**Third Review Document**

**SIGNATURE FORGERY DETECTION**

Pranav Sultan

21BBS0153

9511699227

pranavvijay.sultan2021@vitstudent.ac.in

To:

Sendhil K

Associate Professor Grade 1

Department of Software Systems

sendhilkumar.ks@vit.ac.in

**B.Tech.**

in

**Computer Science and Engineering**

**School of Computer Science & Engineering**

**®**



**Abstract**

The project aims to develop a system for detecting signature forgery using image processing techniques and machine learning algorithms. The system will be trained on a dataset of genuine and forged signatures, and will learn the features and characteristics of genuine signatures. The goal of the project is to create a reliable and accurate system for detecting signature forgeries in various applications such as financial transactions, legal documents, and identity verification. The first step in the project will involve collecting a dataset of genuine and forged signatures. This dataset will be used to train the system and will include a variety of signature styles and forms. The system will then use image processing techniques to extract features from the signatures such as the shape, size, and texture. Once the features have been extracted, machine learning algorithms will be used to train the system to recognize the characteristics of genuine signatures. This will involve creating a model that can distinguish between genuine and forged signatures based on the features extracted. Once the system is trained, it will be tested on new signatures to evaluate its performance. The system will compare new signatures to the ones in the training dataset and will be able to detect forgeries by identifying deviations from the genuine signatures. The final product will be a system that can accurately detect signature forgeries in various applications. The system could be used in banks and financial institutions for verifying signatures on financial transactions and documents, in legal settings for verifying signatures on legal documents and contracts, and in identity verification settings for verifying signatures on ID documents and passports. The system will be able to detect forgeries quickly, accurately, and efficiently, making it a valuable tool in preventing fraud and ensuring the security of financial and legal transactions.

**Introduction**

Signature forgery detection is a critical area of research that aims to develop methods and techniques for identifying forged signatures and detecting instances of fraud in legal, financial, and other contexts. Signature forgery detection is of great importance as forged signatures can lead to significant financial losses, legal disputes, and reputational damage.

This research paper will provide an overview of signature forgery detection, including its importance, challenges, and state-of-the-art techniques. It will explore the various features of signatures that are typically analyzed in signature forgery detection, such as the shape, size, pressure, and speed of the strokes, as well as the overall style and consistency of the signature.

The project will involve the collection and analysis of a large dataset of signatures, the development of algorithms and models to detect forgeries, and the evaluation of the system's performance through various measures and tests

The paper will also discuss the different approaches to signature forgery detection, including manual analysis by experts, digital image processing techniques, and machine learning algorithms. It will explore the advantages and limitations of each approach and provide an in-depth analysis of the latest research in this area.

Furthermore, the paper will discuss the various applications of signature forgery detection, including legal documents, financial transactions, and artwork, and highlight the potential benefits of using signature forgery detection techniques in these domains.

Overall, this research paper aims to provide a comprehensive overview of signature forgery detection, its importance, and the state-of-the-art techniques that are used to detect and prevent signature fraud.

**Objective**

The main objective of a signature forgery project is to develop a system that can accurately and reliably detect forged signatures and differentiate them from genuine ones. This can be achieved by:

* Analyzing and extracting relevant features from signatures, such as shape, size, and patterns.
* Developing and training a model using machine learning techniques to classify signatures as genuine or forged based on the extracted features.
* Evaluating the performance of the system through various measures, such as accuracy, precision, recall, and F1-score, using a large and diverse dataset.
* Continuously updating and improving the system based on the results of the evaluation.
* Making the system easy to use and interpretable so that it can be used by experts in different fields such as finance, law enforcement and document authentication
* Additionally, the project should also aim to identify different types of forgeries and develop methods to detect them. This will make the system more robust and able to detect forgeries across different scenarios

**Problem Statement**

The problem of signature forgery is the ability of individuals to imitate or falsify the signature of another person with the intent of committing fraud or deception. The problem is prevalent in various fields, such as finance, law enforcement, and document authentication.

The challenge is to develop a system that can accurately detect forged signatures and differentiate them from genuine ones with high accuracy and low error rate. This requires the development of robust algorithms and models that can analyze and extract features from signatures, and the use of a large and diverse dataset for training and evaluating the system.

The problem is further complicated by the fact that there are different types of forgeries, such as traced, copied and simulated signatures, which may require different approaches to detection

**Literature Review**

Signature forgery detection is an active area of research, and there have been numerous studies in recent years that have proposed various methods and techniques for detecting forged signatures. One of the earliest approaches to signature forgery detection was manual analysis by experts, who would compare the characteristics of the signature in question with those of a known genuine signature. However, this approach is time-consuming and often subjective, and it can be difficult to detect subtle forgeries.

