IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

SIGNATURE VERIFICATION AND FORGERY DETECTION SYSTEM

[1]Navya V K, [2]Abhilasha Sarkar, [3]Aditi Viswanath, [4]Akshita Koul, [5]Amipra Srivastava

[1]Assistant Professor, [2]Undergraduate Scholar, [3]Undergraduate Scholar [4]Undergraduate Scholar, [5]Undergraduate Scholar

[1]Department of Computer Science and Engineering,

[1]MVJ College of Engineering, Bangalore, Karnataka, India.

Abstract: Forgery and fraud in signature verification pose significant challenges in maintaining security and authenticity in various domains. It introduces a Signature Forgery Detection and Verification system that employs image processing, optical character recognition (OCR), and machine learning techniques to detect forged signatures and ensure their authenticity. The Signature Forgery Detection and Verification system enhances the security and accuracy of signature verification processes. By combining image processing, OCR, and machine learning techniques, it provides a comprehensive solution to identify potential forgery attempts and verify the authenticity of signatures. The system finds practical applications in various domains, contributing to fraud prevention and ensuring secure transactions.

***Index Terms* – OCR, LineSweep, Tesseract, CNN, Pooling, Fully Connected.**

I. INTRODUCTION

Forgery and fraud in signature verification have become significant concerns in various domains, including banking, legal documents, and identity verification. Here, we present a Signature Forgery Detection and Verification system that utilizes image processing, optical character recognition (OCR), and machine learning techniques to detect forged signatures and ensure the authenticity of signatures. The system consists of a user-friendly graphical user interface (GUI) developed using the Tkinter library in Python. The GUI allows users to interact with the system seamlessly. The first page of the GUI provides a login feature to authenticate users, ensuring authorized access to the system. Users can register by providing their username and password, which are securely stored in a credentials file for future login sessions.

Upon successful login, users are directed to the second page, where they can perform various tasks related to signature forgery detection and verification. The page displays the project's title, "Signature Forgery Detection and Verification," creating a visually appealing interface. Users have multiple options on this page, such as browsing and selecting an image for analysis, extracting text using OCR, and classifying the image for signature presence. The OCR feature allows the system to extract text from the image, enabling the detection of specific keywords or information such as IFSC codes or predefined keywords like "Please" or "Sign." This extraction process is achieved using the Tesseract OCR engine, which recognizes text in images.

Furthermore, the system incorporates machine learning algorithms to classify the image based on signature presence. The trained model leverages image features and patterns to determine whether the signature in the image is genuine or forged. This classification provides an additional layer of verification and helps in distinguishing between authentic and fraudulent signatures. The Signature Forgery Detection and Verification system aims to enhance the security and accuracy of signature verification processes. By combining image processing, OCR, and machine learning techniques, it offers a comprehensive approach to identify potential forgery attempts and verify the authenticity of signatures. This system has practical applications in various domains where signature verification plays a crucial role, contributing to fraud prevention and ensuring secure transactions.

II. LITERATURE SURVEY

In this paper, it proposes a detailed method for detecting forged signatures in offline signature verification using Siamese neural networks. Siamese grids are designed to capture two different, similar, or different images and calculate a distance measure between higher-level features to determine their similarity. The authors used a preprocessing step to scale all images to a fixed size of 155 × 220 using bilinear interpolation. The similarity measure used in this network includes the Euclidean distance between tree representations on either side of the Siamese network. The loss function used is divergent loss, which ensures that images of the same class are closer to each other than images of different classes. The activation function used is ReLu, which helps make the network more efficient [1].

This aims to achieve the ability to distinguish between authentic and fake signatures. This feature is implemented using a support vector machine (SVM). Here, the signature is represented as a boundary consisting of x-y coordinates, and the data is then stored as a text file in the signature database. SVMs are commonly used in pattern recognition and regression problems [2].

The proposed model is divided into three phases: the preprocessing phase, the feature extraction phase, and the validation phase. In the preprocessing stage, from the signatures collected with the scanner, signature features are extracted through a preprocessing step consisting of denoising, volume normalization, and skeletonization. In the feature extraction stage, two sets of features are used, such as network features and global features, where the network information features divide the image into an appropriate number of rectangular regions, and the general features provide state-specific information about the structural signature. In the validation phase, a neural network (NN) is trained, in which a standard backpropagation neural network classifier is used for validation. The proposed neural network consists of 30 input variables extracted from signature features, and then designed to verify one signature at a time. It should also be noted that using the proposed algorithm improves the FRR and FAR values compared to the existing algorithms [3].

