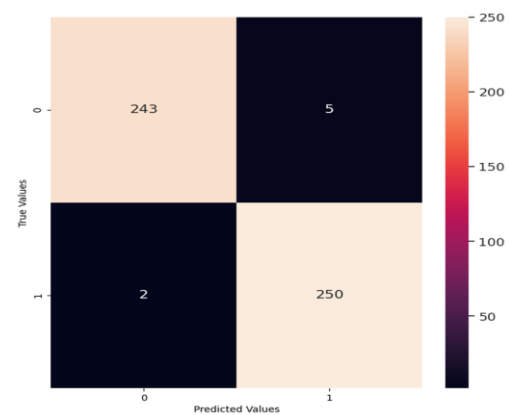


Fig.10. Confusion Matrix

4.2 Performance Comparison

The performance analysis of the suggested dual path network is compared with the current approaches mostly based on deep learning and machine learning. The performance is assessed by comparing the values of accuracy, recall, precision, and F1 score values obtained from existing methods as illustrated in Table 3.

Table.3. Comparison of proposed model with existing methods

Authors and year	Methodology	Accuracy (%)	Precision (%)	Recall (%)	F1 score (%)
Narwade et al. (2018)	Support vector machine	89.58	-	-	-
Upadhyay et al. (2020)	Support vector machine	88	80	90	89
Chandra (2020).	Machine learning	92	90	-	88
Kao et al.	Deep learning	89.5	-	-	-

(2020)					
Khoh et al. (2021)	Transfer learning	87.43	-	-	-
Sharif et al. (2020)	Feature selection algorithm	88.8	-	-	-
Sharma et al. (2022)	Siamese CNN	80	76	83	79
Proposed method		97.39	96.79	97	97.20

5. Conclusion

Signature verification methods are useful for confirming a document's authenticity. The systems are also employed for the purpose of authenticating signatures on financial documents such as money orders, cheques, and other ones. In particular, signature verification techniques are essential for verifying author identity and document validity in security and financial contexts. A significant advance in this area is the suggested deep learning-based offline signature verification technique that uses a dual path network architecture and an extensive dataset. With balanced precision, recall rates, F1 score and great accuracy, the model distinguishes between real and fake signatures with ease. Moreover, its ability to extract features on many scales and visualize training progress shows potential for improving document security and authenticity. The present study highlights the capacity of deep learning algorithms to deal with the complexity involved in signature verification procedures, hence providing opportunities for additional investigation and real-world implementations in this field. With 97.39% accuracy, 96.79% precision, 97% recall, and 97.20% F1-Score, the suggested model performs remarkably well. These findings demonstrate the efficacy of the deep learning-based technique in precisely confirming signatures and identifying suspected forged.

Conflicts of interest

The authors declare no conflicts of interest.

References

- [1] Sharif, M., Khan, M. A., Faisal, M., Yasmin, M., & Fernandes, S. L. (2020). A framework for offline signature verification system: Best features selection approach. *Pattern Recognition Letters*, 139, 50-59.
- [2] Harakannanavar, S. S., Renukamurthy, P. C., & Raja, K. B. (2019). Comprehensive study of biometric authentication systems, challenges and future trends. *International Journal of Advanced Networking and Applications*, 10(4), 3958-3968.
- [3] Ghanim, T. M., & Nabil, A. M. (2018, December). Offline signature verification and forgery detection approach. In 2018 13th international conference on computer engineering and systems (ICCES) (pp. 293-298). IEEE.
- [4] Nam, S., Park, H., Seo, C., & Choi, D. (2018). Forged signature distinction using convolutional neural network for feature extraction. *Applied Sciences*, 8(2), 153.
- [5] Narwade, P. N., Sawant, R. R., & Bonde, S. V. (2018). Offline signature verification using shape correspondence. *International Journal of Biometrics*, 10(3), 272-289.
- [6] Sharif, M., Khan, M. A., Faisal, M., Yasmin, M., & Fernandes, S. L. (2020). A framework for offline signature verification system: Best features selection approach. *Pattern Recognition Letters*, 139, 50-59.
- [7] Masoudnia, S., Mersa, O., Araabi, B. N., Vahabie, A. H., Sadeghi, M. A., & Ahmadabadi, M. N. (2019). Multi-representational learning for offline signature verification using multi-loss snapshot ensemble of CNNs. *Expert Systems with Applications*, 133, 317-330.
- [8] Shariatmadari, S., Emadi, S., & Akbari, Y. (2019). Patch-based offline signature verification using one-class hierarchical deep learning. *International Journal on Document Analysis and Recognition (IJDAR)*, 22(4), 375-385.
- [9] Wei, P., Li, H., & Hu, P. (2019). Inverse discriminative networks for handwritten signature verification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 5764-5772).
- [10] Kao, H. H., & Wen, C. Y. (2020). An offline signature verification and forgery detection method based on a single known sample and an explainable deep learning approach. *Applied Sciences*, 10(11), 3716.
- [11] Poddar, J., Parikh, V., & Bharti, S. K. (2020). Offline signature recognition and forgery detection using deep learning. *Procedia Computer Science*, 170, 610-617.
- [12] Vorugunti, C. S., Pulabaigari, V., Gorthi, R. K. S. S., & Mukherjee, P. (2020). Osvfusenet: online signature

verification by feature fusion and depth-wise separable convolution based deep learning. *Neurocomputing*, 409, 157-172.

- [13] Gumusbas, D., & Yildirim, T. (2019, July). Offline signature identification and verification using capsule network. In 2019 IEEE international symposium on innovations in intelligent systems and applications (INISTA) (pp. 1-5). IEEE.
- [14] Maergner, P., Pondenkandath, V., Alberti, M., Liwicki, M., Riesen, K., Ingold, R., & Fischer, A. (2019). Combining graph edit distance and triplet networks for offline signature verification. *Pattern Recognition Letters*, 125, 527-533.
- [15] Lu, J., Qi, H., Wu, X., Zhang, C., & Tang, Q. (2022). Research on Authentic Signature Identification Method Integrating Dynamic and Static Features. *Applied Sciences*, 12(19), 9904.
- [16] Kiran, P., Parameshachari, B. D., Yashwanth, J., & Bharath, K. N. (2021). Offline signature recognition using image processing techniques and back propagation neural network system. *SN Computer Science*, 2(3), 196.
- [17] Khoh, W. H., Pang, Y. H., Teoh, A. B. J., & Ooi, S. Y. (2021). In-air hand gesture signature using transfer learning and its forgery attack. *Applied Soft Computing*, 113, 108033.
- [18] Batool, F. E., Attique, M., Sharif, M., Javed, K., Nazir, M., Abbasi, A. A., ... & Riaz, N. (2020). Offline signature verification system: a novel technique of fusion of GLCM and geometric features using SVM. *Multimedia Tools and Applications*, 1-20.
- [19] Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of big data*, 6(1), 1-48.
- [20] Li, Z., Liu, F., Yang, W., Peng, S., & Zhou, J. (2021). A survey of convolutional neural networks: analysis, applications, and prospects. *IEEE transactions on neural networks and learning systems*, 33(12), 6999-7019.
- [21] Ketkar, N., Moolayil, J., Ketkar, N., & Moolayil, J. (2021). Convolutional neural networks. *Deep Learning with Python: Learn Best Practices of Deep Learning Models with PyTorch*, 197-242.
- [22] Zhang, Q., Zhang, M., Chen, T., Sun, Z., Ma, Y., & Yu, B. (2019). Recent advances in convolutional neural network acceleration. *Neurocomputing*, 323, 37-51.
- [23] Chen, Y., Li, J., Xiao, H., Jin, X., Yan, S., & Feng, J. (2017). Dual path networks. *Advances in neural information processing systems*, 30.
- [24] Zhu, W., Liu, C., Fan, W., & Xie, X. (2018, March). Deeplung: Deep 3d dual path nets for automated pulmonary nodule detection and classification. In 2018 IEEE winter conference on applications of computer vision (WACV) (pp. 673-681). IEEE.