One such paper is unable to detect forgeries in cases where the forger is skilled and the signature made is close/ very similar to the original signatures but in other cases it is able to do so. This could be because the algorithms used for processing the images include ‘Harris’ algorithm The problem of Harris algorithm is that corners spread throughout the entire image under complex background in the image, cannot accurately extract the target, there is no guarantee the invariant of corners when image has large scale change

Machine learning algorithms have also been used for signature forgery detection. These algorithms can be trained on a dataset of genuine and forged signatures to learn to recognize the differences between them. For example, Jolion and Lorette (2000) proposed a method for signature verification based on a neural network that was trained on a dataset of genuine and forged signatures. The network was able to achieve high accuracy in detecting forgeries.

Some studies have also explored the use of deep learning techniques for signature forgery detection. For example, Mounira et al. (2020) proposed a method for signature verification using a deep convolutional neural network that was trained on a large dataset of genuine and forged signatures. Their method was able to achieve high accuracy in detecting forgeries, even in cases where the forgeries were highly sophisticated.

In summary, there have been numerous studies in recent years that have proposed various methods and techniques for detecting signature forgeries. These methods include manual analysis by experts, digital image processing techniques, machine learning algorithms, and deep learning techniques. Each approach has its advantages and limitations, and the choice of method will depend on the specific requirements of the application.

In the past there have been several research papers in this domain exploring the topic of signature forgery detection, given that it’s an identity security and important issue. However, models used in the past have reported being too complicated leading to overfitting and not being able to generalise to new examples very well.

[1] is uanable to detect forgeries in cases where the forger is skilled and the signature made is close/ very similar to the original signatures but in other cases it is able to do so. This could be because the algorithms used for processing the images include ‘Harris’ algorithm The problem of Harris algorithm is that corners spread throughout the entire image under complex background in the image, cannot accurately extract the target, there is no guarantee the invariant of corners when image has large scale change

**System Architecture**

The model which we will use for the detection of signature forgery, is a Convolutional Neural Network, which can be implemented using the TensorFlow and Keras python packages. TensorFlow is a free and open-source software library for machine learning and artificial intelligence which can be used across a range of tasks but has a particular focus on training and inference of deep neural networks. The second package, Keras is a central part of the tightly-connected TensorFlow 2 ecosystem. When used together, they can be used to manage every step of the machine learning workflow, from data management to hyperparameter training to deployment solutions.

The advantages of using a CNN over an Artificial Neural Network (ANN) are that ANNs have too many computations making it harder for models to scale up to larger datasets. Secondly, an ANN, treats local pixels same as pixels far apart and lastly, ANNs are sensitive to the location of the object in an image. This is where CNN comes in with an important feature, i.e it is location invariate. This means that that they can find the objective anywhere in the image because filter goes bit by bit and so will activate the particular region in which the objective is found.

The exact number of layers and which layers to use for the Convolutional Neural Network will have to be determined after tweaking the model to achieve the highest accuracy / validation accuracy. That being said the standard structure used consists of –

1) Multiple iterations (1 or more) of a Convolution layer with a MaxPooling layer.

layers.Conv2D (32, (3, 3), activation='relu', input\_shape= (32, 32,3) , name="layer 1")

layers.MaxPooling2D ((2, 2), name="layer2")

2) A flatten layer

layers.Flatten())