To identify and verify identity, we only try to use human signatures signed by people, which is a secure way to identify people, especially in areas related to banking and other financial and legal transactions, but some techniques are used with loopholes and abuses; The purpose of these people is to commit fraud. That's why technology is evolving rapidly, not only to keep up with the next trends, but also to stay one step ahead of scammers. Technology exists to facilitate the work of humans so that transactions can be done online or offline as per the convenience. What we are talking about here is a biometric technology that can be used as a means of confirming an individual's identity. It consists of two types: - (1) physical (2) behavioral [4].

The use of Online Signature Verification (OSV) is critical to achieving a paperless office, but it still faces significant challenges. To solve this problem, this paper proposes a new OSV framework for the freelance writer (WI). The framework consists of three parts: (1) a two-dimensional rendering method that converts time-series signature data into linear images with mixed static and dynamic information. (2) A per-channel weight learning (CWL) mechanism built into the feature extractor to detect possible relationships between three dynamic attributes (elevation, azimuth, and pressure). (3) Three-way supervised network (TSN) consisting of three streams of weighted convolutional neural network (CNN) to calculate the distance [anchor, positive, test]. The experimental results showed that the proposed model outperforms the classical CNN and the lightweight model by no less than 1.29% and 11.3% in eligible forged signatures, respectively. Also, TSN patterns are more effective than previous OSV algorithms in detecting WI patterns. This proposes a new framework for author method-independent online signature validation [5].

The proposed system uses the Harris algorithm, the Surf algorithm, and pixel-based methods to efficiently verify signatures. This paper emphasizes the need for collections of signatures rather than signatures for verification, the basic reason being that a single signature cannot show all the factors of its signature that may vary under different circumstances. It is claimed that with these algorithms we can find small details for efficient verification of signatures compared to manual examination [6].

Three types of forgery have been mentioned regarding signature forgery. Accidental infringement, unqualified infringement and qualified infringement. Accidental forgery is when the signature forger does not know the victim's signature but intends to forge the victim's signature. Unconditional forgery is a form of forgery in which the intruder knows the victim's signature but cannot copy it exactly. Skilled forgery is when the forger is fully aware of the victim's signature and is skilled enough to transcribe it. The paper suggests that we should have a system that protects against any spoofing described above. Using a neural network to verify signatures is beneficial because it is very easy to use and capable of solving complex problems. Dynamic functions such as x-y coordinates, pressure, time, etc. is a function of time, so the paper highlights that not even a skilled forger can replicate all of these parameters. This makes these dynamic functions suitable for signature verification [7].

The term "micromorphing" refers to subtle changes in signature strokes or writing style that distinguish real signatures from professionally forged ones. The study shows that convolutional neural networks (CNNs) can identify these subtle distortions using a technique called max pooling, which involves tracking the coordinates of the highest values in a pooling window. By incorporating this location data as an additional feature of convolutional features, the proposed method achieves better performance than existing systems on four publicly available datasets in English, Persian, and Hindi. The results show that CNNs have the ability to accurately detect subtle distortions and improve signature verification systems [8].

Although online signature verification (OSV) techniques have been used in production systems for many years, significant challenges remain in training a model to accurately classify test signatures using only a limited number of training samples for test signatures. However, the development of convolutional neural networks (CNNs) has greatly improved the efficiency of OSV systems. Online Signature Verification consists of two main steps. The first step is to extract unique and useful features from signature data collected online. The second step is to develop a robust model or framework that can use the extracted features to determine the credibility of associated signatures. In the traditional feature extraction process, many dynamic local features, such as x and y coordinates, feature layout, azimuth, pressure, etc., are extracted when the user is connected to a specialized device. Record in each sample. This presents "OSV FuseNet", a framework based on separable deep convolutional neural networks for online signature validation. The framework uses both well-designed and deep learning-based features, and combines them through a feature-level fusion and feature classification process to remove their respective limitations by complementing each other. The use of depth-separable convolution operations significantly reduces the number of operations and parameters required by the framework [9].

