International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 16, No.1 (Jul-24)

https://dx.doi.org/10.12785/ijcds/160121

Authentic Signature Verification Using Deep Learning Embedding With Triplet Loss Optimization And Machine Learning Classification

Andreas Christianto[1], Jovito Colin[1] and I Gede Putra Kusuma Negara[1]

[1]*Computer Science Department, BINUS Graduate Program - Master of Computer Science, Bina Nusantara University, Jakarta, Indonesia, 11480*

Received 9 Feb. 2024, Revised 9 Apr. 2024, Accepted 16 Apr. 2024, Published 1 Jul. 2024

Abstract: Various document types (financial, commercial, judicial) necessitate signatures for authentication. With the advancements of technology and the increasing number of documents, traditional signature verification methods encounter challenges in facing tasks related to verifying images, such as signature verification. This idea is further reinforced by the growing migration of transactions to digital platforms. To that end, the fields of Machine learning (ML) and Deep Learning (DL) offer promising solutions. This study combines Convolutional Neural Network (CNN) algorithms, such as Visual Geometry Group (VGG) and Residual Network (ResNet) or VGG16 and ResNet-50 specifically, for image embedding alongside ML classifiers such as Support Vector Machine (SVM), Artificial Neural Network (ANN), Random Forest, and Extreme Gradient Boosting (XGBoost). While the aforementioned solutions are usually enough, real life scenarios tend to differ in environment and conditions. This problem leads to difficulty and accidents in the verification process, causing the users to redo the process or even end it prematurely. To alleviate the issue, this study employs optimization methods such as hyperparameter tuning via Grid Search and triplet loss optimization to enhance model performance. By leveraging the strengths of CNNs, Machine Learning classifiers, and optimization techniques, this research aims to improve the accuracy and efficiency of signature verification processes while addressing real-world challenges and ensuring the trustworthiness of electronic transactions and legal documents. Evaluation is conducted using the ICDAR-2011 and BHSig-260 datasets. Results indicate that triplet loss optimization significantly improves the performance of the VGG16 embedding model for SVM classification, notably elevating the Area Under the ROC Curve (AUC) from 0.970 to 0.991.

Keywords: Signature Verification, Signature Authentication, Image Embedding, Triplet Loss, Machine Learning Classifiers

1. Introduction

In today's digital world, the need for strong and secure handwritten signature verification methods is more important than ever. Since every person has a distinctive signature that is mostly used for personal identification and the authentication of significant papers or legal transactions [1]. As online electronic transactions and digital documents become more prevalent, there's a growing demand for advanced ways to verify signatures. Traditional methods face challenges in accurately confirming signature authenticity, leading to a shift towards utilizing the latest technologies as a solution. The integration of deep learning and machine learning techniques into signature verification processes presents a promising solution to address the limitations of conventional approaches. Leveraging the capabilities of computer vision in conjunction with these advanced technologies holds the potential to drastically improve the accuracy, adaptability, and overall efficiency of signature verification.

Every day throughout the world, a great number of vital financial, commercial, and legal papers are signed, so validating signatures has become a critical issue that demands attention [2]. For the financial aspect of this issue, financial transactions increasingly migrate to digital platforms, the banking industry, in particular, faces the challenge of securing electronic signatures against sophisticated forgery attempts. The implementation and application of robust signature verification systems is imperative to safeguard the integrity of financial transactions, prevent identity theft, and ensure the trustworthiness of electronic documents. Furthermore, the real-world applications of this research extend beyond the banking sector. Government agencies, legal institutions, and various industries dealing with sensitive information can benefit from an advanced signature verification system. The proposed methodology aims to offer a versatile solution applicable across diverse domains where signature verification is a critical component of security protocols.

Recent advancements in deep learning have demonstrated significant breakthroughs in various computer vision tasks. Cutting-edge models, such as the VGG-16 architecture, showcase remarkable advancements for visual recognition tasks. VGG-16 is a robust model that consists of 16 convolution layers and is fully connected,

E-mail address: andreas.christianto@binus.ac.id, jovito.colin@binus.ac.id, inegara@binus.edu

which is usually used to recognize and classify images [3]. Not to mention, it has been pre-trained using the ImageNet dataset, which makes the model good for image classification tasks. With that in mind, incorporating VGG-16 into authentic signature verification tasks may potentially enhance the accuracy and efficiency of the verification process.

Furthermore, notable research efforts by Manish Bajpai [4] and Xamxidin et al [5] have contributed to the exploration of signature verification methodologies. Bajpai's research focuses on leveraging the VGGNet model for feature extraction in handwritten signature authentication. On the other hand, Xamxidin et al. propose an Improved Inverse Discriminative Network (IDN) to enhance signature verification accuracy. Leveraging the strengths of these methodologies presents an opportunity for further advancement in signature verification tasks. Specifically, utilizing methods like triplet loss optimization reduces input requirements, while employing deeper models like VGG-16 could extract more meaningful information from input images.

Therefore, to improve checking and verifying signatures, this study uses various methods to thoroughly assess how the combination of image embedding methods, such as VGG-16 and ResNet-50, triplet loss optimization, and different deep learning embedding models alongside machine learning classifiers perform on the task of signature verification, such as Support vector machine (SVM), Random Forest (RF), Artificial Neural Network (ANN), and XGBoost (XGB). These methods are selected precisely to give an overview of the performance of the combination of the deep learning embedding models and machine learning classifiers. The main workflow / architecture of this work can be seen in Figure 1.

In Section 2, we review relevant literature and previous works related to signature verification methodologies, discussing their strengths and limitations. Section 3 presents the theoretical background and methodologies employed in our research, including deep learning embedding models, machine learning classifiers, and optimization techniques such as triplet loss. In Section 4, we propose our novel methodology, which integrates various image embedding methods, optimization techniques, and machine learning classifiers to enhance signature verification accuracy. Section 5 details the experimental setup, including dataset selection, model training parameters, and evaluation metrics. In Section 6, we present the results of our experiments and provide a comprehensive discussion of the findings, including insights into the performance of different methods and techniques. Finally, Section 7 concludes the paper by summarizing the key findings, highlighting contributions, and outlining potential avenues for future research in signature verification.

2. Related Works

The process of handwritten signature verification has gone through drastic changes with the integration of deep learning algorithms. In the pursuit of enhancing accuracy and reliability, researchers have delved into the world of machine learning, particularly the utilization of deep learning techniques. This literature review section explores some research papers, each contributing distinct methodologies and insights to the overarching theme of authentic signature verification through deep learning algorithms.

Engin et al [6] addresses the challenge of offline signature verification in real-world scenarios, particularly focusing on a banking context where customers' transaction request documents with occluded signatures are compared to their clean reference signatures. Unlike controlled datasets used in previous research, real-world signatures can include various occlusions such as stamps, seals, ruling lines, and signature boxes, leading to high intra-class variations. The proposed methodology comprises two main components, a stamp cleaning method based on CycleGAN and a signature representation method based on Convolutional Neural Networks (CNNs). The experiment results indicate a 76.8% accuracy when employing VGG-16 for signature representation along with the CycleGAN-based cleaning method, which is better compared to the 75

Poddar et al [7] introduces a novel method for signature recognition and forgery detection while considering the challenges associated with signature verification due to the variability introduced by individual writing styles and environmental factors. The proposed approach employs Convolutional Neural Networks (CNNs) and the Crest-Trough algorithm for signature verification while employing the Harris Algorithm and the Surf Algorithm for forgery detection. In the experimental results, the proposed signature recognition system achieves a high accuracy of 94% and an accuracy range of 85-89

Lu et al [8] introduces a methodology for handwriting identification that integrates both dynamic and static features to enhance the accuracy of signature identification, particularly in the context of forged signatures. The study establishes a Chinese signature forged handwriting database, which contains 44 signatures from different signers, collected from a dot matrix digital pen tool. The data collection involves offline images and online data, capturing information such as X and Y coordinate points, pressure, timestamp, and pen up-down marks. For classification, the study employs machine learning algorithms, such as Support Vector Machine (SVM), and deep learning algorithms, such as Convolutional Neural Network (CNN). The best results for the proposed method are 92.2% and 94.4% for SVM and CNN respectively.