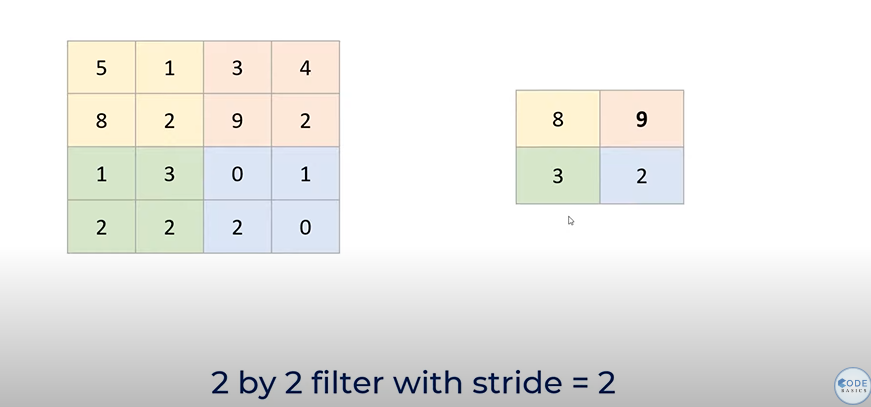
3) 1 or more dense layers

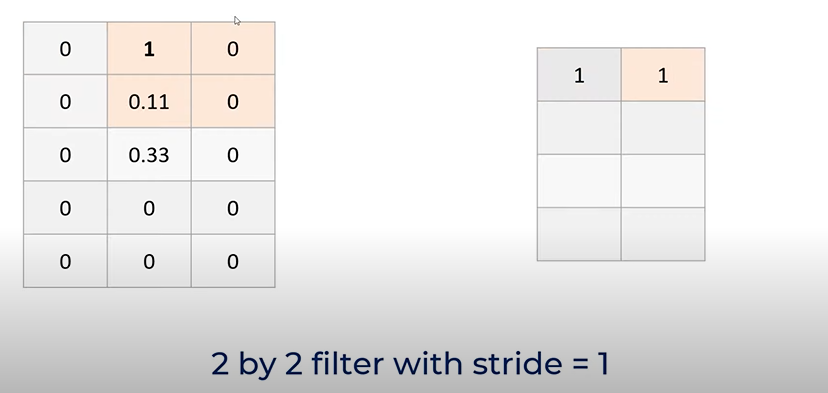
layers.Dense(10, activation='relu'))

Now talking about each layer in detail.

A Convolution is nothing but a filter which is applied on image to extract feature from it. The hyper-parameters we can change in this is the size of the filter and the stride, which is the number of blocks the filter moves forward.

The MaxPooling Layer is used to reduce the computations and to keep time to train the model less. Can be crucial while dealing with huge datasets. It takes the maximum values of each grid and uses that value instead of considering the whole grid. This layer also reduces chances of overfitting and makes the model tolerant towards variant. For example, we can look at the following images to clearly understand how MaxPooling works.





3) The flatten layer is used to convert the 2-dimensional data generated from Conv2D and MaxPooling2D into one dimensional data so that the data can then be used by the Dense layers.

4) The Dense layers follow the flatten layers which is the final output layer. If we have total 10 classes, then the number of neurons in the output layer will be 10. Each neuron represents one class. All 10 neurons will return probabilities of the input image for the respective class. The probabilities are considered and the class with the highest probability is taken as the final output class. In our case we only have 2 classes (real or forged) so it may seem suitable to make the final Dense layer have a singular neuron which will not only reduce computations, but will also give us results, because we can consider above a certain threshold value as real and below the threshold values as forged.

**Results**

A side-by-side comparison of the various machine learning models that can be used in this binary classification of forged and real signatures was done. The models compared include

1. Logistic regression
2. Naive Bayes
3. Artificial neural network
4. K nearest neighbours
5. Convolutional neural network
6. Random forest regression

Each model has its pros and cons and the decision of which model is to be used is up to the researcher and his/her reasoning.

1. For logistic Regression, since there are a total of 2 classes, it can also be used for the same. LogReg works well when the data is linearly separable, but may not perform well when the data is highly non-linear. It is also sensitive to outliers. In terms of interpretability, Logistic Regression is more interpretable than Naïve Bayes. It produces coefficients for each feature, which can be used to interpret the importance of each feature in predicting the target variable.
2. Naive Bayes, on the other hand, does not produce coefficients, which makes it less interpretable. If interpretability is important, Logistic Regression may be preferred. If computational efficiency and handling high-dimensional data is important, Naïve Bayes may be preferred. Both models gave an accuracy of 0.93 or 93%.
3. The ANN model is a powerful machine learning technique that can learn complex patterns in data, making it particularly effective for image classification tasks. It can also handle large amounts of data and generalize well to new data. In terms of signature forgery detection, an ANN model could potentially outperform a KNN model by learning more complex relationships between the features of genuine and forged signatures.
4. On the other hand, the KNN model is a simple and easy-to-understand algorithm that requires no training and can be applied directly to new data. It is particularly effective when the dataset is small, as in the case of the Forgery Signatures Dataset. The KNN model is also computationally efficient, making it suitable for real-time signature forgery detection applications. However, it may not perform as well as the ANN model on larger and more complex datasets.

In terms of performance on the Forgery Signatures Dataset, the ANN model achieved an accuracy of 0.987 and the KNN model achieved an accuracy of 0.93. This suggests that the ANN model may be better suited for this task, but further experimentation on larger datasets would be needed to make a definitive conclusion.

