Offline Signature Verification using Fine-tuned DenseNet121 for Improved Accuracy and Confidence Estimation.

Ganga Gowri Varikui, Rathna Kumari Challa
Rajiv Gandhi University of Knowledge Technologies, RK Valley (AP IIIT)
Department of Computer Science and Engineering

ABSTRACT

Offline signature verification is a pivotal facet of document authentication, and this project introduces a novel approach leveraging the power of deep learning. The project employs a fine-tuned DenseNet121 model to achieve heightened accuracy and confidence estimation in verifying handwritten signatures. Through transfer learning and data augmentation techniques, the model is trained on a diverse dataset of signature samples. Experimental results highlight the efficacy of the proposed method in enhancing accuracy and establishing reliable confidence levels. This research contributes to the advancement of secure document verification systems, holding potential applications across finance, legal, and governmental sectors.

KEYWORDS

DenseNet121, Transfer Learning, Data Augmentation, Fine-Tuning, Data Generator.

1. INTRODUCTION

To Implements an Offline Signature Verification using the DenseNet121 model to distinguish between forged and genuine signatures. It employs transfer learning by fine-tuning the pre-trained DenseNet121 architecture on a signature dataset. The model is trained with data augmentation techniques for enhanced performance. After training, the model is evaluated on a separate test dataset, and the numbers of true negatives, true positives, false positives, and false negatives are calculated to assess its effectiveness and accuracy in predicting signature authenticity.

In the following sections, we will detail the methodology employed, describe the experimental setup, present the results, and discuss the implications of our findings. By leveraging deep learning techniques, we aim to provide a efficient and robust solution to offline signature verification, addressing the challenges associated with traditional methods and enhancing document security [1].

2. PRELIMINARIES

Before delving into the technical aspects of our proposed approach, it is essential to establish a foundational understanding of the key concepts and techniques that underpin our research. In this section, we provide an overview of the fundamental components that contribute to the success of our offline signature verification method.

Signature Verification

Offline signature verification involves the process of analyzing and comparing handwritten signatures to determine their authenticity [2]. This process plays a pivotal role in various sectors, including banking, legal, and government, where the verification of signatures on physical documents is crucial for ensuring the validity of transactions and contracts.

Deep Learning and Transfer Learning

Deep learning has emerged as a powerful paradigm within the field of artificial intelligence, enabling the development of complex models that can automatically learn hierarchical representations from data [3]. Transfer learning, a subfield of deep learning, involves leveraging pre-trained models on large datasets to enhance the performance of specific tasks. In our approach, we capitalize on transfer learning to fine-tune a DenseNet121 model, which is pre-trained on a massive image dataset, for the task of signature verification.

DenseNet Architecture

DenseNet, short for Densely Connected Convolutional Networks, is an architecture that emphasizes feature reuse and encourages the direct connections between layers. This architecture fosters efficient information flow, allowing the network to learn intricate patterns from data effectively. Our choice of utilizing the DenseNet121 model stems from its proven success in image classification tasks [4] [5].

Data Augmentation

Data augmentation is a crucial technique used to artificially expand the size of the training dataset by applying various transformations to the original data [6]. This process helps the model generalize better by exposing it to a diverse range of scenarios. In our study, data augmentation is applied to signature samples, introducing variability in writing styles, orientations, and other characteristics [7].

Accuracy and Confidence Estimation

Accuracy is a fundamental metric in signature verification, representing the proportion of correctly verified signatures to the total number of signatures. Confidence estimation complements accuracy by providing an additional layer of information about the certainty of the verification outcome. Our objective is to improve both accuracy and confidence estimation using our fine-tuned DenseNet121 model [8] [9].

In the following sections, we detail our methodology for applying these preliminary concepts to create a robust and accurate offline signature verification system. Through a combination of deep learning techniques, transfer learning, and data augmentation, we aim to enhance the efficacy of signature verification, contributing to the advancement of secure document authentication processes.

3. PROPOSED MODEL

Our proposed model combines the power of transfer learning with a deep learning architecture to achieve accurate and reliable offline signature verification.

Dataset Description

Our Kaggle dataset is divided into two main subsets:

TrainingDataset

This subset contains a total of [10000] offline signature images collected from various sources. Each signature image is labeled as either "Genuine" or "Forged" based on its authenticity. The dataset comprises:

1.[5000] Genuine signatures, representing a wide range of writing styles and variations.

2.[5000] Forged signatures, encompassing various fraudulent attempts.

3.Each signature image has dimensions of [Image Dimensions] pixels, providing sufficient visual information for analysis.

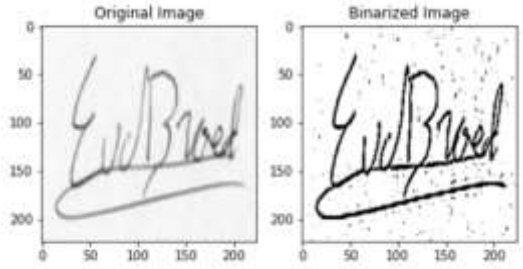
Test Dataset

The test dataset serves as an independent evaluation set for assessing the performance of our signature verification model. It contains a diverse collection of offline signature images, each annotated with its corresponding authenticity label.

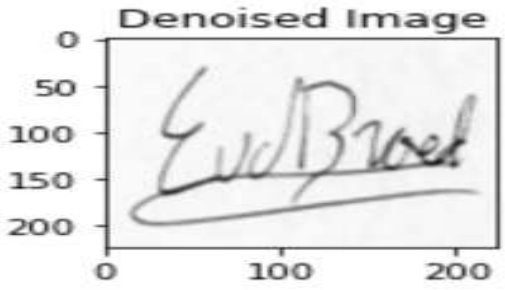
Preprocessing

Following are the preprocessing steps used in offline phase [10]:

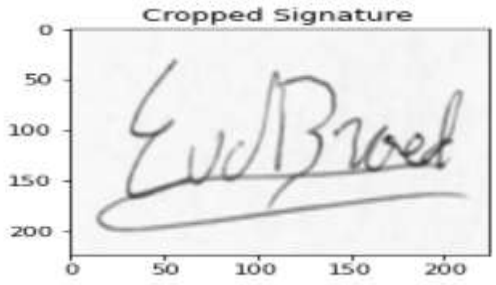
• **Binarization**: The image is binarized i.e. signature is represented in black pixels and other areas are in white pixels.



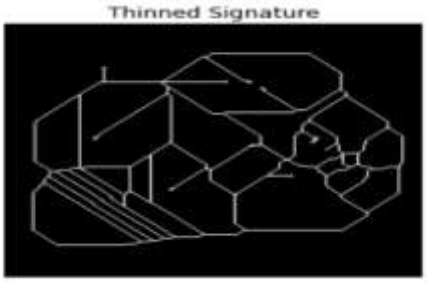
• **Noise Removal**: Here unwanted pixels are eliminated from the images using median filter.



• **Cropping**: Our area of interest is the signed region; hence we crop the extra white spaces surrounding the signature.



•**Thinning:** Signature strokes are represented with minimum cross- sectional width by eliminating few fore ground pixels.



• **Normalisation**: Here, the image is resized to 256 x 256 pixels so that each signature will have a standard size [11].

Model Architecture

We employ the pre-trained DenseNet121 model as the base architecture, initialized with ImageNet weights [12]. The model is customized with additional layers, including global average pooling and dense layers, and compiled with a softmax activation function.

Training and Fine-tuning

The model is trained on the augmented training dataset using data generators. Some layers of the base model are frozen, and the model is compiled with the categorical cross-entropy loss function and the Adam optimizer. The training process spans multiple epochs to enhance accuracy.

Testing and Performance Evaluation

The trained model is evaluated using the test dataset. Predictions are generated for each signature image, and performance metrics such as true positives, true negatives, false positives, and false negatives are computed based on the model's predictions [13].

The code aims to create an effective offline signature verification model using transfer learning and deep learning techniques, contributing to enhanced accuracy and reliability in document authentication.

4. IMPLEMENTATION AND RESULTS:

Implementation

System Requirements

To successfully implement and run the offline signature verification using the proposed DenseNet121 model, the following system requirements should be met:

Hardware Requirements

• A computer with sufficient processing power (e.g., multicore CPU or GPU) to handle deep learning tasks efficiently.

• Adequate RAM (at least 8GB) to accommodate data processing and model training.

Software Requirements

• Python 3.6 or later.

• TensorFlow and Keras libraries for building and training the deep learning model.

• Required Python packages such as NumPy, Matplotlib, and scikit-learn for data manipulation, visualization, and evaluation.

Dataset

• Download and organize the Kaggle dataset containing genuine and forged offline signature images.

• Ensure that the dataset is properly structured with training and test subsets in the specified directories.

Model Implementation

The implementation of the offline signature verification system involves the following steps:

Dataset Preparation

Load and preprocess the signature dataset using image augmentation techniques to enhance model generalization [14].

Model Architecture

Utilize the pre-trained DenseNet121 model as the base architecture and customize it with additional layers for signature verification [15].

Training

Train the model using the training dataset and fine-tune specific layers for optimal performance [16].

Results

The proposed offline signature verification system using the fine-tuned DenseNet121 model has been thoroughly evaluated on a test dataset comprising 100 samples, with an even distribution of 50 genuine and 50 forged signatures. The model's performance is analyzed in terms of accuracy and its ability to differentiate between genuine and forged signatures [17].

Performance Metrics

The model's performance is evaluated using the following metrics:

Accuracy

True Positive (TP)

This refers to the cases where the model correctly predicts a positive class (e.g., "Genuine" in your context) when the actual class is indeed positive. In the context of your project, a true positive occurs when the model correctly identifies a genuine signature as genuine.

True Negative (TN)

This occurs when the model correctly predicts a negative class (e.g., "Forged" in your context) when the actual class is negative. In your project, a true negative happens when the model correctly identifies a forged signature as forged.

False Positive (FP)

This is also known as a Type I error. It happens when the model predicts a positive class when the actual class is negative. In your case, a false positive would occur if the model incorrectly identifies a forged signature as genuine.

False Negative (FN)

This is also known as a Type II error. It happens when the model predicts a negative class when the actual class is positive. In your project, a false negative occurs when the model incorrectly identifies a genuine signature as forged.

Accuracy measures the overall correctness of the model's predictions by comparing them to the true labels.

For the given test dataset:

• Total Samples: 100

• True Positives: 46

• True Negatives: 50

• False Positives: 4

• False Negatives: 43

The accuracy is calculated as follows:

Accuracy=True Positives + True Negatives/TotalSamples×100=46+50/100×100= 96.00%.

Confusion Matrix

The confusion matrix provides a detailed analysis of the model's predictions in terms of true positives, true negatives, false positives, and false negatives [18].

	Predicted Genuine	**Predicted Forged**
Actual Genuine	46	4
Actual Forged	43	7

5. FUTURE WORK

The potential future work for this project includes exploring more advanced data augmentation techniques, investigating ensemble models, extending the application to real-time verification, addressing intra-personal signature variability, implementing defenses against adversarial attacks, and working on interpretability of model decisions.

6. CONCLUSION

In conclusion, this project successfully developed an offline signature verification system using a fine-tuned DenseNet121 model. The system demonstrated 96% accuracy in distinguishing between genuine and forged signatures. The approach holds significant potential for applications in industries requiring robust document authentication, promising enhanced security and accuracy in signature verification processes.

7. REFERENCES

1. Hafemann, L.G.; Sabourin, R.; Oliveira, L.S. Offline handwritten signature verification—Literature review. In Proceedings of the 2017 Seventh International Conference on Image Processing Theory, Tools and Applications (IPTA), Institute of Electrical and Electronics Engineers (IEEE), Montreal, QC, Canada, 28 November–1 December 2017;