It discusses a new idea called "stability" in signature verification, which explains why a signature may vary slightly each time it is signed. The most consistent parts of a signature across multiple executions are considered the most important for verifying its authenticity. To incorporate this idea into signature verification, a new algorithm called the Stability Modulated Dynamic Time Warping has been developed, which compares the most similar parts of two signatures to calculate the distance between them. This algorithm has been tested on two datasets commonly used to evaluate signature verification systems and has shown to improve the accuracy of the current system and performs comparably to other top-performing signature verification systems. It explains how a signature is made up of small movements executed in a specific order to create a unique pattern, and the motor plan for the signature is independent of the body part used to sign. However, the specific execution may vary slightly due to factors such as visual and proprioceptive feedback, also highlights how different algorithms have been developed to exploit the concept of signature stability, including model based, feature-based, and data-based approaches. This new approach differs from others in that it explicitly represents motor plans in terms of stability regions and uses this information to evaluate signature similarity. The main goal of the study is to provide experimental evidence to support the idea that multiple executions of a signature should produce very similar trajectories, as the signing habits of a subject are encoded in the motor plan that has been learned [10].

III. WORKING PRINCIPLE

The Prototype is designed to detect forged signatures and verify its authenticity. Signature verification and forgery detection systems are utilized to authenticate and analyze signatures for determining their legitimacy. These systems are essential in industries like banking, legal, and government sectors where the verification of signatures is critical for maintaining document integrity and authenticity. The process of signature verification and forgery detection involves a series of steps. Firstly, the signature is acquired either by capturing it digitally or scanning a physical signature. The acquired signature is then converted into a digital format. Next, the signature undergoes pre-processing to enhance its quality and eliminate any distortions or imperfections that may affect the analysis. Techniques like noise removal and image enhancement are applied during this stage.

Following pre-processing, features are extracted from the signature. These features can be categorized as global and local features. Global features encompass overall characteristics such as stroke width, aspect ratio, and signature dimensions, providing information about the general shape and structure of the signature. On the other hand, local features focus on specific details within the signature, such as stroke curvature, pen pressure, and directional information, capturing the unique writing style and individual traits. The extracted features are then utilized to represent the signature in a numerical or mathematical form, often in the form of feature vectors or templates. Simultaneously, a reference signature database is created, containing genuine signatures for comparison. This database is populated during the enrollment process, where individuals provide their genuine signatures for future verification purposes.

The signature to be verified is compared with the reference signatures in the database using various comparison techniques, such as distance-based metrics or machine learning algorithms. Additionally, forgery detection techniques are employed to identify any attempts to forge or tamper with a signature. These techniques analyze inconsistencies between the questioned signature and genuine signatures, considering factors like stroke directions, pen pressure variations, or unnatural breaks in the signature. Based on the results of signature comparison and forgery detection, a decision is made regarding the authenticity of the signature. This decision can be binary, indicating whether the signature is genuine or forged, or it can provide a confidence score or likelihood estimation.

The accuracy and effectiveness of a signature verification and forgery detection system depend on factors such as the quality of the acquired signatures, robustness of the feature extraction algorithms, size and representativeness of the reference signature database, and sophistication of the forgery detection techniques employed. These systems play a crucial role in maintaining document integrity, preventing identity fraud, and facilitating secure transactions by accurately verifying the authenticity of signatures.

Signature verification and forgery detection systems can leverage Convolutional Neural Networks (CNNs) to analyze and classify signature images. CNNs, designed for processing grid-like data like images, can learn features from genuine and forged signature datasets. The trained CNN model extracts feature from questioned signatures and compares them with genuine signatures in a reference database. Additionally, CNNs can detect forgery by analyzing inconsistencies and irregularities in signature features. The decision regarding signature authenticity is made based on similarity measures and forgery detection analysis. CNNs enhance system efficiency by automatically learning and extracting relevant features, improving the overall effectiveness of signature verification and forgery detection.