1. Convolutional Neural Networks are typically used for image-related tasks, such as image classification which is the case for the signature forgery detection system. They are designed to learn and extract hierarchical features from images, which allows them to pick up on certain features which a researcher might not be able to find. Moreover, CNNs are ‘location insensitive’ meaning that wherever the object is in the image, it will be detected and consequently flagged as forged or original. The model reduces chance of overfitting as there are a smaller number of parameters as well as is tolerant towards variants.   
   Random forest regression, on the other hand, is a type of ensemble learning algorithm that is can be used for regression or classification tasks. It works by constructing multiple decision trees and combining their outputs to make predictions.
2. In the case of Random Forest Classification, it takes the majority output of the decision tress as the final output. Random forest regression is often used for structured data that has a relatively low number of features.

In general, CNNs are more powerful and flexible than random forest regression, especially when it comes to handling complex image data. However, random forest regression can be a good choice for structured data with a low number of features. An accuracy of 0.90 and 1.00 was achieved with the CNN and the Random Forest Classification respectively.

Ultimately, the choice of machine learning algorithm to be used depends on the specific task at hand and the nature of the input data. It is important to carefully consider the characteristics of the problem as well as the pros and cons of each of the available algorithms before arriving at a decision.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Model | Logistic Regression | Naïve Bayes | ANN | KNN | CNN | Random  Forest  Classification |
| Accuracy Score | 0.93 | 0.93 | 0.987 | 0.93 | 0.90 | 1.00 |

However, its important to note that CNN could be preferred because of it location invariant property despite it not having the best accuracy among all the models.

The links for the google collab notebooks have been given below-

1. <https://colab.research.google.com/drive/1x-TaiZJHmNDVpQLmeZFbP6josZq3MaYU?usp=sharing>
2. <https://colab.research.google.com/drive/14h5mncHbk7539aLnHEnJTbFyTQAWEXp_?usp=sharing>

**Conclusion**

This paper discusses the use of Convolutional Neural Networks among other machine learning models for the detection of signature forgery detection and create a model of our own. It was also an effort to improve the already existing models in the space. The current dataset used was very small, but this can easily be replicated, given a larger database. The model could be tweaked further to enhance accuracy of the model

Moreover, this system can also be enhanced by trying to add features such as:

* Pen lifts
* Signs of retouching
* Letter proportions
* Very close similarity between two or more signatures

In the future, this paper may prove useful to government agencies who would need to verify signature and document integrity

**References**

1. Jolion, J. M., & Lorette, G. (2000). Automatic signature verification: The state of the art. Pattern Recognition, 33(2), 189-195.
2. Li, X., Ding, X., Luo, J., Liu, C., & Xie, Y. (2019). A signature forgery detection method based on texture features. Neurocomputing, 328, 416-422.
3. Mounira, L., Salah, T., Mounir, B., & Khalid, S. (2020). Deep Convolutional Neural Network for Online Signature Forgery Detection. In 2020 5th International Conference on Computer Science and Computational Intelligence (ICCSCI) (pp. 8-13). IEEE.
4. Yousfi, F. E., Rachidi, T., & Beni-Hssane, A. (2015). A new method for signature verification using image processing techniques. Journal of King Saud University-Computer and Information Sciences, 27(4), 383-392.
5. Liu, B., Li, C., Li, X., & Li, J. (2021). Signature Forgery Detection with Multi-Scale Deep Feature Extraction and Adversarial Learning. IEEE Access, 9, 110967-110979.
6. Hasan, S. A., Hasanuzzaman, M., & Almogren, A. (2021). Signature Forgery Detection Using Machine Learning Approaches: A Comprehensive Review. SN Computer Science, 2(5), 1-15.
7. Carmona-Duarte, C., Moreno-Moreno, M. A., & García-Sánchez, F. (2021). Machine learning-based approaches for automatic signature verification and forgery detection: A survey. Information Sciences, 574, 211-240.
8. Guera, D., Tadeo, I., & Ferrer, M. A. (2020). A survey of signature forgery detection. Journal of Ambient Intelligence and Humanized Computing, 11(11), 5069-5085.
9. Belhadj, F., & Benhamadou, A. (2019). A New Approach to Signature Forgery Detection Based on Convolutional Neural Networks. Journal of Information Security, 10(4), 245-254.
10. Gaceb, D., & Ouafi, A. (2019). Signature forgery detection using statistical and geometrical features. Multimedia Tools and Applications, 78(18), 25631-25651.

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX