SIGNATRUE VEERIFICATION USING NEURAL NETWORK

Niveditaa T.A[1], Sachin K M[2], Sushini. M[3], Suseedran S[4]

[1,2,3]*Student, Department of Computer Science and Engineering, Bannari Amman Institute ofTechnology, Tamilnadu, India*

[4]*Associate professor, Department of Computer Science and Engineering, Bannari Amman Institute of Technology, Tamilnadu, India*

ABSTRACT

The In the digital age, digital signatures are crucial instruments for verifying the reliability and accuracy of electronic documents and transactions. The Structural Similarity Index (SSIM) algorithm is used in this study to improve the precision and dependability of the digital signature verification process. This study adapts the SSIM method, which is well known for its efficiency in assessing the caliber of photos and movies, to assess the authenticity of digital signatures, resulting in a novel approach to verification. This substantial study provides a thorough analysis of the existing literature and approaches while exploring the theoretical foundations of digital signatures and SSIM. Additionally, it describes the mathematical underpinnings of SSIM and illustrates how it can be used to analyze digital signatures. The study methodology entails the creation of a unique verification system that incorporates the SSIM algorithm into the process of validating signatures. This study also thoroughly examines noise, changes, and scaling, three aspects that affect the verification of digital signatures. It examines the SSIM algorithm's resilience to these difficulties and gives actual findings that show how effective it is in comparison to more conventional approaches. The practical usefulness of this approach is demonstrated by the evaluation of real-world scenarios, such as e-commerce transactions and document authentication. The consequences of this discovery extend to fields including finance, healthcare, and law where data integrity and authenticity are essential.

Keywords: *data integrity, transaction, validation, precision, uniqueness, authenticity.*

1. INTRODUCTION

Signature verification is an important aspect of identity authentication, with applications in banking, legal proceedings, and security systems. Traditional methods frequently struggle with variations in writing styles and forgeries. SSIM offers a modern and effective solution to these problems. To assure the validity of signatures on papers, contracts, financial transactions, and more, signature verification is a crucial procedure utilized in many different industries and applications. In order to evaluate whether two signatures match and prove the transaction or document is authentic, it entails comparing a provided signature with a reference or real signature. The Structural Similarity Index (SSIM) algorithm is one method for verifying signatures.

Since ancient times, signatures have been employed as a form of verification and authentication. Signatures are essential in many facets of our life, whether they are used to confirm the arrival of a parcel, sign a contract, or authorize a financial transaction. The ability to manipulate digital documents easily and the evolution of technology have made it more difficult than ever to verify the legitimacy of signatures.

The process of confirming if a given signature corresponds to a recognized or reference signature is known as signature verification. The integrity of papers and transactions must be maintained, legal compliance must be ensured, and fraud must be avoided through this verification process. It is employed in many different fields and contexts, including banking, law, medicine, government, and other areas.

A popular approach for determining how similar two photos are structurally is the Structural Similarity Index (SSIM). Its concepts can be used for signature verification even though it was primarily created for image quality evaluation. By taking brightness, contrast, and structure into account, SSIM estimates the perceived change in

structural information between two images. The SSIM algorithm compares a test signature picture (the signature to be verified) with a reference signature image (a real signature) in the context of signature verification. It determines a similarity score, which shows how close the two signatures are to one another. The legitimacy of the signature can then be determined using this score. The SSIM algorithm's components, mathematical formulas, and other details will all be covered in more detail in the sections that follow.

The decision-making phase of the signature verification process is based on the SSIM score. The test signature's authenticity or forgery is determined using a judgment threshold. Critical decisions must be made regarding the threshold, which is determined by the application and the desired level of security. The test signature is recognized as authentic if the SSIM score is higher than the cutoff.

The test signature is deemed suspicious or perhaps fabricated if the SSIM score is below the cutoff. The threshold can be changed to manage the trade-off between false positives (real signatures mistakenly labelled as fakes) and false negatives (forged signatures mistakenly labelled as real). The selection of the threshold may require empirical research and thought on the effects of incorrect classifications in a given application.