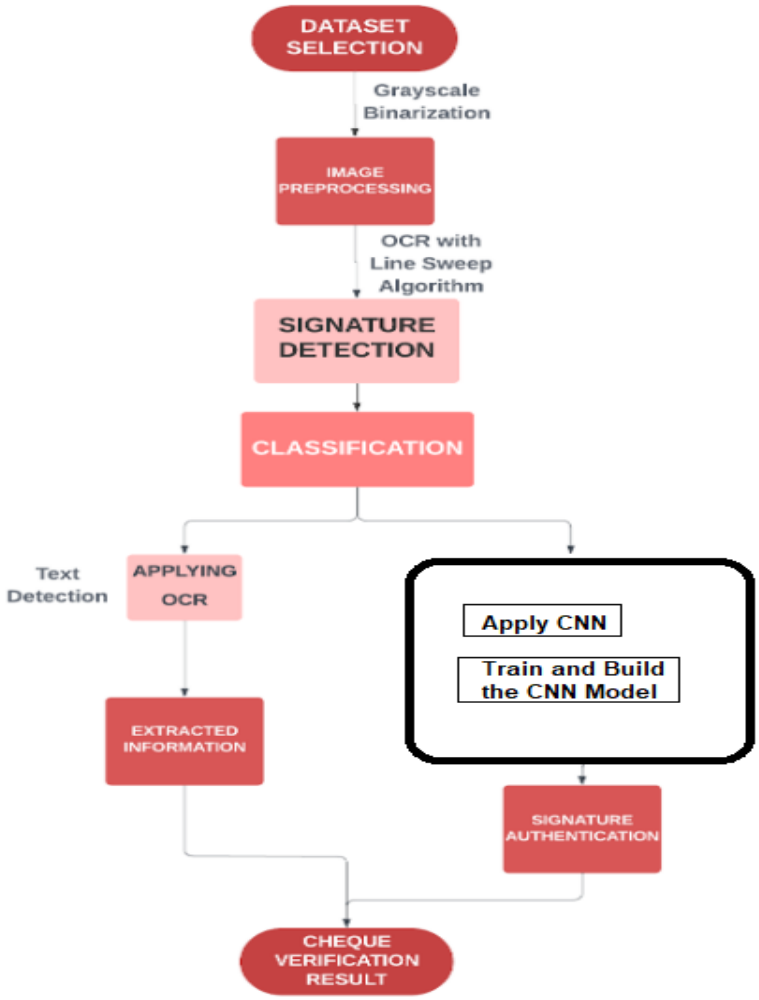

Figure 1: Workflow of the prototype

IV. IMPLEMENTATION

This prototype utilizes different concepts at different stages to achieve the needful. Firstly, we have used OCR algorithm to extract the signature from the cheque. OCR algorithms are applied to extract the textual information from documents or images that contain signatures; in this scenario, the signatures are extracted from cheques. This is done by scanning physical documents, capturing digital images, or targeting specific regions of interest within an image. The OCR feature allows the system to extract text from the image, enabling the detection of specific keywords or information such as IFSC codes or predefined keywords like "Please" or "Sign." This extraction process is achieved using the Tesseract OCR engine, which recognizes text in images. The OCR system analyzes the visual characteristics of the text, including the shapes of individual characters and their spatial relationships, to convert them into machine-readable text. Once the text is extracted, it is associated with the corresponding signature, establishing the connection between the textual elements and the signature itself. This linking process enables the system to establish the identity of the signer by extracting relevant information such as the signer's name, date, or other identifying details from the accompanying text.

In addition, OCR also allows for the analysis of the contextual information surrounding the signature including metadata related to the document, timestamps, or any other text present in the document. The system can also identify inconsistencies or anomalies that may indicate forgery attempts or irregularities in the signature verification process. OCR also contributes to forgery detection by assisting in the identification of text alterations or tampering attempts within a document or image. By comparing the extracted text with known reference texts or analyzing text characteristics, such as font consistency or irregularities, the system can detect forged or manipulated text elements. This can provide valuable insights into the overall integrity of the document and the signature itself.

Furthermore, the system incorporates machine learning algorithms to classify the image based on signature presence. The trained model leverages image features and patterns to determine whether the signature in the image is genuine or forged. This classification provides an additional layer of verification and helps in distinguishing between authentic and fraudulent signatures.

Next phase incorporates the line sweep algorithm in our prototype, the prototype iterates through each file in the directory, checking the file size to ensure it is a valid image. For valid files, it opens the image, converts it to grayscale, and performs thresholding to create a binary image. we use a line sweep algorithm; it identifies the start and end positions of the signature within the binary image. It then crops the original image based on the signature's bounding box and saves the cropped signature image in a separate directory. Finally, the code outputs the number of processed files and directs the user to the directory containing the cropped images. The line sweep algorithm allows the script to identify the bounding box or boundaries of the signature within the image. The algorithm iterates over the rows and columns of the binary image and looks for consecutive rows and columns containing white pixels, indicating the presence of the signature.