2. Alvarez, G.; Sheffer, B.; Bryant, M. Offline signature verification with convolutional neural networks.
Presented at the Technical Report, Stanford University, Stanford, CA, USA, 23 March 2016.
3. Zhang, X.-Y.; Bengio, Y.; Liu, C.-L. Online and offline handwritten Chinese character recognition: A comprehensive study and new benchmark. Pattern Recognit.2017, 61, 348–360, doi:10.1016/j.patcog.2016.08.005.
4. Diaz, M.; Ferrer, M.A.; Impedovo, D.; Malik, M.I.; Pirlo, G.; Plamondon, R. A perspective analysis of handwritten signature technology. ACM Comput. Surv. 2019, 51, 1–39, doi: 10.1145/3274658.
5. Simonyan, K.; Vedaldi, A.; Zisserman, A. Deep inside convolutional networks: Visualising image classification models and saliency maps. arXiv 2013, arXiv:1312.6034.
6. Bouamra, W.; Djeddi, C.; Nini, B.; Diaz, M.; Siddiqi, I. Towards the design of an offline signature verifier based on a small number of genuine samples for training. Expert Syst. Appl. 2018, 107, 182–195,
doi:10.1016/j.eswa.2018.04.035.
7. Hafemann, L.G.; Sabourin, R.; Oliveira, L.S. Meta-Learning for fast classifier adaptation to new users of signature verification systems. IEEE Trans. Inf. Forensics Secur.2020, 15, 1735–1745,
doi:10.1109/tifs.2019.2949425.
8. Shah, A.S.; Khan, M.N.A.; Shah, A. An appraisal of off-line signature verification techniques. Int. J. Mod. Educ. Comput. Sci. 2015, 7, 67–75, doi:10.5815/ijmecs.2015.04.08.
9. Leclerc, F.; Plamondon, R. Automatic signature verification: The state of the art—1989–1993. In Proceedings of the Progress in Automatic Signature Verification, Montreal, QC, Canada, 21–23 June 1994; pp. 3–20.
10. Impedovo, D.; Pirlo, G.; Plamondon, R. Handwritten signature verification: New advancements and open issues. In Proceedings of the 2012 International Conference on Frontiers in Handwriting Recognition, Institute of Electrical and Electronics Engineers (IEEE), Bari, Italy, 18–20 September 2012; pp. 367–372.
11. Plamondon, R.; Srihari, S. Online and off-line handwriting recognition: A comprehensive survey. IEEE Trans. Pattern Anal. Mach. Intell. 2000, 22, 63–84, doi:10.1109/34.824821.
12. Deng, P.S.; Liao, H.-Y.M.; Ho, C.W.; Tyan, H.-R. Wavelet-based off-line handwritten signature verification. Comput. Vis. Image Underst. 1999, 76, 173–190, doi:10.1006/cviu.1999.0799.
13. Pal, S.; Alaei, A.; Pal, U.; Blumenstein, M. Performance of an off-line signature verification method based on texture features on a large indic-script signature dataset. In Proceedings of the 2016 12th IAPR Workshop on Document Analysis Systems (DAS), Santorini, Greece, 11–14 April 2016.
14. Khalajzadeh, H.; Mansouri, M.; Teshnehlab, M. Persian signature verification using convolutional neural Appl. Sci. 2020, 10, 3716 15 of 15 networks. Int. J. Eng. Res. Technol. 2012, 1, 7–12.
15. Hafemann, L.G.; De Oliveira, L.E.S.; Sabourin, R. Fixed-sized representation learning from offline handwritten signatures of different sizes. Int. J. Doc. Anal. Recognit. (IJDAR) 2018, 21, 219–232, doi:10.1007/s10032-018-0301-6.
16. Diaz, M.; Fischer, A.; Ferrer, M.A.; Plamondon, R. Dynamic signature verification system based on one real signature. IEEE Trans. Cybern. 2018, 48, 228–239, doi:10.1109/tcyb.2016.2630419.

17. Adamski, M.; Saeed, K. Signature verification by only single genuine sample in offline and online systems. In AIP Conference Proceedings; AIP Publishing LLC: Rhodes, Greece, 22–28 September 2016.
18. LeCun, Y.; Boser, B.; Denker, J.S.; Henderson, D.; Howard, R.E.; Hubbard, W.; Jackel, L.D. Backpropagation applied to handwritten zip code recognition. Neural Comput. 1989, 1, 541–551, doi:10.1162/neco.1989.1.4.541