2. LITERATURE SURVEY

In the literature survey concentrated on that the robotized identification of neonatal rest apnea is fundamental for compelled conditions with high persistent to nurture proportion. Existing examinations on apnea location generally target grown-ups, and utilize obtrusive sensors. Hardly any methodologies recognize apnea utilizing video observing, by distinguishing nonattendance of respiratory movement. They apply outline differencing and thresholding, not reasonable for youngsters because of their inconspicuous respiratory movement intermixed with other body developments. Proposed technique first applies movement amplification. Therefore, it channels breath movement utilizing dynamic thresholding. The procedure is benchmarked with mimicked movement of differing breath frequencies. When approved with neonatal video information, proposed technique accomplishes both > 90% awareness and particularity.

the application TYDR (Track Your Day-to-day Daily practice) which tracks cell phone sensor and use information and uses normalized psychometric character surveys. Their UI is intended to boost clients to introduce the application and finish up surveys. TYDR processes and pictures the followed sensor and use information as well as the aftereffects of the character polls. They advance the following of sensor information by surveying the compromise of size of information and battery utilization and granularity of the put away data.

wellness are many times depicted as a condition that assists us with looking, feel and do our most noteworthy. They fostered a unique start to finish Single Page Application (SPA) upheld Wellness, for the individuals who will move forward and pick a solid way of life. There are three substances inside the framework specifically EnerGyM Planet for example Administrator, Part and Mentor. Administrator will do authoritative undertakings like adding, eliminating and refreshing data related with different elements, Part will have highlights like intending to pick their mentor from recorded coaches accessible inside the rec center thus advancing to watch their recordings and acquire wellness tips and different data related with wellness industry.

3. STRUCTURAL SIMILARITY INDEX MEASURE

The main goals of signature verification using the SSIM algorithm are reduced human error, efficiency, fraud detection, document security, and authenticity. SSIM is a useful tool for automating signature verification procedures since it provides a strong metric for evaluating structural similarity between two signature images. By breaking down images into brightness, contrast, and structural components, the SSIM technique allows for a more thorough comparison of signatures. Data collection and preprocessing are essential processes in collecting high-quality signature datasets and getting them ready for analysis, which will lead to correct verification. SSIM values are frequently used as important features for categorization in feature extraction, which is a crucial step in the verification of digital signatures. Support Vector Machines (SVMs) and neural networks, among other machine learning methods, have been used to match and categorize signatures. The efficiency of SSIM-based signature verification systems is evaluated using performance assessment criteria like accuracy, precision, recall, F1-score, and ROC curves. These metrics give a complete picture of how well a system can tell real signatures from fakes.

3.1. Main Approaches Of SSIM

3.1.1. Preprocessing:

Create a grayscale version of the reference signature image and the target image. Grayscale makes the comparison easier because black and white is the usual representation of signatures.

3.1.2. SSIM Calculation:

Identify overlapping patches or sections in the target image. These patches' sizes are determined by the anticipated size of the signatures in your photographs. Calculate the SSIM score between each patch and the reference signature picture for each patch in the target image using the SSIM formula. The SSIM score ranges from -1 to 1, with higher scores indicating greater similarity. To compute SSIM scores in Python, use a package like OpenCV. For this use, OpenCV provides the cv2.SSIM function.

3.1.3. Localization:

Find the locations of the target image's patches where the SSIM score is insufficient. Signatures or portions of signatures are probably present in these areas.

3.1.4. Post-processing:

You might need to further hone the discovered signature regions depending on your application. For instance, you can combine nearby patches into coherent signature regions using morphological procedures or region-growing algorithms.

3.1.5. Visualization or extraction:

The discovered signature regions can be seen on the original image or extracted for additional examination or processing.

3.1.6. Temporal Analysis:

Analysing the evolution of a person's signature through time can help us understand the temporal aspect. Determine whether age or other factors have significantly changed the signature.

3.1.7. Error Analysis:

Conduct error analysis to understand the kinds of errors the system makes and then modify the model in accordance with your findings to continuously study and enhance the detection performance.

4. PROPOSED MODEL

Given in this visual description. This example helps explain the fundamental idea of how SSIM is used to compare signatures for verification, while there may be extra complications and optimizations in practice.

4.1 Data Collection and Preprocessing:

Data collection: Amass a massive collection of signature pictures. Make sure it includes a sizable number of both real and fake signatures, ideally more than 5000 samples. Increase the variety of the dataset by including different signature characteristics, traits, and variations that might be seen in real-world situations. Make sure the dataset complies with data privacy laws, and secure the relevant authorizations for data use.

4.2 Preprocessing:

Create grayscale versions of all signature images. SSIM is frequently used to compare grayscale photos, making the procedure easier. Resize signature photographs to a uniform scale and resolution to standardize their dimensions. This makes sure that all signatures, no matter how big or small they were originally, are equivalent. Use image enhancement techniques to enhance the signatures' clarity and legibility. Sharpening, noise reduction, and contrast correction are examples of this.