A research paper by Manish Bajpai [4] focused on enhancing the accuracy of handwritten signature authentication using the VGGNet model, specifically for feature extraction, along with hyperparameter tuning. The final experiment, which produced the best result, reached a testing accuracy of 95% for detecting genuine and forged handwritten signatures. The conclusion from the experiment's result emphasizes the critical role of hyperparameter optimization and highlights that a faster learning rate does not necessarily enhance efficiency.

Borse et al [9] aims to implement a handwritten signature verification model using machine learning and deep learning to distinguish between genuine and forged

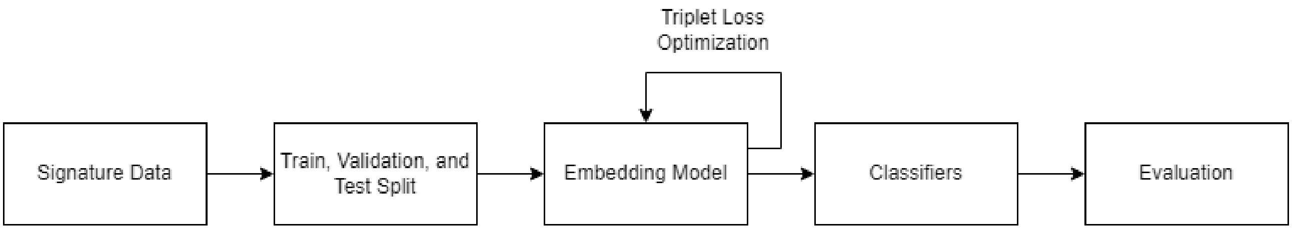

Figure 1. Overall System Architecture

signatures in order to ensure normalizing signature images for accurate comparisons and improving the accuracy of detecting correct signatures by utilizing deep learning models and multiple machine learning models such as Multilayer Perceptron (MLP), Support Vector Machine (SVM), and Random Forest (RF) models. The MLP model outperformed SVM and RF models, achieving an accuracy of 95.4%. The MLP model's effectiveness and robustness suggest its potential in various applications, particularly in government offices where handwritten signatures play a crucial role in approval and authentication processes.

Melhaoui and Benchaou [10] fixates on the challenges and methodologies associated with offline signature recognition systems due to its complexity compared to the online mode. The study utilizes a proprietary signature database, containing 240 signatures from 12 individuals. Feature extraction methods, including Histogram of Oriented Gradients (HOG), Profile Projection (PP), and Loci, are discussed. Classification methods, specifically FMMC and K Nearest Neighbors (KNN), are employed for evaluating the recognition system performance. The proposed system, through combining HOG features and FMMC classification, achieves a recognition rate of 96

Xamxidin et al [5], conducted the study on a different approach and a much broader multilingual dataset to test its effectiveness in signature diversity. The paper introduces an Improved Inverse Discriminative Network (IDN) to enhance the identification of genuine and false signatures to address the challenges in said task, such as sparse signature information, language diversity, and the arbitrary nature of signature styles. The experiments conducted on this research paper involves testing the proposed method on various datasets, including Chinese, Uyghur, Bengali, and Hindi signatures. The conclusion that is reached based on the results of the experiments emphasizes that the proposed IDN model effectively improves the accuracy of signature verification for single and mixed languages with an ACC of 92.40% using a Chinese dataset, an ACC of 92.96% using a Uyghur dataset, and an ACC of 96.33% using a dataset mixed with both Chinese and Uyghur languages.

A paper by Muhtar et al [2] centers on the critical task of handwritten signature verification, emphasizing its significance in authenticating crucial financial, commercial, and judicial documents globally. The experimental methodology employs the ResNet18 network and introduces the Convolutional Block Attention Module (CBAM) to improve the model. The proposed method, FC-ResNet, optimizes the ResNet18 structure for size while introducing CBAM in the residual block to better learn correlations between different feature channels and spatial positions. The study reports an accuracy rate of 96.21% on the CEDAR dataset and 96.41% on the Uyghur language dataset, demonstrating the method's effectiveness for signature data with few samples and its ability to accurately identify signature samples across languages.

In addition, Lopes et al [11] conducts signature verification using a modified version of the AlexNet deep learning model implemented through TensorFlow. This model is designed to recognize and verify individual signatures, marking potential forgeries for further manual verification. In the binary classification model test, multilayer perceptron (MLP) serves as a binary classifier for signature/non-signature test data, achieving an accuracy of 98.4% and an F1 score of 98.3% on test data. However, this model only confirms the presence or absence of a signature without verifying its authenticity. The paper concludes by underlining the importance of handwritten signatures in attendance verification. It suggests that the proposed methods, including the MLP classifier and CNN model, offer a reliable solution for automating signature verification.

Aljrami et al [12] address the critical task of signature verification and forgery detection, distinguishing between static (offline) and dynamic (online) methods. The significance of handwritten signatures as a widely accepted personal attribute for identity verification in various sectors is highlighted. The paper introduces the use of Deep Convolutional Neural Networks (CNNs) for both writer-independent feature learning and writer-dependent classification. The proposed methodology considers handwritten signatures as behavioral biometrics, acknowledging the changes in an individual's signature over time. The authors demonstrate the effectiveness of their approach by showcasing accuracy and loss plots for different dataset split ratios. The model achieves its highest accuracy rate of 99.7% on the validation dataset with an 8:2 dataset split, suggesting its efficacy in signature verification.

In summary, the exploration of various methodologies in handwritten signature verification through deep learning algorithms has revealed significant advancements in the field. The discussed papers contribute diverse approaches, each addressing specific challenges and introducing innovative techniques. Engin et al. emphasize the complexity of real-world scenarios, achieving a commendable accuracy of 76.8% by combining CycleGAN and VGG-16. Poddar et al. present a novel method achieving 94% accuracy for signature recognition and 85-89% for forgery detection. Lu et al.'s integration of dynamic and static features yields promising results of 92.2% and 94.4% for SVM and CNN, respectively. Notably, Aljrami et al. showcase the highest accuracy of 99.7%, emphasizing the significance of dynamic methods and considering signatures as behavioral biometrics. While each method demonstrates strengths, the choice of the most effective approach may depend on specific use cases and dataset characteristics. Overall, this review highlights the evolving trends in deep learning-based signature verification, with a continued emphasis on addressing real-world challenges and improving accuracy in various contexts.

3. Theories and Methods

A. VGG16

In the realm of computer vision, around mid-2010s, some convolutional networks were able to achieve high accuracy on ILSVRC classification and localisation tasks. Not only that, they are also tested against other image recognition datasets and were able to achieve excellent performance [13]. These convolutional networks were later known as VGGNet, a family of convolutional networks. VGGNet introduced a novel architecture characterized by its depth and capability to extract intricate features from images.

One of the members of VGGNet, VGG16, stands out as an excellent candidate for image recognition tasks. In the training process, VGG16 receives fixed-size 224x224 RGB images. The architecture consists of a stack of convolutional layers while primarily utilizing 3x3 filters for capturing directional information. Aside from that, 1x1 convolutional filters are employed in one of the configurations as linear transformations of input channels. The fixed convolutional stride of 1 pixel and spatial padding ensure that the spatial resolution is preserved after convolution. Through five max pooling layers, spatial pooling is introduced, each operating over a 2x2 pixel window with a stride of 2.

Following the convolutional layers, VGG16 employs three Fully-Connected (FC) layers with the channels of 4096, 4096, and 1000 respectively. The final layer, which exists after the FC layers, employs a soft-max activation to convert the raw output into probability between the values of 0 and 1. It is also important to note that all hidden layers incorporate the rectification non-linearity (ReLu) which enhances the network's capability to learn intricate features.

To summarize, the VGGNet algorithm is capable of extracting features well [14], hence could be beneficial for image embedding in handwritten signature verification. In this case study, the signature images act as inputs to the network, and the deep layers of VGG-16 capture intricate patterns and features inherent in genuine signatures.

B. ResNet-50

Due to the difficulty of training deep neural networks. A residual learning framework was made to ease the training process and to overcome the degradation problem in learning. The residual learning framework, known as ResNet, presented a novel framework for residual learning [15].

ResNet architectures are widely known for their use of residual blocks, which include skip connections to facilitate the flow of information through the network. One of the popular variants of the ResNet architecture is ResNet-50. This particular version of the architecture consists of 50 layers. A residual block consists of three layers, the first layer of the block reduces dimensionality with a 1x1 kernel, the second employs a 3x3 kernel, while the third layer restores the original dimensionality. This design alleviates the vanishing gradient problem which enables the training of exceptionally deep networks.

Following the residual blocks, Global Average Pooling (GAP) is employed. This produces a compact 1x1 feature map. Afterwards, an FC layer with 1000 nodes is employed. The final layer utilizes a softmax activation function, converting the raw output scores into a probability distribution over the classes.

In summary, ResNet-50 serves as a testament to the benefits of residual learning, facilitating the training of exceptionally deep neural networks while maintaining interpretability. In this case study, the signature images act as inputs to the network and ResNet-50 obtains the intricate patterns and features from the images.

C. Triplet Loss

To further improve the embedding models, an additional method, called triplet loss, was introduced. Originally, this method was made to improve face recognition tasks by enforcing a margin between each pair of faces from one person to all other faces. This allows the faces of one identity to stay near each other, while still enforcing the distance to other identities [16]. The steps to the triplet loss method are triplet selection, obtaining the embeddings through convolutional neural networks, and utilizing the loss function.

In this case study, triplet loss is utilized in order to enhance the discernment between genuine and forged signatures. This metric optimizes the arrangement of embeddings in the feature space. In the context of handwritten signature verification, triplet loss ensures that the distance between embeddings of genuine signatures is minimized, while the distance between genuine and forged signature embeddings is maximized. This addition may potentially improve the model's ability to create compact clusters for genuine signatures and increase separation from forged signatures.

D. Signature Verification Methods

In this section, we introduce the machine learning models proposed as classifiers for the task of handwritten signature verification using advanced deep learning techniques. The primary objective is to establish robust and accurate verification methods capable of distinguishing between genuine and forged signatures. Each classifier utilized deep learning architectures such as VGG-16 and

ResNet-50, and triplet loss for enhanced feature extraction. The following subsections delve deeper into each proposed classifier, illustrating their beneficial attributes and contributions to the overall signature verification framework.

1) Support Vector Machine (SVM):

Support Vector Machine (SVM) is a machine learning algorithm that has shown a good learning ability and generalization ability in classification, regression and forecasting [17]. This algorithm operates by finding an optimal hyperplane in a high-dimensional space that effectively separates data points belonging to different classes. In the field of handwritten signature verification, SVM could play a pivotal role in distinguishing between genuine and forged signatures. The algorithm could be integrated with advanced techniques such as VGG-16 image embedding and triplet loss to enhance the accuracy and reliability of the signature verification process.

SVM is particularly fitting for handwritten signature verification due to a couple noticeable attributes. First of all, the basic idea of SVM is to translate the input vector into a high-dimensional space by nonlinear transformation, and then create the best classification surface in said space [18], causing it to be effective in high-dimensional spaces and suited for situations such as where signature data is transformed into complex feature vectors using methods like VGG-16 and ResNet-50. Moreover, SVM is resilient to overfitting, which is essential for dealing with limited training data in signature verification tasks. Furthermore, its ability to handle non-linear decision boundaries through kernel functions provides flexibility in capturing intricate relationships within signature data. In closing, Vapnik and Cortes first proposed the support vector machine (SVM) for binary classification in 1995 [19]. Since signature verification is inherently a binary classification problem which is distinguishing between genuine and forged signatures, SVM's natural binary nature aligns well with the task at hand.

In conclusion, the integration of SVM with triplet loss optimized VGG-16 and ResNet-50 image embedding for handwritten signature verification capitalizes on the strengths of each component, resulting in a robust and accurate system for authenticating signatures across various applications and industries.

2) Artificial Neural Network (ANN):

Artificial Neural Network (ANN) is an algorithm that resembles or mimics the biological human brain functions to accomplish a given task [20]. Comprising interconnected nodes or artificial neurons organized into layers, ANNs are designed to learn and recognize complex patterns within data. These networks consist of input, hidden, and output layers, with weighted connections between neurons. During training, the network adjusts these weights to optimize its ability to capture intricate relationships, enabling ANNs to excel in tasks like pattern recognition, classification, and decision-making [21].

The operation of an ANN involves a feedforward and backward propagation process. During feedforward, input data is processed through the network, and the output is computed based on weighted sums and activation functions. The calculated output is then compared to the desired output, and the network's error is computed. Backward propagation entails adjusting the weights in the direction opposite to the error gradient, a process repeated iteratively until the network converges to a state of accurate predictions [22].

The characteristics and features of ANN are shown to be a fitting algorithm for handwritten signature verification. Beginning with how handwritten signatures exhibit intricate and unique patterns, and ANNs, with their ability to capture complex non-linear relationships, are well-suited for recognizing and learning these patterns [23]. Furthermore, ANN has proven its adaptability to high-dimensional feature spaces. In handwritten signature verification, features extracted from signature images, such as those obtained from deep learning architectures like VGG-16, often result in high-dimensional spaces. ANNs can effectively operate in these spaces, providing a suitable framework for processing and learning from the intricate details present in signature images [13]. Lastly, the diversity in signature styles demands a model that can dynamically adapt to different characteristics. ANNs, by learning and adjusting weights during training, can adapt to various signature styles, enhancing the model's ability to accurately verify signatures across a range of writing styles [24].

Artificial Neural Networks, as a fitting algorithm for handwritten signature verification, bring essential capabilities such as handling complex patterns, adaptability to high-dimensional feature spaces, learning from limited data, and dynamic adaptation to diverse signature styles. These inherent characteristics make ANNs a powerful tool for accurately distinguishing between genuine and forged signatures in various applications, providing a robust and versatile solution for signature verification tasks.

3) Random Forest:

Random Forest is an ensemble learning algorithm that excels in both classification and regression tasks. In fact, they are considered to be one of the most accurate general-purpose learning techniques available [25]. The algorithm constructs multiple decision trees during the training phase, with each tree trained on a random subset of the training data and features. This introduction of diversity and randomness is crucial in preventing overfitting, contributing to the model's robustness and adaptability. In classification tasks, the final prediction is determined by aggregating the predictions of individual trees through a voting mechanism, while in regression tasks, the predictions are averaged.

The construction of a Random Forest involves the creation of numerous decision trees, each independently trained on a subset of the training data and features. The process, known as bagging (Bootstrap Aggregating), ensures that each tree is unique, contributing its predictions to the overall ensemble. . In order to grow these ensembles, often random vectors are generated that govern the growth of each tree in the ensemble [26]. The ensemble approach enhances the model's predictive accuracy and generalization to new data, making Random Forest a powerful tool for various machine learning problems.

In the context of handwritten signature verification, where signature images are often transformed into high-dimensional feature spaces, Random Forest has the potential and beneficial characteristics to be effective. Techniques like VGG-16 and ResNet-50 image extraction generate complex feature representations, and Random Forest's ability to handle such intricacies is essential in distinguishing between genuine and forged signatures.

The ensemble learning nature of Random Forest is particularly beneficial in signature verification. By combining predictions from multiple decision trees, the model captures the nuanced patterns and variations present in different signatures, leading to enhanced accuracy in the verification process. Random Forest is known for its versatility and this approach has proved its high accuracy and superiority with imbalanced datasets [27]. Since handwritten signatures exhibit diverse styles, Random Forest's ability to handle this variability makes it adaptable to the inherent complexities of signature verification. The ensemble of trees allows the model to learn and generalize across different signature characteristics, providing a versatile solution.

Random Forest emerges as a fitting algorithm for handwritten signature verification due to its robustness, accuracy, and adaptability to high-dimensional feature spaces. The ensemble learning approach, combined with the algorithm's ability to handle diverse and complex data, positions Random Forest as a reliable solution for distinguishing between genuine and forged signatures in various applications.

4) Extreme Gradient Boosting (XGBoost):

In closing, this study also includes XGBoost, short for Extreme Gradient Boosting, is a powerful machine learning algorithm that belongs to the family of gradient boosting methods. Developed to address limitations of traditional gradient boosting techniques, XGBoost is renowned for its efficiency, speed, and high predictive accuracy. It leverages an ensemble of weak learners, typically decision trees, to iteratively optimize a cost function, enhancing its ability to model complex relationships within data [28].

XGBoost combines the strengths of boosting algorithms and regularization techniques to improve model performance. During training, weak learners are added sequentially, each correcting the errors of its predecessors. The use of decision trees as base learners, coupled with regularization terms in the objective function, prevents overfitting and enhances the model's generalization ability [29]. Additionally, XGBoost incorporates features like parallel processing, handling missing values, and incorporating user-defined loss functions, making it a versatile and customizable algorithm [30].

XGBoost possesses certain characteristics and features that make it susceptible to handwritten signature verification. Beginning with how XGBoost handles high-dimensional feature space. In handwritten signature verification, feature extraction methods often result in high-dimensional spaces. XGBoost is well-suited to operate in these spaces, efficiently handling the intricate and complex features extracted from signature images [31]. In addition, XGBoost's optimized gradient boosting algorithm enhances accuracy by iteratively improving the model's predictive performance. This iterative nature is particularly beneficial when learning and capturing the subtle nuances present in handwritten signatures [28]. Since handwritten signature data may contain noise and variations. XGBoost's resilience to noisy data ensures that the model can discern genuine signatures from forged ones, even in the presence of irregularities [32]. Finally, XGBoost's adaptability to diverse data types allows it to effectively handle the varying signature styles encountered in handwritten signature verification tasks, making it a fitting choice for applications where different writing styles must be accommodated [33].

XGBoost emerges as a fitting algorithm for handwritten signature verification due to its ability to handle high-dimensional feature spaces, optimized gradient boosting for improved accuracy, robustness to noisy data, and adaptability to diverse signature styles. The algorithm's efficiency, speed, and customizability contribute to its suitability for real-world applications, making it a valuable tool for accurately distinguishing between genuine and forged signatures.

4. PROPOSED METHODS

A. Dataset

The dataset used in this case study consists of two signature datasets taken from kaggle. They are the ICDAR-2011 signature dataset and BHSig260-Bengali signature dataset. The dataset contains 64 and 100 individuals respectively. Each of these individual's signatures produce multiple images which are divided into two groupings which are genuine signatures and forged signatures. An individual in the ICDAR-2011 dataset contains an average of 14 genuine signatures and 12 forged signatures while an individual in the BHSig260-Bengali dataset contains an average of 24 genuine signatures and 30 forged signatures. The ICDAR-2011 dataset was chosen because it was used in a competition while the BHSig260-Bengali dataset was chosen to compare the proposed method's capability against a larger dataset. The samples of these datasets can be seen in Figure 2.

B. Preprocessing

In this experiment, the images are resized to a uniform size so that they can fit inside the embedding models. The size of the images, which were in the range of 200 by 200 to 400 by 400, are shrunken down to 64 by 64. In addition, The dataset is split to a ratio of 60/20/20 with stratification for training, validation, and testing data respectively. This split is chosen to balance between having enough data for training the models effectively while also ensuring robust evaluation and validation of the models' performance. The largest portion of the dataset is allocated for training the models. A majority of the data is needed for training to ensure that the models can learn meaningful patterns and representations from the data. With more data for training, the models have a

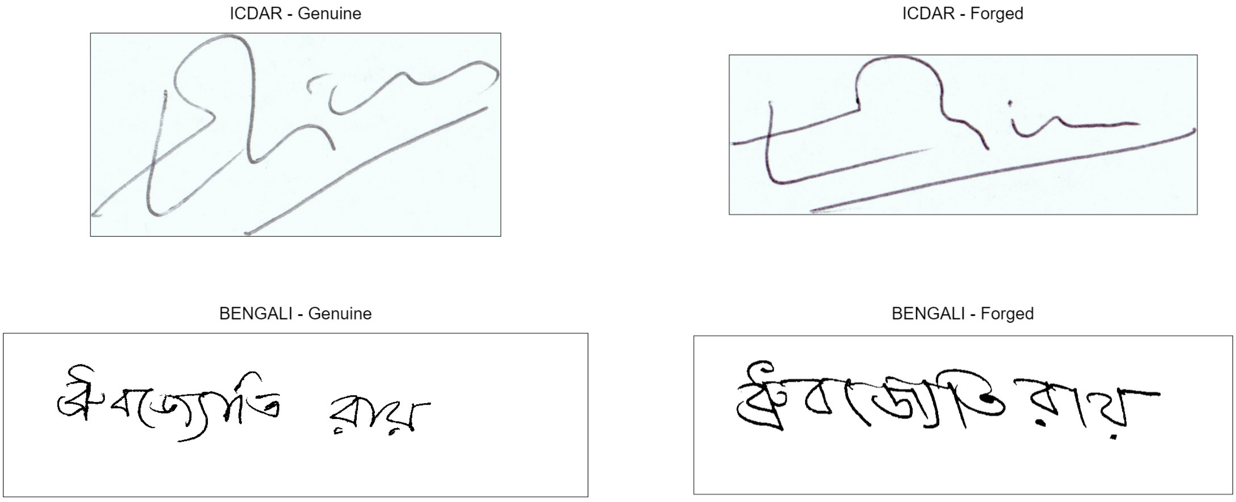

Figure 2. ICDAR-2011 and BHSig260-Bengali Dataset Visualization

TABLE I. Evolution of test-bed clusters

Dataset	Train	Validation	Test
ICDAR	857 images	391 images	41 images
BENGALI	3200 images	1100 images	1100 images

better chance of generalizing well to unseen examples and avoiding overfitting. A 60% allocation provides a substantial amount of data for training while leaving enough for validation and testing. A smaller portion of the dataset is set aside for validation. This portion is used during the training process to monitor the model's performance and adjust hyperparameters accordingly. The validation set helps in tuning the model's parameters to improve its performance without introducing bias from the testing data. A 20% allocation is sufficient for validation purposes while still allowing for effective model tuning. The remaining portion of the dataset is reserved for testing the trained models. This independent dataset is crucial for evaluating the final performance of the models and assessing their generalization capabilities on unseen data. By keeping a separate testing set, it ensures that the evaluation metrics are reliable and not influenced by the training or validation process. Allocating 20% of the data for testing provides a substantial sample size for robust evaluation. Overall, the 60/20/20 split strikes a balance between effective training, validation, and testing of the models, ensuring that they can learn meaningful patterns from the data, generalize well to new examples, and provide reliable performance metrics. Additionally, the use of stratification ensures that each class or category within the dataset is represented proportionally across the training, validation, and testing sets, reducing the risk of bias in the evaluation process. The following split is executed to each type of sample for a single instance. Lastly, the final data count of this research can be seen in Table I.

C. Feature Extraction for Image Embedding

This study utilizes the strength of Convolutional Neural Network, specifically the VGG-16 and ResNet-50 models, to extract features from signature images for verification purposes. These models take the signature images dataset as inputs and the pre-trained VGG-16 and ResNet-50 models will then obtain the necessary information such as simple edges and textures or even patterns and structures. This information can be obtained by finding the Red, Green, and Blue (RGB) values associated with the signature images.

The VGG-16 and ResNet-50 models in this experiment are obtained from the Keras library. These models were pre-trained using the ImageNet dataset and further trained by the signature training dataset. In addition, the layers after the last convolutional layer are removed and replaced with a Global Average Pooling (GAP) layer. This will result in 512 and 2048 embeddings, respectively. Afterwards, these embeddings will be used for the triplet loss optimization method to further enhance the embedding models. Lastly, after obtaining the embeddings from the respective signature images, a feature gallery is formed by averaging all the embeddings. Their final modified form can be seen in Figure 3 and 4.

D. Triplet Loss Optimization

To further improve the embedding model's capabilities, the proposed method takes advantage of an optimization method called Triplet Loss. This optimization process requires triplets as inputs which contain three different inputs, which are the anchor image, the positive image (genuine signature), and negative image (forged signature). The anchor image is obtained from the first instance of the genuine signature images of an identity while the postive images will take the remaining genuine signatures.

However, due to the nature of triplets, the inequality of the genuine signatures and forged signatures data must be equalized. This is done by first counting which class has the most instances and randomly repeating the signature

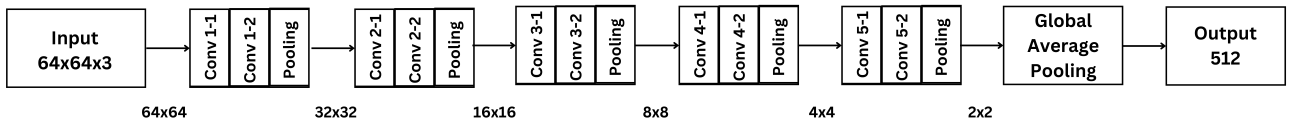

Figure 3. Feature Extractor from the VGG-16 architecture [34]

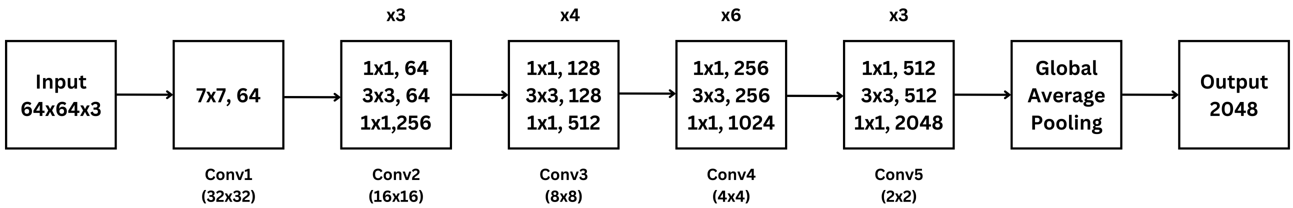

Figure 4. Feature Extractor from the ResNet-50 architecture [35]

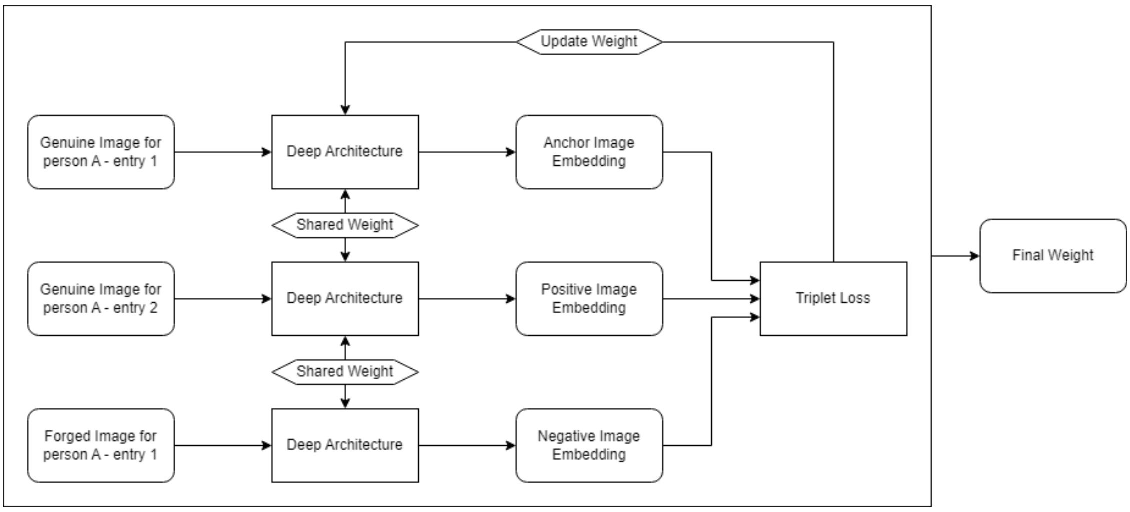

Figure 5. Triplet Loss Optimization process [36]

images of the other class until the numbers are the same while the anchor images are simply the same for each triplet. The triplet loss optimization architecture can be seen in Figure 5.

E. Classifiers for Signature Verification

In order to fully utilize the image embeddings, this research uses several machine learning classifiers. These classifiers are SVM, ANN, Random Forest, and XGBoost. Externally, the classifiers are not much different as all of them take the vector difference between the feature gallery and a signature's image embeddings as inputs. However, their internal structures are different from one another and may offer different perspectives on how the data can be used. Thus, offering various results on various occasions such as data size.

The models are trained using the training data obtained from preprocessing. Additionally, to make sure that the models are optimal, the proposed method takes advantage of hyperparameter tuning method, specifically the grid search method. This process uses the validation data instead of the training data to save computational power and time.

F. Evaluation

For the evaluation of this case, the AUC evaluation metric is used [37], which is a commonly employed performance measure in binary classification tasks. This method measures the machine learning model's discriminative ability across different decision thresholds by plotting the Receiver Operating Characteristic curve and calculating the area beneath it, which is referred to as Area Under Curve (AUC). In addition, the True Positive Rate (TPR) values at certain False Positive Rate (FPR) values such as 0.1, 0.01, and 0.001 are also evaluated.

5. EXPERIMENTAL SETUP

A. Hardware Specification

This research is conducted on a laptop equipped with an Intel(R) Core(TM) i5-8265U CPU @ 1.60GHz processor, featuring four cores and eight threads. The CPU operates at a base clock rate of 1.60GHz, with a maximum turbo boost frequency of [insert boost clock speed if available]. The system is configured with 8 GB of system memory (RAM). Additionally, the laptop is equipped with a 476 GB INTEL SS solid-state drive (SSD) for storage.

B. Software Tools

The software used to conduct the research is Visual Studio Code (Version 1.86.2) as the Primary Integrated Environment (IDE) for coding and project management. Python (Version 3.11.1) was the programming language used for implementing the entire system. Additionally, the libraries numpy, tensorflow, os, cv2, sklearn, and matplotlib for the various tasks. Furthermore, Visual Studio Code was used on a Windows 10 operating system for compatibility and ease of development.

6. RESULTS AND DISCUSSION

A. Testing Results without Triplet Loss Optimization

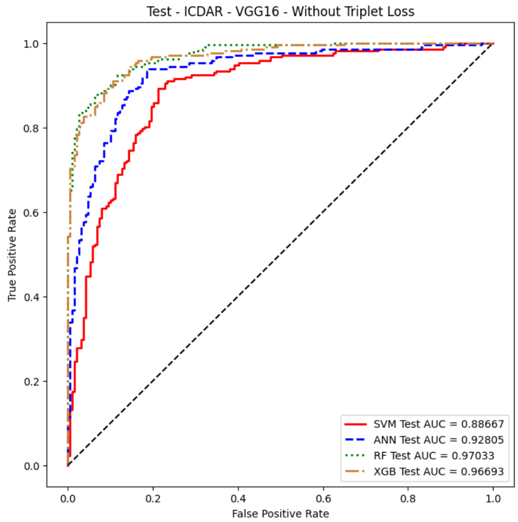

Figure 6. ROC Curve for ICDAR dataset with VGG16 and without Triplet Loss Optimization

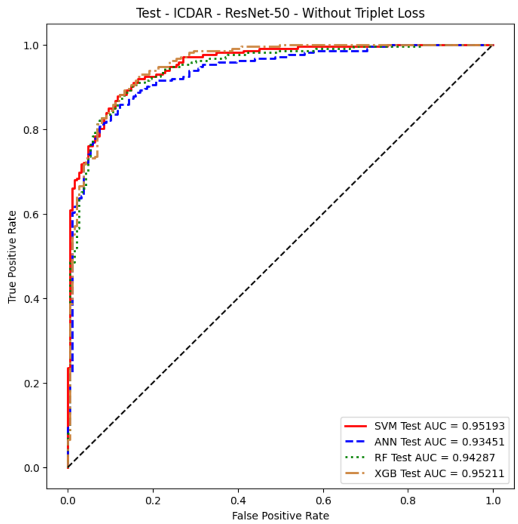

Figure 7. ROC Curve for ICDAR dataset with ResNet-50 and without Triplet Loss Optimization

From Figure 5 to Figure 8, the results of the different type of combinations of datasets, deep learning embedding models without triplet loss optimization, and machine learning classifiers are shown through ROC curves. Additionally, the numerical summary of the combinations are also shown in Table II.

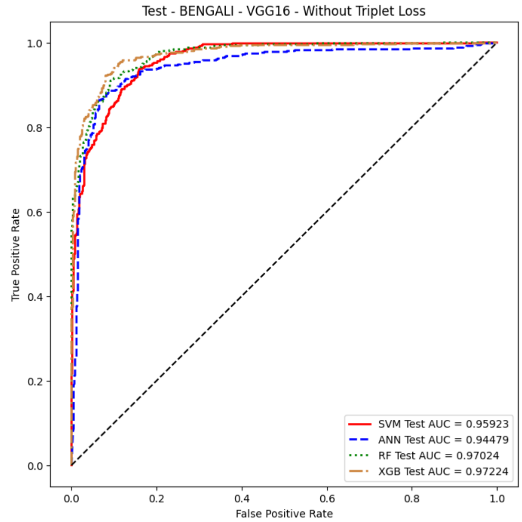

Figure 8. ROC Curve for BENGALI dataset with VGG16 and without Triplet Loss Optimization

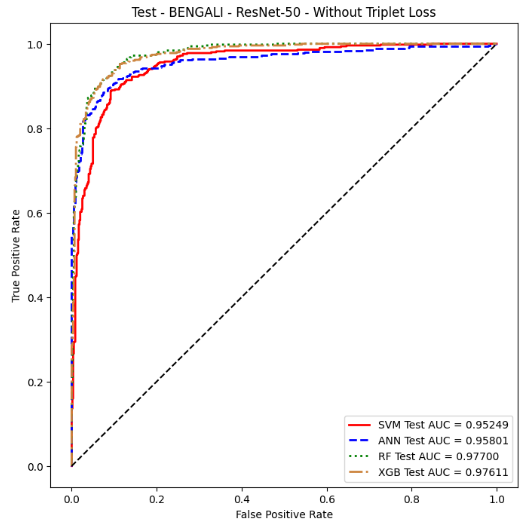

Figure 9. ROC Curve for BENGALI dataset with ResNet-50 and without Triplet Loss Optimization

It can also be seen from Figure 5 that two curves, belonging to Random Forest and XGBoost respectively, stood out as the top two curves the graph due to them being entirely above the other two curves. However, in Figure 6 to Figure 8, all curves nearly overlaps with one another making harder for visual analysis. Thus, for more information, the numerical analysis in Table II is needed.

For the combination of ICDAR dataset and VGG16 embedding model, the best machine learning classifier is Random Forest with an AUC of 0.970 and an average TPR of 0.698. On the other hand, the best machine learning classifier for the ResNet-50 embedding model is Support Vector Machine with an AUC of 0.952 and an average TPR of 0.564.

Unlike the ICDAR dataset, the combination of the BENGALI dataset and the VGG16 embedding model's best machine learning classifier is XGBoost with an AUC of 0.972 and an average TPR of 0.677. While the best

TABLE II. RESULTS TABLE WITH TRIPLET LOSS OPTIMIZATION

Without Triplet Loss Optimization				
ICDAR Dataset				
Model	AUC	TPR at FPR 0.1	TPR at FPR 0.01	TPR at FPR 0.001
VGG16 + SVM	0.887	0.623	0.142	0.024
VGG16 + ANN	0.928	0.764	0.34	0.113
VGG16 + RF	0.97	0.896	0.651	0.547
VGG16 + XGB	0.967	0.896	0.703	0.542
ResNet-50 + SVM	0.952	0.849	0.608	0.236
ResNet-50 + ANN	0.935	0.821	0.222	0.094
ResNet-50 + RF	0.943	0.835	0.486	0.08
ResNet-50 + XGB	0.952	0.844	0.387	0.066
BENGALI Dataset				
VGG16 + SVM	0.959	0.854	0.544	0.216
VGG16 + ANN	0.945	0.886	0.254	0.023
VGG16 + RF	0.97	0.916	0.65	0.558
VGG16 + XGB	0.972	0.94	0.716	0.374
ResNet-50 + SVM	0.952	0.89	0.45	0.136
ResNet-50 + ANN	0.958	0.906	0.684	0.548
ResNet-50 + RF	0.977	0.936	0.684	0.198
ResNet-50 + XGB	0.976	0.938	0.756	0.286

machine learning classifier for the ResNet-50 embedding model is Random Forest with an AUC of 0.977 and an average TPR of 0.606.

B. Testing Results with Triplet Loss Optimization

Just like in sub section A, Figure 9 to Figure 12 represents the results of the combinations of datasets, deep learning embedding models, and machine learning classifiers. However, these embedding models in the four figures are optimized through triplet loss optimization. Additionally, Table III represents the numerical summary of the combinations.

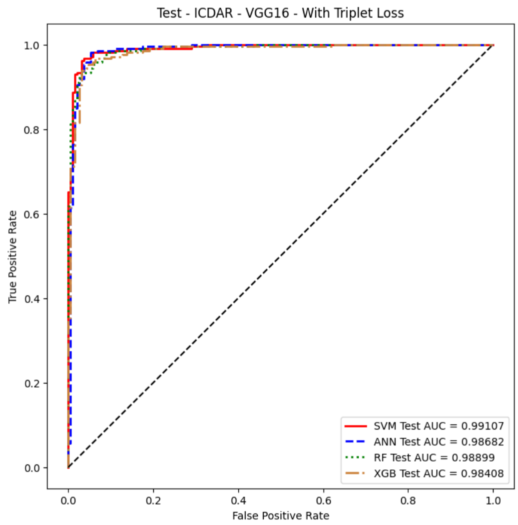

Figure 10. ROC Curve for ICDAR dataset with VGG16 and with Triplet Loss Optimization

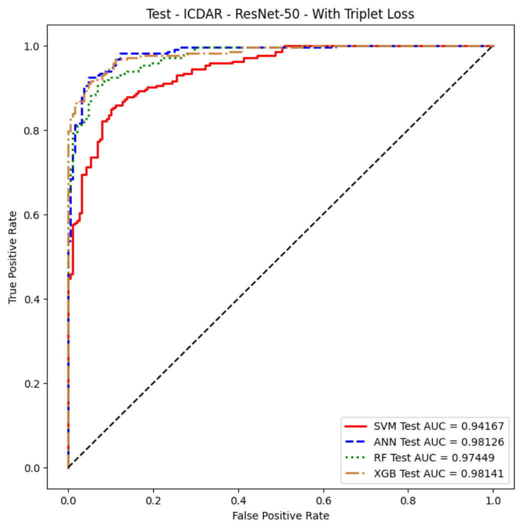

Figure 11. ROC Curve for ICDAR dataset with ResNet-50 and with Triplet Loss Optimization

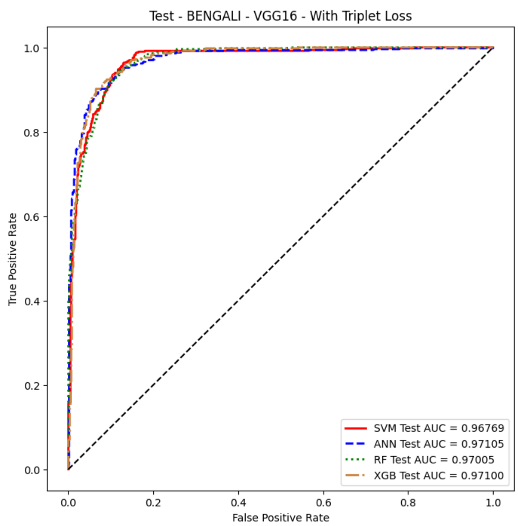

Figure 12. ROC Curve for BENGALI dataset with VGG16 and with Triplet Loss Optimization

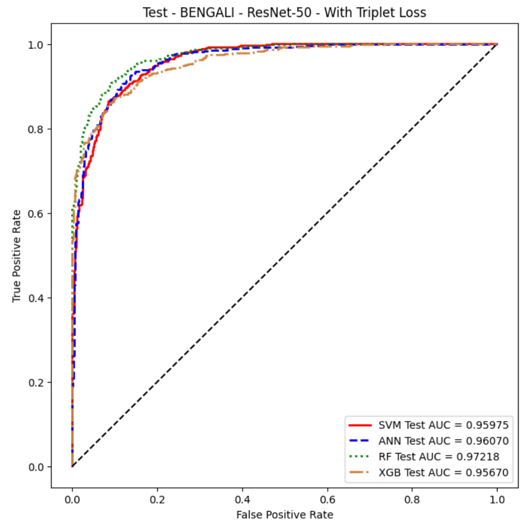

Figure 13. ROC Curve for BENGALI dataset with ResNet-50 and with Triplet Loss Optimization

In Figure 10, there are three curves, which represent the ANN, Random Forest, and XGBoost classifiers, which stood out among the four curves. However, Figure 9, Figure 11, and Figure 12 showed that the curves nearly overlap with each other. Thus, further numerical analysis using Table III is necessary.

For the ICDAR dataset and VGG16, SVM is the best classifier, achieving an AUC of 0.991 and an average TPR of 0.778. Other than that, the XGBoost classifier achieved an AUC of 0.981 and an average TPR of 0.857 for the ResNet-50 embedding model.

In contrast to the ICDAR dataset, the BENGALI and VGG16 combination's best machine learning classifier is ANN, achieving an AUC of 0.971 and an average TPR of 0.527. Lastly, the best machine learning classifier for the combination of BENGALI and ResNet-50 is Random Forest with an AUC of 0.972 and an average TPR of 0.648.

C. Summary of Testing Results

For the ICDAR dataset, the best method without triplet loss optimization is through the combination of VGG16 and Random Forest with an AUC of 0.970 and an average TPR at 0.698. On the other hand, the best method with triplet loss optimization is through the combination of VGG16 and SVM with an AUC of 0.991 and an average TPR of 0.778.

Meanwhile for the BENGALI dataset, the best method without triplet loss optimization is through the combination of ResNet-50 and Random Forest with an AUC of 0.977 and an average TPR of 0.606. In addition, the same combination still retained its spot for the best method with triplet loss optimization, having an AUC of 0.972 and an average TPR of 0.648.

It can be summarized that the triplet loss method can improve the AUC of the proposed methods. Instances such as the best BENGALI method losing around 0.005 points in AUC after triplet loss optimization does not necessarily mean that the overall model got worse. While the AUC indeed decreased, the average TPR received an increase of 0.042 which means that the overall model improved its sensitivity or consistency under different conditions. Furthermore, from the two datasets, the model with triplet loss optimization performs better or showed more improvements on the ICDAR dataset, which contains a lesser amount of data. Hence, it can also be summarized that the triplet loss optimization method can help when data are scarce.

TABLE III. RESULTS TABLE WITHOUT TRIPLET LOSS OPTIMIZATION

With Triplet Loss Optimization				
ICDAR Dataset				
Model	AUC	TPR at FPR 0.1	TPR at FPR 0.01	TPR at FPR 0.001
VGG16 + SVM	0.991	0.981	0.703	0.651
VGG16 + ANN	0.987	0.986	0.613	0.057
VGG16 + RF	0.989	0.981	0.811	0.618
VGG16 + XGB	0.984	0.967	0.712	0.354
ResNet-50 + SVM	0.941	0.835	0.458	0.448
ResNet-50 + ANN	0.981	0.939	0.684	0.538
ResNet-50 + RF	0.974	0.925	0.712	0.656
ResNet-50 + XGB	0.981	0.943	0.83	0.797
BENGALI Dataset				
VGG16 + SVM	0.968	0.928	0.518	0.156
VGG16 + ANN	0.971	0.916	0.656	0.008
VGG16 + RF	0.97	0.918	0.584	0.398
VGG16 + XGB	0.971	0.928	0.48	0.07
ResNet-50 + SVM	0.96	0.874	0.562	0.36
ResNet-50 + ANN	0.961	0.876	0.576	0.188
ResNet-50 + RF	0.972	0.91	0.682	0.614
ResNet-50 + XGB	0.957	0.872	0.698	0.534

Although the performance metric and the dataset used in this study are different than the other state of the arts methods, comparisons can roughly be made. The proposed method's best performance achieved the AUC of 0.991 and 0.977 for the ICDAR and BENGALI dataset respectively. On the other hand, the state of the arts methods produced various results as such, we will compare this paper's results with the papers that inspired this research. They achieved the accuracy of 95% and 96.33% respectively [4, 5]. Thus, it can be concluded that the proposed method can contend with other state of the arts methods.

7. Conclusion and Future Work

A. Conclusion

In this research, the integration of advanced technologies, particularly deep learning and machine learning techniques, was explored to overcome the limitations of traditional signature verification methods. The study focused on leveraging the combination between two powerful deep learning models (VGG-16 and ResNet-50) and machine learning signature verification models (SVM, ANN, Random Forest, and XGBoost), alongside the triplet loss optimization method, to enhance the accuracy of signature verification. The experiments conducted involved the application of these models on two distinct signature datasets, ICDAR-2011 and BHSig260-Bengali. The evaluation metrics, including AUC and TPR at specific FPR values, provided a comprehensive assessment of the proposed methods' performance. The findings indicated that the triplet loss optimization method contributed to better overall model performance, emphasizing its role in enhancing the discernment between genuine and forged signatures.

The results highlighted the effectiveness of combining VGG-16 and Random Forest for the ICDAR dataset and ResNet-50 with Random Forest for the Bengali dataset without the triplet loss optimization method. On the other hand, the combination of VGG-16 and SVM is better for the ICDAR dataset while ResNet-50 and Random Forest remain as the best combination for the BENGALI dataset with the triplet loss optimization method. Notably, the triplet loss method demonstrated improvements in the models' sensitivity, especially in scenarios with limited data, as observed in the ICDAR dataset. In conclusion, the combination of deep learning embedding models (VGG-16 and ResNet-50), triplet loss optimization, and machine learning classifiers (SVM, Random Forest, ANN, XGBoost) presents a versatile and robust solution for handwritten signature verification. The proposed methods exhibit promising results, opening avenues for improved security protocols in various sectors such as banking, legal institutions, and government agencies. As technology continues to advance, the integration of sophisticated signature verification systems becomes crucial for maintaining the integrity of electronic transactions and safeguarding sensitive information.

B. Future Work

In this work, the images were resized down to 64 by 64 due to the limitations of the devices used in this research. Future works may consider the possibility of larger image resizes for the possibility of better results. In addition, the proposed method already achieved high results even without triplet loss optimization to improve the embedding model, future work may consider the possibility of a noisier dataset and more extreme data sizes, be it lower or higher than the ones used in this experiment. In addition, this study's evaluation only considers the results of the proposed method, not including efficiency. Therefore, any future work continuing this research may consider improving the efficiency aspect in general.

References

[1] P. Bharti and C. Natarajan, "Deep learning-based signature verification," *IJARIIE-ISSN(O)-2395-4396*, vol. 9, no. 4, p. 720, 2023.

[2] Y. Muhtar, M. Muhammat, N. Yadikar, A. Aysa, and K. Ubul, "Fc-resnet: A multilingual handwritten signature verification model using an improved resnet with cbam," *Applied Sciences*, vol. 13, no. 14, p. 8022, Jul 2023. [Online]. Available: http://dx.doi.org/10.3390/app13148022

[3] M. Haratatik and K. Anam, "Comparison of convolutional neural network architecture on detection of helmet use by humans," vol. 8, p. 44, May 2023.

[4] M. Bajpai, "Handwritten signature verification using deep learning technique in conjunction with image processing," 2022, national College of Ireland, Jan. 2022.

[5] N. Xamxidin, Mahpirat, Z. Yao, A. Aysa, and K. Ubul, "Multilingual offline signature verification based on improved inverse discriminator network," *Information*, vol. 13, no. 6, p. 293, Jun 2022. [Online]. Available: http://dx.doi.org/10.3390/info13060293

[6] D. Engin, A. Kantarci, S. Arslan, and H. K. Ekenel, "Offline signature verification on real-world documents," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Jun 2020.

[7] J. Poddar, V. Parikh, and S. K. Bharti, "Offline signature recognition and forgery detection using deep learning," *Procedia Computer Science*, vol. 170, pp. 610–617, Apr 2020. [Online]. Available: http://dx.doi.org/10.1016/j.procs.2020.03.133

[8] J. Lu, H. Qi, X. Wu, C. Zhang, and Q. Tang, "Research on authentic signature identification method integrating dynamic and static features," *Applied Sciences*, vol. 12, no. 19, p. 9904, Oct 2022. [Online]. Available: http://dx.doi.org/10.3390/app12199904

[9] Y. Borse, A. Patil, S. Shah, and A. Gharu, "Signature verification using deep learning," *International Journal of Creative Research Thoughts (IJCRT)*, vol. 11, no. 4, pp. h956–h960, Apr 2023.

[10] O. E. Melhaoui and S. Benchaou, "An efficient signature recognition system based on gradient features and neural network classifier," *Procedia Computer Science*, vol. 198, pp. 385–390, 2022. [Online]. Available: http://dx.doi.org/10.1016/j.procs.2021.12.258

[11] J. Lopes, B. Baptista, N. Lavado, and M. Mendes, "Offline handwritten signature verification using deep neural networks," *Energies*, vol. 15, no. 20, p. 7611, Oct 2022. [Online]. Available: http://dx.doi.org/10.3390/en15207611

[12] E. Alajrami, B. A. M. Ashqar, B. S. Abu-Nasser, A. J. Khalil, M. M. Musleh, A. M. Barhoom, and S. S. Abu-Naser, "Handwritten signature verification using deep learning," *International Journal of Academic Multidisciplinary Research (IJAMR)*, vol. 3, no. 12, pp. 39–44, Dec 2019.

[13] K. Simonyan and A. Zisserman, "Very deep convolutional net-

works for large-scale image recognition," *arXiv*, 2014, doi: 10.48550/ARXIV.1409.1556.

[14] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," in *British Machine Vision Conference*, 2015.

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *arXiv*, 2015, doi: 10.48550/ARXIV.1512.03385.

[16] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," 2015, arXiv, 2015, doi: 10.48550/ARXIV.1503.03832.

[17] H. Wang, J. Xiong, Z. Yao, M. Lin, and J. Ren, "Research survey on support vector machine," Dec 2017. [Online]. Available: http://dx.doi.org/10.4108/eai.13-7-2017.2270596

[18] Y. Guerbai, Y. Chibani, and B. Hadjadji, "The effective use of the one-class svm classifier for handwritten signature verification based on writer-independent parameters," *Pattern Recognition*, vol. 48, no. 1, pp. 103–113, Jan 2015. [Online]. Available: http://dx.doi.org/10.1016/j.patcog.2014.07.016

[19] P.-Y. Hao, J.-H. Chiang, and Y.-D. Chen, "Possibilistic classification by support vector networks," *Neural Networks*, vol. 149, pp. 40–56, 2022. [Online]. Available: http://dx.doi.org/10.1016/j.neunet.2022.02.007

[20] M. Madhiarasan and M. Louzazni, "Analysis of artificial neural network: Architecture, types, and forecasting applications," *Journal of Electrical and Computer Engineering*, vol. 2022, Apr 2022. [Online]. Available: http://dx.doi.org/10.1155/2022/5416722

[21] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[22] S. Haykin, *Neural Networks and Learning Machines*. Pearson, 2009.

[23] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[24] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.

[25] G. Biau, "Analysis of a random forests model," *Journal of Machine Learning Research*, vol. 13, p. 1063, Oct 2012.

[26] L. Breiman, "Random forest," *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct 2001. [Online]. Available: http://dx.doi.org/10.1023/a:1010933404324

[27] A. S. More and D. P. Rana, "Review of random forest classification techniques to resolve data imbalance," in *2017 1st International Conference on Intelligent Systems and Information Management (ICISIM)*, 2017.

[28] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016.

[29] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *The Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232, 2001.

[30] O. L. Mangasarian and E. W. Wild, "Multiple additive regression trees with application in artificial intelligence," *Journal of Human–Robot Interaction*, vol. 2, no. 1, pp. 9–32, 2013.

[31] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, 2012.

[32] D. Sculley *et al.*, "Hidden technical debt in machine learning systems," 2015.

[33] S. Rendle, "Factorization machines," in *Proceedings of the 2012 IEEE International Conference on Data Mining*, 2012.

[34] F. Liu, Y. Wang, F.-C. Wang, Y.-Z. Zhang, and J. Lin, "Intelligent and secure content-based image retrieval for mobile users," *IEEE Access*, vol. 7, pp. 119 209–119 222, Aug 2019. [Online]. Available: http://dx.doi.org/10.1109/access.2019.2935222

[35] V. N. Vinaykumar, J. A. Babu, and J. Frnda, "Optimal guidance whale optimization algorithm and hybrid deep learning networks for land use land cover classification," *EURASIP Journal on Advances in Signal Processing*, vol. 2023, no. 1, Jan 2023. [Online]. Available: http://dx.doi.org/10.1186/s13634-023-00980-w

[36] H. Xuan, A. Stylianou, X. Liu, and R. Pless, "Hard negative examples are hard, but useful," pp. 126–142, Nov 2020. [Online]. Available: http://dx.doi.org/10.1007/978-3-030-58568-6_8

[37] A. P. Bradley, "Auc: A statistically consistent and more discriminating measure than accuracy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 3, pp. 289–293, Mar 1998. [Online]. Available: http://dx.doi.org/10.1109/34.667881

Andreas Christianto is a Computer Science master's student in Bina Nusantara University. His interest is in the field of machine learning and deep learning.

Jovito Colin is a student from Bina Nusantara University currently pursuing a Master's in Computer Science. His research interest's include deep learning, computer vision, embedded systems, and data mining.

Gede Putra Kusuma received PhD degree in Electrical and Electronic Engineering from Nanyang Technological University (NTU), Singapore, in 2013. He is currently working as a Lecturer and Head of Department of Master of Computer Science, Bina Nusantara University, Indonesia. Before joining Bina Nusantara University, he was working as a Research Scientist in I2R – A*STAR, Singapore. His research interests include computer vision, deep learning, face recognition, appearance-based object recognition, gamification of learning, and indoor positioning system.