By performing the line sweep, the code can determine the start and end positions of the signature vertically and horizontally. This information is crucial for accurately isolating and cropping the signature region from the original image. The identified bounding box helps define the region of interest containing the signature, enabling further analysis or processing specific to the signature itself. The line sweep approach simplifies this task by systematically scanning the image and detecting consecutive rows and columns with white pixels, ensuring accurate and consistent cropping of the signature region across different images.

The line sweep algorithm also helps locating and isolating the signature region within an image. By systematically scanning the image, the algorithm identifies the start and end positions of the signature, enabling accurate localization and extraction of the signature region. This allows for further analysis, feature extraction, and comparison against genuine reference signatures or known patterns of forgery. The line sweep algorithm plays a crucial role in identifying discrepancies and anomalies that indicate potential forgery, contributing to the system's overall accuracy and effectiveness in verifying the authenticity of signatures.

Finally, the last phase of the prototype utilizes the Convoluted Neural Networks (CNN) concept. The concept of CNNs is suited for handling image-based data and here we employ dozens of signature images to be preprocessed for various different tasks. In this Signature Verification, the primary goal is to accurately differentiate between genuine signatures and forged ones. CNNs process it by automatically learning and extracting meaningful features from signature images, enabling the system to make informed decisions.

CNNs consist of multiple layers, including convolutional layers, pooling layers, and fully connected layers. These layers work together to detect patterns and features at different levels of abstraction within an image. In the context of signature analysis, the convolutional layers perform spatial filtering, capturing local patterns and textures that are relevant to distinguishing genuine signatures from forgeries. The pooling layers help reduce the spatial dimensions, aiding in capturing the most salient features while minimizing computational complexity.

Additionally, CNNs learn and extract hierarchical representations from signatures. The lower layers of the network learn low-level features such as edges and corners, while the higher layers learn more complex features like loops, strokes, and overall structure. By combining these learned features, the network can discern the unique characteristics of genuine signatures and identify inconsistencies or anomalies that indicate forgery.

The prototype consists of multiple layers, starting with a 2D convolutional layer followed by a max-pooling layer. The layers are then used to extract relevant features from the input images. Another set of convolutional and pooling layers is added to further enhance the feature extraction process. The Flatten layer is then uses to convert the multidimensional feature maps into a flat vector. Dense layers are added to perform classification, with the final layer using sigmoid activation for binary classification between genuine and forged signatures. To handle the image data, we rescale the image pixel values, applies shear and zoom transformations,

and performs horizontal flips on the training data. This augmentation helps in improving the model's ability to generalize and handle diverse signatures.

During the training phase, a large dataset of genuine signatures and forgery samples is used to train the CNN model. The model learns to recognize the discriminative features that differentiate genuine signatures from forgeries by adjusting the weights and biases of its layers. The training process enables the network to generalize and recognize genuine signatures accurately while detecting various types of forgery attempts. In the testing phase, the trained CNN model is put to test under unseen signature samples for verification and forgery detection. The prototype processes the input signature images through its layers, extracting relevant features, and produces a prediction or score indicating the likelihood of the input being genuine or forged. Based on the threshold or decision criteria set by the system, the model classifies the signature as genuine or identifies it as a forgery.

Overall, the code demonstrates the construction and training of a CNN model for signature forgery detection. It incorporates data augmentation techniques, handles image data using the ImageDataGenerator, and saves the trained model for future use. The visualization of training metrics helps in evaluating the model's performance and progress.

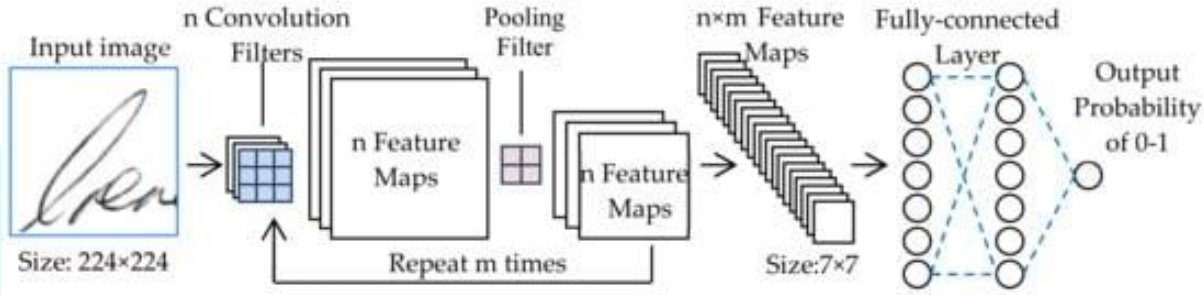

Figure 2: CNN Model

V. RESULTS

This is the graphical user interface (GUI) for the Signature Verification and Forgery Detection system, it is a simple and easy to use application; where not much training is required. The GUI is intuitive and user-friendly, allowing users to navigate and interact with the system effortlessly. The GUI has a clean and organized layout, featuring clear instructions and guidance at each step of the signature forgery detection process. Users are able to easily upload signature samples, and interpret the results. The system will leverage user-centered design principles, conducting usability tests and incorporating user feedback to optimize the interface for efficient and enjoyable user experience.

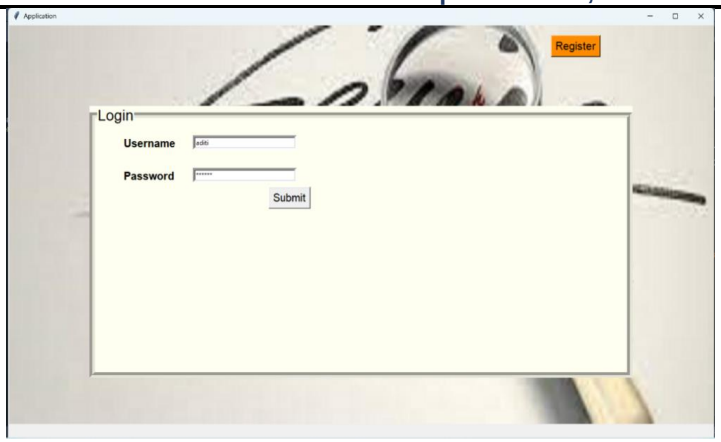

Figure 3: Login page

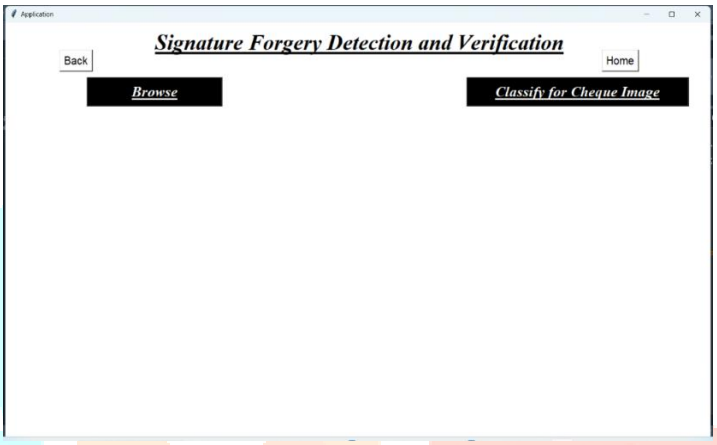

Figure 4: Main page

After logging into your details in the login page. You reach the main page for signature verification. Here the user can upload any cheque.

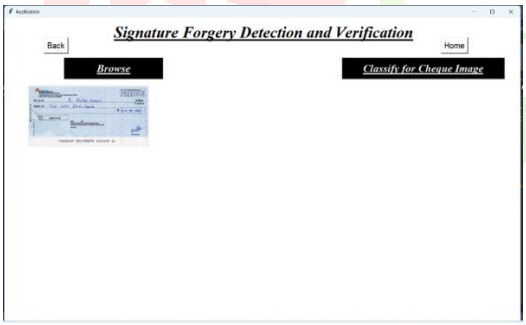

Figure 5: Uploading the cheque

Figure 6: Genuine Signature

After uploading the cheque, you will click on classify the cheque image whether it is genuine or forged.

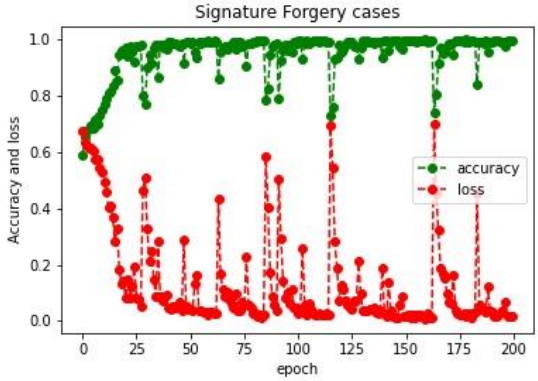

Figure 7: Graphical Representation of Accuracy and Loss

This is the graphical representation of the value_Accuracy and value_Loss observed in Signature Verification.

VI. Future Scope

[1] Incorporating multi-modal information could enhance the system's accuracy. By combining signature images with additional data sources such as text, timestamps, or user-specific behavioral patterns, the program can capture a more comprehensive understanding of signature authenticity.

[2] Integrating an online learning component into the system would enable continuous adaptation and improvement. As new data becomes available, the program can update its models and algorithms in real-time, allowing it to handle evolving forgery techniques and adapt to changing patterns in signature fraud.

[3] Leveraging blockchain technology could provide additional security and trust in the signature verification process. By storing signature metadata and verification results on a distributed ledger, the program can ensure immutability and transparency, making it more resistant to tampering and manipulation and hence making it accurate.

VII. Conclusion

Signature verification and forgery detection systems are crucial for ensuring the authenticity of signatures. These systems involve capturing signature images, preprocessing them, extracting features, creating a model, comparing signatures, and making a decision on their authenticity. Thus, we employ advanced techniques like deep learning and image processing techniques. Forgery detection systems focus on identifying signs of tampering or forgery within signatures. Therefore, we use a combination of automated verification and manual examination to ensure reliable results. Overall, these systems provide valuable assistance but still require human expertise and judgment for additional scrutiny and verification.

References

[1]. Sounak Deya, Anjan Duttaa, J. Ignacio Toledoa, Suman K.Ghosha, Josep Llados´a , Umapada Pal ”SigNet: Convolutional Siamese Network for Writer Independent Offline Signature Verification” Issue 30, September,2017.

[2]. Julita A., Fauziyah S., Azlina O., Mardiana B., Hazura H., Zahariah A.M.”Online Signature Verification System” Issued April 2009.

[3]. Shashi Kumar D R, K B Raja, R. K Chhotaray , Sabyasachi Pattanaik ”Off-line Signature Verification Based on Fusion of Grid and Global Features Using Neural Networks” Vol. 2(12), 2010.

[4]. Poonam Chaudhary and Vijay Kumar Singh,” Online Signature Verification: A Review”, Volume-3, Issue-2, Feb 2016.

[5]. Liyang Xie, Zhongcheng Wu, Xian Zhang, Yong Li, Xinkuang Wang “Writer-Independent Online Signature Verification based on 2-D Representation of Time Series Data using Triplet Supervised Network” June 2022.

[6]. S. Priya, A.K.R.N.Supreeth, K. Somesh, A. Hruday Kumar ” Signature Verification System using Different Algorithms”, Volume-8, Issue6S3, April 2019.

[7]. Tushara D, Shridevi Raddy, Shreya KM, Spoorthy Y,” Signature Verification System using Neural Networks”, NCCDS - 2021 Conference Proceedings.

[8]. Yuchen Zheng, Brian Kenji Iwana, Muhammad Imran Malik, Sheraz Ahmed, Wataru Ohyama, Seiichi Uchida “Learning the MicroDeformation by Max-Pooling for Offline Signature Verification” October 2021.

[9]. Chandra Sekhar Vorungunti, Viswanath Pulabaigari, Rama Krishna Sai Subrahmanyam, Gorthi, Prerana Mukherjee “OSVFuseNet: Online Signature Verification by Feature Fusion and Depth-Wise Separable Convolution Based Deep

Learning" October 2020.

[10]. Antonio Parziale, Moises Diaz, Miguel A. Ferrer, Angelo Marcelli "SM-DTW: Stability Modulated Dynamic Time Warping for Signature Verification" April 2019.