4.3 Database Setup:

Divide the dataset into subsets for training, validation, and testing. Allocate typically a sizable chunk (for example, 70%) to training, a lesser portion (for example, 15%) to validation for hyperparameter adjustment, and the remaining fraction (for example, 15%) to testing for the final evaluation. The training subset should be used to create a reference database. Real signatures will be kept in this database for comparison during verification.

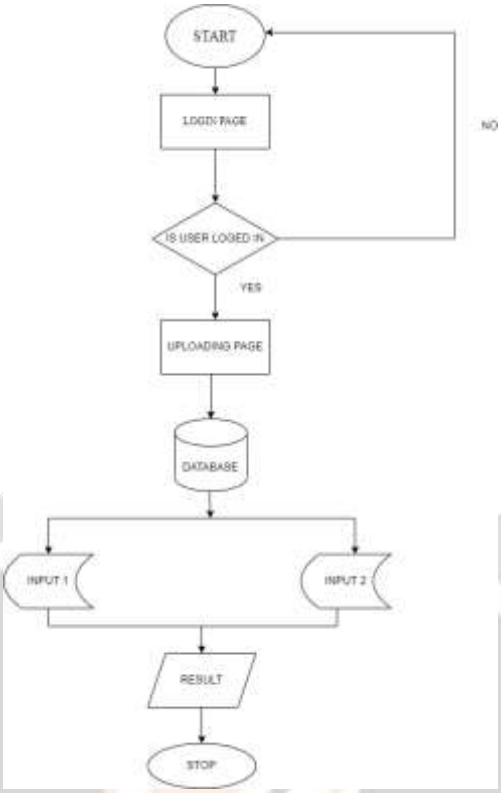

Fig-1: Methodology proposed

4.1. Feature Extraction Based On Dense-169

The previously altered images are then fed into the DenseNet-169 model, which generates a set of feature vectors. In the process of recognizing images, DCNN is a highly effective framework since it has access to particular pooling and convolution layer types. However, as the system becomes more complex, the amount of input data or gradient that must pass through each layer increasingly decreases, opening up access to the layer below it in the network. DenseNet ingeniously addresses the gradient vanishing problem by establishing direct connections between layers of the same feature size. This unique architectural feature ensures the seamless flow of gradients during training. One prominent advantage of employing the DenseNet model as a feature extractor lies in its ability to unearth richer and more comprehensive general features through meticulous layer-to-layer scrutiny. The DenseNet-169 model kicks off with an initial layer comprising pooling and convolutional operations, succeeded by three transitional layers, culminating in four densely connected blocks.

Stride 2 and 3x3 max pooling are used in the first convolution layer to achieve 7x7 convolution. The network then consists of three sets of dense blocks, each of which has a layer of change. By shifting a direct link from one layer to another, the DenseNet creates its dense connections. The gradient flow across the network is improved since the last layer receives the feature maps from every layer before it. This requires joining together the feature maps from the previous layer, which cannot be done unless each feature map has the same size. On the other hand, a convolution neural network is primarily concerned with down sampling the feature-map's dimensions. Several tightly linked blocks make up the DenseNet model, as was already explained. Described as a transition layer, this layer sits in between these thick blocks. A layer of batch normalization, a layer of 1x1 convolution with a stride of 2, and a layer of average pooling with a 2x2 size are included in each transition layer. There are four dense blocks, as was indicated previously, and each one is made up of two convolutional layers of 1x1 and 3x3 sizes. DenseNet169's pre-trained model, which was built on ImageNet, has 4 dense blocks that are each 6 by 12 by 32 pixels in size. The last classification layer employs global average pooling of 7x7 in contrast to the final FC layer, which employs the activation "softmax".

```
def captureImage(ent, sign=1):
    if(sign == 1):
        filename = os.getcwd()+'\\temp\\test_img1.png'
    else:
        filename = os.getcwd()+'\\temp\\test_img2.png'
    # messagebox.showinfo(
    #     'SUCCESS!!!', 'Press Space Bar to click picture and ESC to exit')
    res = None
    res = messagebox.askquestion(
        'Click Picture', 'Press Space Bar to click picture and ESC to exit')
    if res == 'yes':
        capture_image_from_cam_into_temp(sign=sign)
        ent.delete(0, tk.END)
        ent.insert(tk.END, filename)
    return True


def checkSimilarity(window, path1, path2):
    result = match(path1=path1, path2=path2)
    if(result <= THRESHOLD):
        messagebox.showerror("Failure: Signatures Do Not Match",
                             "Signatures are "+str(result)+f" % not similar!!")
        pass
    else:
        messagebox.showinfo("Success: Signatures Match",
                            "Signatures are "+str(result)+f" % similar!!")
    return True
```

Fig-2: code for

4.2. AE-Based Classification

To correctly identify the class labels for test photographs, the classification process makes use of the AE model. The components of AE are an activation function, a reconstruction layer of the d unit, a concealed layer of the h unit, and an activation function. In the training phase, the input $x\varepsilon R^d$ is first mapped to a hidden state, creating the latent activity, $y\varepsilon R^h$. The network, also known as an "encoder", corresponds to the process described in the boxed area. After that, y is translated into an output layer called "reconstruction" that is the same size as the input layer using a "decoder". $Z\varepsilon R^d$ represents the reconstructing value. While the reconstructed layer and parameter are removed during network training, the hidden node's learned features can be used for classification or as a source for future deep features by high layers.

4.3. COA-Based Parameter Optimization

COA is used during the parameter optimization process to determine the AE model's parameter values. In the case of irregular irradiance circumstances (i.e., partial shade), COA employs the searching for the neighborhood's greatest worldwide operational voltage points while driving and chasing it.

5. USING DEEP TRANSFER LEARNING IN ORAL CANCER

5.1. Data Collection And Annotation

Assemble a large database of mouth cancer photos, with benign and malignant instances included. Make sure the dataset is varied, inclusive of numerous demographics, and reflective of the lesion characteristics. Each picture should be classified as benign or malignant to annotate the dataset, a task normally carried out by knowledgeable physicians.

5.2. Data Preprocessing

By deleting duplicate or poor-quality photographs, clean up the dataset. Create uniform picture dimensions by standardizing them. Utilize data augmentation methods, such as rotations, flips, and scaling, to broaden the training dataset's variety. Benign/malignant category labels should be converted to numerical format.

```
root = tk.Tk()
root.title("Signature Matching")
root.geometry("500x700")  # 300x200
uname_label = tk.Label(root, text="Compare Two Signatures:", font=10)
uname_label.place(x=90, y=50)

img1_message = tk.Label(root, text="Signature 1", font=10)
img1_message.place(x=10, y=120)

image1_path_entry = tk.Entry(root, font=10)
image1_path_entry.place(x=150, y=120)

img1_capture_button = tk.Button(
    root, text="Capture", font=10, command=lambda: captureImage(ent=image1_path_entry, sign=
img1_capture_button.place(x=400, y=90)

img1_browse_button = tk.Button(
    root, text="Browse", font=10, command=lambda: browsefunc(ent=image1_path_entry))
img1_browse_button.place(x=400, y=140)

image2_path_entry = tk.Entry(root, font=10)
image2_path_entry.place(x=150, y=240)

img2_message = tk.Label(root, text="Signature 2", font=10)
img2_message.place(x=10, y=250)

img2_capture_button = tk.Button(
    root, text="Capture", font=10, command=lambda: captureImage(ent=image2_path_entry, sign=
img2_capture_button.place(x=400, y=210)

img2_browse_button = tk.Button(
    root, text="Browse", font=10, command=lambda: browsefunc(ent=image2_path_entry))
img2_browse_button.place(x=400, y=260)

compare_button = tk.Button(
    root, text="Compare", font=10, command=lambda: checkSimilarity(window=root,
                                                                   path1=image1_path_entry.g
```

Fig-3: Pre-process dataset

5.3. Pre-Trained Model

A cutting-edge pre-trained deep learning model should be used as the foundation of your architecture. The options that are frequently used include DenseNet, ResNet, Inception, VGG, or EfficientNet. The pre-trained model need to have been developed using a sizable, multi-purpose picture dataset, such as ImageNet.

5.4. ransfer Learning

Set the learnt weights and architectural parameters for the chosen pre-trained model. Add more layers to the architecture to add your own features for feature extraction and classification on top of the pre-trained model. Some of the early layers, known as the feature extraction layers, should be frozen to stop them from changing

during training. By training the model on the oral cancer dataset, you may fine-tune it and update certain layers while still using the pre-trained model's information.

5.5. Accuracy

An important worldwide health issue, oral cancer frequently has poor patient outcomes due to late-stage detection. A potential approach to improve early cancer diagnosis is deep learning, particularly transfer learning. Accuracy is crucial in healthcare. It immediately affects the results and patient care. Accuracy is the capacity to correctly classify oral lesions as benign or malignant in the context of detecting mouth cancer. Correct identification guarantees that patients receive prompt care, increasing survival rates and lowering morbidity. The strength and variety of the training and assessment datasets serve as the foundation for efficient deep learning models for oral cancer early detection.

6. RESULT AND DISCUSSION

The quality of the signature dataset, the preparation procedures, the selection of the classification technique, and the particular use case can all affect the outcomes of signature verification utilizing the Structural Similarity Index (SSIM) algorithm. I can, however, provide you a general overview of the kinds of outcomes and performance indicators generally employed to assess the efficiency of SSIM-based signature verification systems. Discussion for the Structural Similarity Index (SSIM) algorithm-based signature verification entails a thorough examination of the benefits, difficulties, applications, and possible developments in this area.

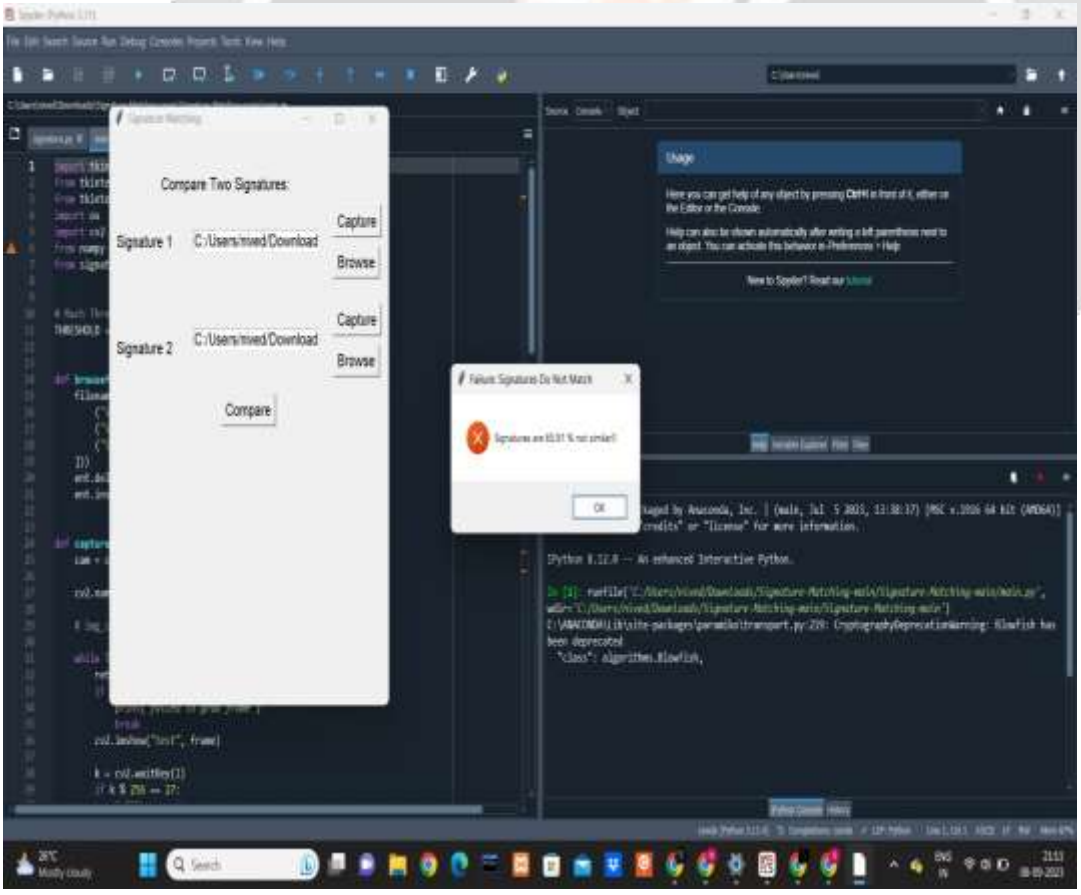

Fig-6: verification of the signature.

7. CONCLUSION

The Structural Similarity Index (SSIM) algorithm can be a useful tool for verifying signatures to determine the legitimacy and authenticity of handwritten signatures. A quantitative evaluation of the structural similarity between two images is provided by SSIM. This makes it a reliable tool for comparing signature photographs side by side and determining how similar or different they are. The ability of SSIM to identify structural changes in signatures is very impressive. It is helpful for spotting forgeries or unauthorized modifications since it may spot changes to a signature's general shape, stroke thickness, and spatial distribution of elements. SSIM is intended to replicate how the human visual system responds to changes in an image's structural composition. It considers luminance, contrast, and structure, which is in line with how people interpret variations in signatures. Setting a threshold value at which two signatures are deemed close enough to be regarded as genuine is frequent when utilizing SSIM for signature verification. Depending on the needs of the application, this threshold can be changed to regulate the ratio of false positives to false negatives. The usual variations in signatures that happen over time in real signatures, such as changes in pen pressure and writing speed, can be handled by SSIM. While SSIM is efficient at catching many different kinds of forgeries, it might not be impenetrable to highly competent forgers who can precisely recreate signatures. Challenges can arise from intricate forgeries that closely resemble the actual signature. To increase overall accuracy and resilience, SSIM can be used in conjunction with other signature verification methods, such as feature-based approaches or machine learning algorithms. To increase overall accuracy and resilience, SSIM can be used in conjunction with other signature verification methods, such as feature-based approaches or machine learning algorithms. The incorporation of deep learning methods is one of the most potential areas for future study and improvement in SSIM-based signature verification. Convolutional Neural Networks (CNNs) have proven to be incredibly effective at image processing tasks, and using them to verify signatures can greatly increase accuracy and robustness. Future research should investigate architectures created expressly for tasks involving signature verification, taking into account the distinctive qualities of signature images. On big signature datasets, transfer learning and fine-tuning of pre-trained models may lessen the need for laborious data collecting and annotation activities.

8. REFERENCES

[1]Prachi Chauhan; Subhash Chandra; Sushila Maheshkar Static digital signature recognition and verification using neural networks, 12-14 August 2016,17029494, Electronic ISBN:978-1-4673-6984-8 Print on Demand(PoD) ISBN:978-1-4673-6985-5

[2]F. Leclerc and R. Plamondon, "Automatic signature verification: The state of the art 1989–1993", International Journal of Pattern Recognition and Artificial Intelligence, vol. 8, no. 03, pp. 643-660, 1994. signature is unique for each individual pertaining to pressure, intensity, speed, area and such other properties, hence they often become difficult to verify with the naked eye

[3]D. Impedovo and G. Pirlo, "Automatic signature verification: the state of the art", Systems Man and Cybernetics Part C: Applications and Reviews IEEE Transactions on, vol. 38, no. 5, pp. 609-635, 2008.Personal verification and recognition is an actively growing area of research which includes tremendous exploitation of biometrics of an individual

[4]B. Herbst and H. Coetzer, "On an offline signature verification system", 9th Annual South African Workshop on Pattern Recognition, pp. 39-43, 1998. Biometrics are characterized by personal traits such as fingerprint, retina, iris, odor and gait; they are reliable because of their natural uniqueness for every person

[5]A. Pansare and S. Bhatia, "Offline signature verification using neural network", International Journal of Scientific & Engineering Research, vol. 3, no. 2, 2012. Creating a system with the ability to recognize handwritten signature and verify its authenticity is a challenging issue to deal with.

[6]D. Uppalapati, "Integration of offline and online signature verification Figure 6. Pattern Recognition of Neural network systems", Department of Computer Science And Engineering IIT Kanpur, 2007. Forgery is an act of deceit designed to falsify an individual's identity or counterfeit a document.

[7]D. Bertolini, L.S. Oliveira, E. Justino and R. Sabourin, "Reducing forgeries in writer-independent off-line signature verification through ensemble of classifiers", Pattern Recognition, vol. 43, no. 1, pp. 387-396, 2010. Although technology has given a lot to the world positively, it has indeed given efficient tools to criminals to commit crimes like forgery.

[8]A. Karouni, B. Daya and S. Bahlak, "Offline signature recognition using neural networks approach", Procedia Computer Science, vol. 3, pp. 155-161, 2011. Artificial Neural Networks ANNs): Creating a system with the ability

to recognize handwritten signature and verify its authenticity is a challenging issue to deal with, it generally goes out of the conventional practice of writing algorithms .
[9]H.H. Wai and S.L. Aung, "Feature extraction for offline signature verification system", IJCCER, vol. 1, no. 3, pp. 84-87, 2013. A neuron receives excitatory input sufficiently large compared with its inhibitory input and then sends a spike of electrical activity down its axon.
[10]K. Huang and H. Yan, "Off-line signature verification based on geometricy feature extraction and neural network classification", Pattern Recognition, vol. 30, no. 1, pp. 9-17, 1997. In this paper, the recognition and verification of offline signature samples using ANNs is presented as it follows a paradigm which models human learning patterns.