National College of Ireland

Handwritten Signature Verification using Deep Learning Technique in Conjunction with Image Processing

MSc Research Project
Data Analytics

Manish Bajpai
Student ID: x19229887

School of Computing
National College of Ireland

Supervisor: Dr.Hicham Rifai

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Manish Bajpai
Student ID:	x19229887
Programme:	Data Analytics
Year:	2021
Module:	MSc Research Project
Supervisor:	Dr.Hicham Rifai
Submission Due Date:	31/01/2022
Project Title:	Handwritten Signature Verification using Deep Learning Technique in Conjunction with Image Processing
Word Count:	8568
Page Count:	23

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Manish Bajpai
Date:	30th January 2022

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Handwritten Signature Verification using Deep Learning Technique in Conjunction with Image Processing

Manish Bajpai
x19229887

Abstract

Handwritten signatures have been used in financial and legal documents for a long time. It is crucial in the financial sector to distinguish between authentic handwritten signatures and forgeries. As the naked eye cannot always distinguish between real and forged signatures, the majority of handwritten signatures are accepted in banks by human intervention, which can lead to human errors or fraud. A deep neural network is used to recognize genuine handwritten signatures. In this study the VGGNet neural network is employed to recognize handwritten signatures. The case study's evaluation metrics include accuracy, precision, and recall, which were compared to the other state of the arts in the field of handwritten signatures. The findings of this case study are designed to aid the banking industry in avoiding financial sector forgeries. The VGGNet model outperforms other state of the art models in the domain of handwritten signature datasets.
Keywords— Deep Neural Network,VGGNet,Image Processing and CNN

1 Introduction

Handwritten signature identification has long been a problem in the banking and legal industries. There has been a lot of counterfeiting in the financial realm in the past and still to this day. The problem is that in most banking institutions, handwritten signature identification is still done manually, which leads to human error. Also, the human eye can't always tell the difference between forged and real original signatures. There are two types of handwritten signature authentication one is online and the other is offline and for this case study, this research has taken into account offline signature authentication. Offline handwritten signatures are typically difficult to recognize owing to a variety of factors, such as the picture of the handwritten signature being faded. The challenges in obtaining good accuracy for forged handwritten signatures have been significant, therefore in (Yapici et al.; 2019) study, Convolution Neural Network (CNN) was used and obtained higher accuracy over other state of the art approaches in the handwritten signature domain at the time.

Selecting the correct feature has long been a problem in the identification of handwritten signatures, and it plays a big part in detecting forgeries. (Batool et al.; 2020) used a Gray level matrix to extract features in his research. Principal component analysis (PCA) was used to extract the best feature to match the model, which did not include all of the features. The challenge of extracting local features from handwritten signatures had been raised and addressed in a number of studies on handwritten signatures. The same problem was covered in (Coetzer et al.; 2004) work. For the detection of handwritten signatures, he used the Hidden Markov method. In the pre-processing step, Discrete Fourier transform was applied to extract the local and global features, which is crucial in the domain of handwritten signatures. This study is an expansion of the other scholar's work which has been discussed in the preceding sections, and it adds novelty by integrating the VGGNet neural network with the convolutional neural network. This research case study addresses the issue of feature extraction which has been discussed by various researchers and therefore, it raises a research question.**To what extent can a VGGNet network be used to detect counterfeit signatures?**

The following are the goals that this case study tries to achieve.

1. Building a VGGNet neural network to boost performance of the evaluation metrics in the handwritten signature domain.

2. Using assessment measures to compare to other researchers studies in the domain of handwritten signatures.

The main significance of this case study is the use of VGGNet neural network to detect the handwritten signature and to achieve better performance as compared to other state of Art. As a result, this case study can assist in identifying fraud in handwritten signatures in banking and legal documents. This research has been divided into eight sections, wherein section 2 gives deep insight into the academic papers that have been researched in the handwritten signature domain. The third section focuses on the research methods used for this case study. The fourth section follows, emphasizing the design specification aspect. Sections 5 and 6 illustrated the execution of the proposed technique and the evaluation of the results, respectively. Sections 7 and 8 explore the findings and future implications of this research case study.

2 A comprehensive analysis of related work.

2.1 An Introduction to literature review.

Many studies have been conducted in the field of handwritten signature identification. The field of handwritten signatures is dominated by deep neural network applications. In previous studies, several applications of neural networks such as ResNet and DenseNet were utilized to identify handwritten signatures. Some academic papers in the domain of handwritten signatures that have been thoroughly critically assessed are discussed below. Sections from 2.2 to 2.6 shows the case study and applications of deep neural network.

2.2 Comprehensive case study of Deep Neural Network

Detecting text images in machine learning algorithms has always been a difficult issue, especially in the case of handwritten signatures. Along with the deep neural network, feature extraction and feature engineering have always played a critical role in the identification of authentic handwritten signatures, as illustrated in the study article by (Javidi and Jampour; 2020). For the identification of genuine signatures, Javidi and the team used the Resnet architecture. The use of thickness to detect the authentic characteristic is a unique feature in the area of handwritten signature utilization, and it is the most fascinating element of the study article. The study can detect signals that haven't been trained by the model, which is a first in this case study.

For the identification of handwritten signatures, Deep Neural Network have been frequently employed and for forged signature detection, (He and Schomaker; 2020) and his team used a two-component FragNet network in this study. The feature pyramid was used to extract feature maps from handwritten signatures, while the FragNet route was utilized to train the model and predict legitimate signatures. Image segmentation, which is almost unattainable in the case of cursive signatures, was one of the study paper's shortcomings. The most significant aspect of this study was that it outperformed previous studies on neural architecture that had only been trained on text images.

The Convolutional Neural Network (CNN) has been widely utilized for image categorization, particularly in the identification of genuine handwritten signatures. (Tang and Wu; 2016) paper combined traditional CNN with joint Bayesian methods. The research was unique and in that it used data argumentation in the pre-processing section because there was less data available to train the model to recognize genuine signatures. Tang and his team comes with the new unique way to extract feature maps from handwritten signatures in place of local patches which were being used at that time in other state of art used in detecting signatures. This research also used Bayesian model which was very important to detect writer identification in place of cosine or chi-square distance metrics.

CNN has been used to detect signature classification in the domain of handwritten signature classification and the article by (Xing and Qiao; 2016) employed CNN to detect signature classification, in addition as there was a scarcity of data, the researchers used data augmentation. Patch scanning was also utilized to deal with the text pictures of various lengths, which is a novel approach. The researchers have done something new here by combining Chinese and English linguistic signatures to improve performance. The use of only four-letter English was one of the study's significant drawbacks.

The importance of features in picture categorization is demonstrated by (Chahi et al.; 2018) in article. Block Wise Local Binary Count (BW-LBC) was used by researchers to identify genuine handwritten signatures. The basic goal of (BW-LBC) is to distinguish the style of handwritten signatures by grouping signature components and detecting signatures using histograms. The lack of data for training the model is a major drawback of this study. N closest neighbor is utilized for signature categorization, and prediction is based on hamming distance, which was a better option than Euclidean distance.

The use of transformation in pre-processing for image classification has been widely employed, and (Quraishi et al.; 2013) study focused on the transformation of handwritten signature images. For picture alteration, Quraishi and his colleagues employed spatial and frequency domain methods. First, a specific portion of the image is extracted, and then the Ripplet transformation and the polar transformation are applied. A backward propagation neural network was utilized, which resulted in excellent accuracy. Other approaches for identifying signatures used in this case study included height, width ratio, and scale invariant area.

Various Deep learning Techniques had been used for the classification of images. (Ghosh; 2021)utilized the Recurrent neural network for detection of authentic signatures. Different features were extracted locally from handwritten signatures and then feature map is generated which was then used in two types of RNN models. The first one is Long short-term memory and the second one is Bidirectional long-short term memory.

RNN(Recurrent Neural Network) has outperformed the other state of art which involve CNN as a base model.

Handwritten signature verification has long been a source of concern since identifying a signature can be challenging due to the numerous styles and other minute characteristics contained within it. (Rodrigues et al.; 2016) collected seven traits from feature maps to compare real and fake signatures on paper. Height to width ratio, Density Ratio, and Centre Gravity were the most relevant variables recovered and the tiny window size is a key weakness of this study.

In the identification of genuine signatures, deep neural networks have been frequently employed. To make the most of both networks' benefits, (Jampour et al.; 2021) combined ResNet and CapsNet neural architecture. The team has utilized the advantages of CapsNet architecture, which is known for recognizing components in detail and their placements, which is critical in handwritten signature validation. The ResNet architecture, on the other hand, was employed for feature extraction, which is an important part of signature authentication. The study's biggest limitation was that it required a huge number of images of handwritten signatures for the training set in order to achieve higher accuracy. It was also discovered that the system's accuracy had not altered over the intermediate epochs, implying that the model was over fitted.

2.3 Comprehensive case study of Geometrical feature in handwritten signature verification

The recognition of handwritten signatures is aided by a geometric characteristic. (Arora et al.; 2014) research concentrated on the extraction of geometric characteristics such as the Quad Surface feature, Area Ratio, and Distance Ratio and the Gaussian Mixture Model(GMM) is used to train the model. The Euclidean distance is a handy method for determining if a handwritten signature is authentic or not. The pre-processing step had been precisely exploited to achieve better accuracy in which the image dataset has been transformed from RGB to binary image, allowing significant features to be retrieved and, most importantly, noise from the image to be removed, which is critical for image detection.

The recognition of handwritten signatures relies heavily on feature extraction. The discrete Fourier transform was used in (Nathwani; 2020) article to extract characteristics that might help in the identification of legitimate signatures. The collected characteristics are then sent into an artificial neural network, which is a hybrid of a Gated Recurrent Unit and a Long-Short term Memory(LSTM) and in comparison to several previous studies, it has attained superior accuracy. Padding was not used in this study since the length and height of each signature varies.

Pre-processing, feature extraction, feature selection, and feature verification are the four basic processes in handwritten signature authentication, according to the work by (Sharif et al.; 2020). Area of signature, pure width, pure height, and normalized height were the global features retrieved. The slope angle and distance are two local characteristics. The evolutionary algorithm is used to pick characteristics, and then a Support Vector Machine(SVM) is used to verify the selected features in order to identify valid handwritten signatures. The use of the median filter to eliminate noise was unique in this study, as was the use of segmentation with the assistance of the Otsu's method to accomplish grey scaling.

In order to identify legitimate signatures, you must first understand the differences between handwritten and typed signatures. These variances in handwritten signatures were used in (Zheng et al.; 2021) study to identify signatures and were referred to as micro deformations. In their research, Zheng and his colleagues employed position coordinates to determine micro deformations, which they were able to do using a Convolutional neural network (CNN). Max pooling is employed for this; however, location coordination should not be overlooked during pooling operations because it is a critical component for handwritten signature recognition. The investigation was limited to a single data set and yielded better results, but this can also be a restriction because we cannot trust the reliability of the system architecture presented for handwritten signature authentication identification.

2.4 Comprehensive case of Siamese Neural Network

Siamese neural networks have been used in handwritten signatures because of their advantages, such as the twin network for both forge and authenticity. A Siamese neural network was utilized in (Ruiz et al.; 2020) research to accomplish this. The use of on-demand random generators to provide variance in training the model is a key innovation of the study, and we also have enough data to train the model. Three datasets were utilized to enhance the model's performance, and it outperformed other neural network models. However, there were certain flaws, such as the absence of triplet and the use of only one reference signature per writer in training the model in this study.

The use of several neural networks to create a hybrid model for handwritten signature authenticity identification and for the recognition of handwritten signatures, (Jagtap et al.; 2020) research used a Siamese neural network and a Convolutional neural network as a foundation model. The loss function, which is a critical in every deep neural network construction, has been considered for this research. The assessment metrics of False Acceptance rate and False Rejection rate were well-used, and it outperformed other state of the art handwritten signature authentication methods.

Deep neural networks, such as Convolutional Neural Networks(CNN), were used to authenticate handwritten signatures and (Rateria and Agarwal; 2018) study combined traditional CNN with a Siamese neural network. For the detection of handwritten signatures, two configurations were employed. The first configuration is a feature extractor, which is important for detecting whether or not a signature is genuine, and the second configuration is a classifier, which employs a Siamese neural network. For the recognition of handwritten signatures, a Siamese neural network consists of twin identical networks. It is the first time that two networks have been used to extract features to identify authentic and forge signatures.

The Siamese neural network has been used to detect handwritten signatures for a long time because of its benefit of having two distinct sub neural networks carrying the same weights in the convolutional layer and having it trained in learning feature vector Research by (Dey et al.; 2017). The Euclidean distance is used to determine the similarity score, and the goal of the Siamese neural network is to reduce the distance when the pair is similar, i.e. when two handwritten signatures are similar in terms of feature, and to raise the distance when the pair is dissimilar. The use of Euclidean distance is ideal for this study since it makes calculating and determining similarity scores was easy.

2.5 Comprehensive case study of Resnet neural network

For the identification of handwritten signatures, the ResNet architecture was frequently utilized. Digital signal processing(DSP) was employed in the pre-processing of (Ishikawa et al.; 2020) article. ResNet architecture was used to solve the limitations of Convolutional Neural Networks(CNN), one of which being the vanishing gradient problem, which was gently fixed by ResNet architecture. The skip function, also known as the unity function, is used in ResNet design to alleviate the problem of disappearing gradients by directly coupling output and input layers. As a consequence of human mistake in the reference signature, there were some discrepancies in the results.

Identifying handwritten signatures by extracting elements that are important in their identification, (Hu and Chen; 2013) research used three pseudo dynamo features to extract the best feature from a handwritten signature, all of which were obtained during the pre-processing phase. The first dynamic feature was grey scaling, which means we removed RGB colors from the image's background in handwritten signature authentication. The second dynamic feature was local binary pattern, and the third was Histogram of Oriented Gradients(HOG). In this study, two classifiers were employed: the first was a Support vector machine, which was used to train extract features from a reference signature, and the second was an AdaBoost classifier, which was used to train authentic and random handwritten signatures.

2.6 Comprehensive case study of VGGNet neural network

(Foroozandeh et al.; 2020) has made substantial use of pre-trained models for recognizing handwritten signatures. Foroozandeh and his team fine-tuned CNN to work as a feature extractor and VGG16 to work as a classifier in this research study. The curve Angle was used as a geometric feature in the handwritten signature image, which is unique. In this study, the performance of VGGNet was compared to that of a Siamese neural network, and VGGNet outperformed the Siamese neural network in terms of assessment criteria.

(Bhat et al.; 2) has used a deep neural network for feature extraction. The VGG16 network, which contains 13 convolutional layers and three dense layers, was employed as a feature extractor in this research case study to extract features from a handwritten signature image. The use of VGGNet as a feature extractor is novel in this case study and SVM was employed as a classifier in this case study. However, there was no hyper parameter tuning, which is crucial for handwritten signature authentication. For image detection, a neural network was deployed. For the detection of handwritten signature images in (Junior et al.; 2020) research VGGNet was used. The study is unusual as in that it captures multiple geometric Features for handwritten signature image recognition, such as ink, pen type, and ink hue. However, by using the CV2 library to create feature vectors, the pre-processing might have been done more efficiently.

In the Related work that was discussed above in reference to Handwritten signature authentication, deep neural networks, feature extraction techniques, and their enlarged application to the best use were utilized. After analysing the state of the art in the literature study, it is clear that there is still a lot of room for development in the field of handwritten signature authentication. This research will be based on the VGGNet design, which uses many convolutional layer to extract useful features from handwritten signature image and to achieve better performance as compared to other studies.

3 Methodology

Figure 1 depicts the flow diagram which have been used for this case study. The figure 1 depicts the methods and methodology used in my handwritten signature authentication case study. For my case study, I will be utilizing Knowledge discovery Databases(KDD) as a Data analytics Approach. KDD has been also used by many researchers in the field of handwritten signature authentication.

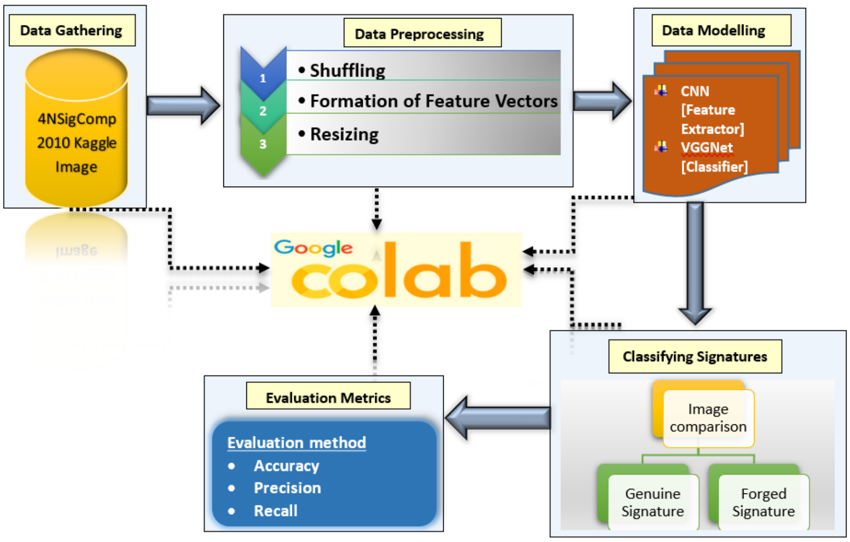

Figure 1: Research Methodology adopted for handwritten signature authentication

3.1 Data Gathering

[1]For the handwritten signature authentication, data was gathered from the Kaggle website.The signs are in Dutch and contain both faked and actual handwritten signatures. The Model will be trained on 148 classes, with 42 classes set aside for testing.Its public source dataset for handwritten signature authentication stems from the Kaggle website and complies with all General Data Protection Regulation(GDPR).

3.2 Data-Pre-processing

When working with deep neural networks, data pre-processing is particularly important. All of the pre-processing techniques that have been used in this case study are detailed below.

- **Resizing:** Resizing is necessary because, in VGGNet neural architecture, all images of identical size must be supplied to ensure consistency. The advantages of having equal-sized images to feed into the VGG16 neural network model are numerous.One of the advantages of using a fixed image size is that it will shrink less since the size will be entered in the model's correct dimensions. Less shrinkage means less deformation in the image, and all features will be captured, which is critical for handwritten signature detection.

[1]https://www.kaggle.com/robinreni/signature-verification-dataset

- **Shuffling:**Shuffling has been used as a pre-processing step, which is important in deep neural network architecture. (Horn et al.; 2017) had employed shuffling in his research on handwritten signature authenticity. The following are some of the benefits of shuffling in a neural network.

  1. Training of the Model is faster and fewer epochs are required to train the model.
  2. It also reduces biasing in the model implemented.
  3. The main advantage of shuffling the images is that it helps in reducing the contrastive loss.

- **Formation of feature vectors:** The Opencv2 library (Open Source Computer Vision Library) was utilized to create feature vectors, which was accomplished by transforming an image into a NumPy array. The following OpenCV2 library functions were used in this case study and are detailed below.

  1. **imread:** The Opencv2 package's imread function loads the picture of handwritten signatures in BGR (Blue-Green-Red) format.
  2. **Cvtcolor:** Once the image has been loaded, the color of the image is changed according to the available schemes. BGR2GRAY was used in this case study.

3.3 Processing of the image and Feature extraction

In the model-building process, feature extraction is crucial. Convolutional neural networks were utilized mostly for feature extraction. Convolutional layers and pooling layers have been used as feature extractor in this case study and in feature extraction, the convolution layer acts as a filter. The main function of convolution layers is to extract significant information from handwritten signatures, and it also helps in reducing the size of the input image with the aid of filter. The filter employed in Convolution layers slid over the input handwritten images, and the weight of the filter is multiplied by the input image to generate a feature map. The feature map of handwritten signatures contains crucial local and global features that can be exploited by the input network for the pooling layer. (Iranmanesh et al.; 2014) utilizes the CNN for feature extraction in his case study of image classification. Fig 2. Shows the pooling layer and convolution layer acts as feature extraction network and act as input for classifier network.

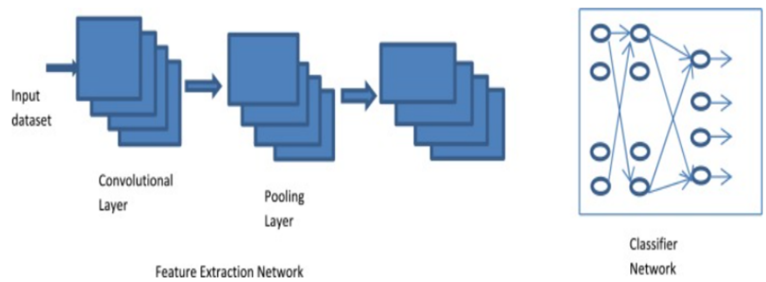

Figure 2: Feature Extraction along with classifer network

In feature extraction this case study used a pooling layer,for which the primary role is to minimize the dimension of the feature map obtained from the Convolution layer. Pooling layer, as the name implies, will aid in the reduction of learning parameters. Another purpose of the pooling layer is that it summarizes the features that are present in a certain region of the feature map. It chooses the best feature to feed into the classifier network, which in my case is a VGGNet neural network. In a deep neural network, there are different forms of pooling available; however, in my case study, I used max pooling since, for classification of a handwritten signature, we want the most potential feature from the feature map to be available for the VGGNet neural network's input(Classifier network).

3.4 Modelling

In order to authenticate handwritten signatures, two neural network models were used. The first is a Convolutional Neural Network (CNN), which serves as a base model and feature extractor, and the second is a VGGNet Network, which serves as a classifier for handwritten signatures images. CNN has employed the convolutional layer and pooling layer to minimize the size of the image and provide the optimum feature for handwritten signature detection in the CNN network. The feature retrieved with the help of a convolutional layer serves as the input image for a VGGNet. VGG16, which consists of 13 convolutional layers and three fully connected layers or Dense layers, is used in this case study to recognize handwritten signature images. Because it is a pre-trained model, the accuracy it achieves is always higher and in order to compare two neural networks, several experiments were conducted. Various hyper parameter optimization strategies were utilized to obtain the perfect parameters for both the model and the simulation.

3.5 Rationale for opting VGGNet for Handwritten Signature Authentication.

VGGNet architecture was opted for this research case study for the reason mentioned below.

1. Since the VGGNet is a pre-trained model that has already been trained on a larger dataset, we may use the adjusted weights and overall architecture in handwritten signature authentication. In neural networks, this is referred to as transfer learning. As a result, the model's performance in image recognition is always better, which is the case in this study.

2. As artificial neural networks require a large amount of computer power, they were not explored for this case study. Also, feature extraction is vital in identifying handwritten signatures, and the best feature has been reached by Deep Neural network by increasing the number of convolution layers.

3. Alexnet was not considered for this case study since the model does not have enough layers, allowing it to extract all features, resulting in poor model performance, but VGGNet can have more number of layers and aid best feature extraction with better performance of the model.

4. Some researchers have employed ResNet, as described in the Critical Assessment of the research paper, however, for this case study it's not been due to the rise

in complexity also it does not perform well when we have less data available for training

3.6 Evaluation Metrics

For this case study, three evaluation metrics were employed to assess the model's performance and will aid in answering the research question which is **"To what extent can a VGGNet network be used to detect counterfeit signatures?**. All the three evaluation metrics used are explained below along with the reasons for opting them.

1. **Accuracy:** As this case study has balance class labels, the initial evaluation metric employed in this case study is accuracy. In this evaluation metric true positive will refer to authentic handwritten signatures, and true negative will refer to counterfeit signatures ,as it is crucial to distinguish between authentic and forged signatures. In the below equation 1, TP stands for true positive denoting genuine handwritten signature image and TN stands for True negative representing forge signatures. False positive here would mean the handwritten signatures image are forge but the model predicted it as genuine and False negative are the forge signatures which are considered genuine by the model.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

2. **Recall:** It's also used to calculate the proportion of positive forecasts among all positive predictions. Recall is useful when the cost of false positives is high, which means the forge signatures were mistaken for genuine signatures by the model. This should not be the case because the model should always be able to distinguish between fake and authentic handwritten signatures.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2}$$

3. **Precision:** Precision, in a broader sense, determines the total amount of correct predictions achieved. It takes into account not only true positives, which are authentic handwritten signature images, but also false positives, which are predictions produced by the model that believe forged handwritten signature images to be genuine. The mathematical equation for precision is shown in Equation 3.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{3}$$

4 Design Specification

For handwritten signature authentication, this case study employs two networks: a Convolutional neural network that serves as a base network and a VGGNet neural network that serves as a classifier network which will serve as the main model for the comparison of handwritten signatures.

4.1 Convolution Neural Network (CNN)

When it comes to detecting forgeries in handwritten signatures, extraction is crucial. Convolution neural network extracts local and global characteristics that can be used to identify handwritten signatures. For feature extraction, the convolution layer used a filter and this filter aids in the reduction of the size of the handwritten signature input image while keeping intact the critical features required for handwritten signature authenticity recognition. Fig 3. Shows the max pooling with 2 strides which aid in feature extraction.

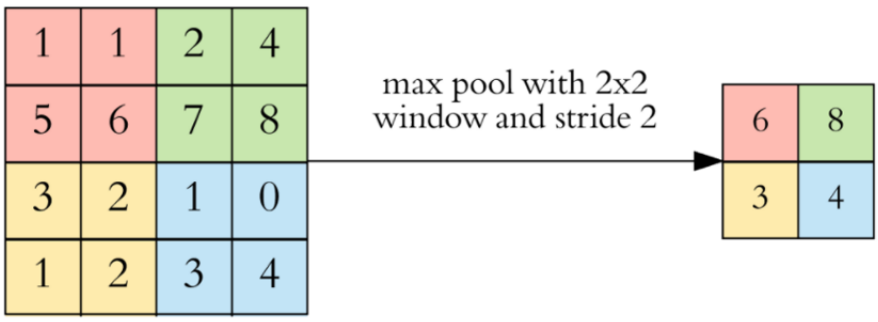

Figure 3: Max Pooling in Convolution network

The technique of pooling has also been utilized to reduce further size of the feature map obtained from convolutional layer as it helps in reducing the learning parameter. In this case study CNN network is used as a base model for comparison with VGG16 neural Network.

4.2 Visual Geometry Group Network (VGGNet)

In this case study, the VGG16 neural network is employed as a classifier network for handwritten signature identification. Because three Relu activation layers are utilized in VGGNet, classes of images are now linearly separable, which is the neural network's ultimate goal and this problem was solved using a VGGNet neural network. The main advantage of this neural network is that its windows are smaller, and VGGNet is a highly deep layer design that results in various CNN layers, resulting in enhanced accuracy. VGGNet decreases training time by using the Relu function as an activation in hidden layers, as previously indicated. For the purposes of this case study, the keras library module is utilized in this case study to deploy a pre-trained convolutional network. As previously stated, the pre-trained model was built using initialization weights. The high-level architecture diagram of VGGnet used in this case study is shown below in figure 4. Each convolutional layer in the VGGNNet design has a filter size that is clearly increasing. Research by (Fakhiroh et al.; 2021) utilized VGGNet for mobile based offline signature authentication.

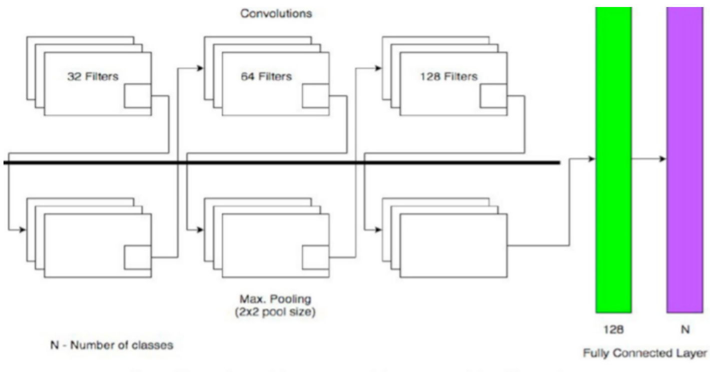

Figure 4: VGGNet neural network architecture

5 Implementation

This section outlines the VGGNet was used to develop a model for recognizing handwritten signature authenticity, including step-by-step model implementation and model training, as well as an explanation of the data processing approach utilized in this case study. Figure 5 shows the implementation of VGG16 model.

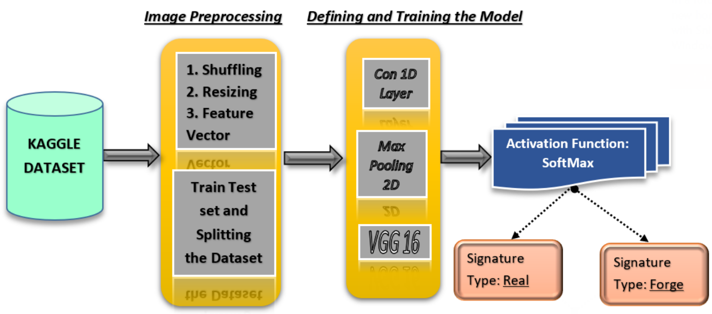

Figure 5: Block Diagram of VGG16 Model Implementation

5.1 Materials Employed

This study uses Google Colab to create a VGGNet neural network. The reason for adopting Google Colab is that we can make use of the general processing unit(GPU), which is faster than a standard CPU due to its capacity to execute parallel computation. For the implementation of the code and the model for handwritten signature authentication, a Jupyter notebook is used. The configuration manual mentions that all of the necessary libraries have been installed to build the neural network. Kaggle API was used to retrieve data directly from Kaggle website, it saves lot of computational power and storage. Using Google Drive, the data is directly retrieved from Kaggle. Using Google Drive and a kaggle.json file , the images for handwritten signature authentication were collected. Various library has been utilized in the handwritten signature image authentication which has been discussed below.

1. **Tensorflow:** In this case study, Tensorflow was used as a callable, and there are several more sub objects that are included in Tensorflow and in addition it is an open source library.

2. **keras:** The Keras library was also used, which includes other sub-modules such as optimizers and applications that can be imported from Keras.

3. **scikit-learn:** Scikit learn has also been utilized for the calling back evaluation metrics such as accuracy and confusion matrix.

5.2 Data pre-processing

In order to recognize handwritten signatures, data pre-processing is essential. Images of forging and genuine signatures can be found in two separate directories. The initial step was to resize the handwritten signature images; one of the main benefits of having equal sized images is that they will shrink less and that is the reason owing to less deformation. Less deformation indicates that the image will retain all of the spatial information that can be utilized to identify a handwritten signature.The resizing is done with the help of resizing module present in Cv2 library. The input handwritten signature images are shuffled next in the pre-processing phase, which helps to generate randomness and remove bias. It also decreases training loss while the model is learning and ensures that it takes fewer epochs, implying that it speeds up learning by modifying layer weights. Shuffling is done with the aid of Shuffle module imported of scikit learn. In the training folder, there were genuine and forge handwritten signatures, as shown in Fig 6. The goal of the OpenCv2 library was to create feature vectors by translating handwritten signatures of images into an array. The function of feature extractor has been completed by CNN, and handwritten signature recognition has been done with the help of the VGGNet neural network.

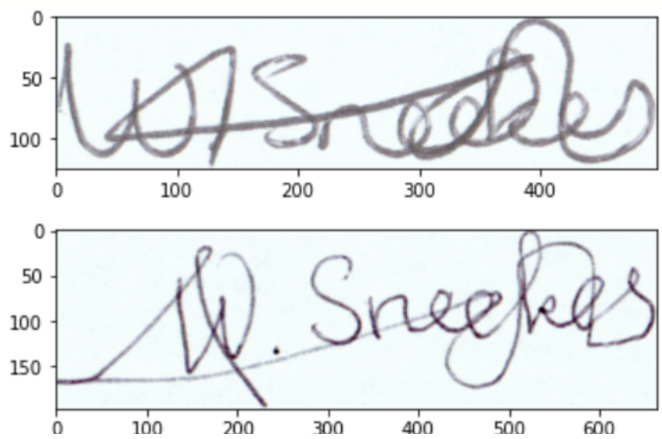

Figure 6: Genuine and Forge signature

5.3 Implementation of VGGNet Model:

Convolutional layers, sequential layers, lambda layers, and dense layers make up the framework for handwritten signature recognition. VGGNet, in a broader sense, are a combination of Convolutional networks and distinct layers. Below is a detailed description of the model construction process.

1. **Convolutional layers:** The convolutional layer of a VGGNet neural network aids in feature extraction and reduces the size of the input handwritten signature images.

Convolutional layers employed a filter known as a kernel to minimize the size of the input images of handwritten signatures by extracting key features. Feature vectors are the output of the convolutional layers. Conv2D layers were employed in this case study which Conv2d can move in two directions at the same time. Figure 7 shows two input images with (224,224,3) representing the image's height, breadth, and depth. In convolution layers, filter is used and where the 64 filters of size 3*3 is utilized in the first two layers, and it keeps increasing in following layers. In convolutional layers, element wise operations have been carried out with the use of a kernel filter, which aids in the restoration of input signatures and to add layers, this case study utilizes the Add function. The output of the convolution layer is flattened in order to transform the dense layer's output to a one-dimensional array. In this case study 13 convolution layer are used and 3 fully connected layer are used. The first two convolutional layers utilized a 64-bit filter, the second and third layers used a 128-bit filter, and the fourth and fifth layers used a 256-bit filter. The following convolutional layer, which is the 7th, 8th, and 9th, utilized a filter size of 512, as did the next convolutional layer. Fig 7 shows the layers in VGG16 model in this case study.

Layer (type)	Output Shape	Param #
input_7 (InputLayer)	[(None, 224, 224, 3)]	0
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36928
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73856
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147584
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295168
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590080
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590080
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1180160
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2359808
block4_conv3 (Conv2D)	(None, 28, 28, 512)	2359808
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2359808
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0
sequential_6 (Sequential)	(None, 2)	6423298

Figure 7: VGG16 network layers

2. **Activation layers:** The Activation layer's main purpose is to introduce non-linearity in the output of neural network neurons. The activation function Rectified Linear Unit(Relu) is employed in this case study. When the input is negative, this function returns zero, and when the input is positive, it returns the same number. The relu function is used here with the goal of making image data modified linearly separable. It actually aids in the resolution of the vanishing gradient problem that occurs in other activation functions. Below equation describes the equation of activation function. Equation 4 shows the function for Relu activation function.

$$f(x) = max(0, x) \tag{4}$$

3. **Pooling layers:** The pooling layers were used to minimize the image's size while keeping entact the most critical features of handwritten signature photos. Another major goal of the pooling layer is to minimize the network's learning parameter while retaining the majority of its properties. The pooling layers also aid in preventing the model from being overfit. The convolutional layer generates a feature map and keeps the minute details of the images, such as coordinates, but if these coordinates change as a result of operations such as resizing, shuffling, and rotation of a handwritten signature image, a new feature map is generated, so the Pooling layer ensures that the new feature map does not contain the minute details that are present in the output of the convolutional layer. In this case study, maximum pooling was used to extract the maximum value from each patch in the feature map. In the case study of handwritten signature authentication, four max-pooling layers have been used.

4. **Dense layer (Fully connected layer):** A dense layer that is connected to all preceding levels is referred to as a fully connected layer. The weight of the layers, as well as the bias, are key parameters in the activation function of dense layers. In this case study, the Softmax and Relu activation function are applied. In this case study, the Softmax activation function was implemented and the reason being that when there are more than one classes to predict, the Softmax function is employed in the output layer, which is appropriate for this research because there are more class labels in this case study. The Dense layer's input will always be in the form of (batch size,input dim), and its output will always be in the format of (batch size, units). Equation 5 shows that mathematical form for the output for dense layers and Equation 6 shows the expression for Softmax activation function. Total of 3 fully connected layers has been utilized in this case study.

$$Output = activation(dot(input, kernel) + bias) \tag{5}$$

$$\text{Softmax}(x_i) = \frac{\exp(x_i)}{\sum_j \exp(x_j)} \tag{6}$$

5.4 Training of the implemented VGG16 Model:

There are 28,954 images in the dataset and out of the 28,954 images, 23206 were utilized to train the model and 5748 were used to test the VGGNet model. So, in terms of percentages, we'll use 80 percent of the images to train the model and 20 percent of the images to test the model. The hyper parameters used to optimize the model are listed below.

1. **Adam Optimizers:** In deep neural networks, it is utilized as an alternative to stochastic gradient. Actually, the noise in the patches of handwritten signature images is solved using a combination of ADAgrad and RMSProp. Back propagation has been utilized to make the model learn and alter its weight using learning parameters (alpha).

2. **Binary Cross entropy:** It's employed when a loss function between true and predicted classes needs to be generated, which is the situation in our study because we're using VGGNet neural network with more than 2 class labels. The purpose

of cross entropy loss is to compare the output probabilities to the original values. Equation 7 shows the mathematical function cross entropy loss.

$$\text{Loss} = \sum_{j} (t_i) log(p_i) \tag{7}$$

3. **Callbacks:** Callbacks are employed effectively in this case study, and the key reason for this is that it smoothens the learning curve and trains a VGG16 Model considerably faster. It also aids in the model's avoidance of overfitting. The following are some of the reasons why callbacks are used in neural networks.

➤ **Earlystopping:** It is a method of mentioning a huge number of Epochs in the neural network and stop training the model when there is no increase in the model's performance. In this study, an early stop was placed on the parameter validation loss, which indicates that if the validation loss does not improve in continuous epochs, the training of the VGG16 model will be halted.

➤ **ReduceLROnPlateau:** It helps in reducing the learning rate when the accuracy is not increasing of the VGGNet neural network.

➤ It determines statistics of the model execution plan.

6 Results and Evaluation

Three experiments were carried out in this case study of handwritten signature authentication to improve the accuracy of the VGGNet neural network . Tuning hyperparameters is critical in model development because these parameters are solely used to improve the performance of the VGG16 model. The goal of optimization is to boost overall accuracy and recall while lowering losses. In this case study, binary cross entropy loss was utilized to determine gradient and aided in determining weights for neural network layer. Three tests were conducted to increase accuracy and recall while experimenting with various hyper parameter adjustments.

6.1 Experiment 1

The first experiment was conducted using a learning rate of 0.0004 and a dropout value of 0.25. The outcome was evaluated utilizing a number of epochs and the accuracy attained at the training set with epoch 3 is 53.29%. The precision and recall value for the test set was 0.5, and the accuracy for the test set was 50% which indicates that the VGGNet model is unable to detect forge handwritten signature images. When the number of epochs is increased to 5, the training set accuracy drops to 51.56%, while the test set accuracy drops from 50 percent to 48 percent. It's also worth noting that the cross-entropy loss has been increased from 0.79 to 0.91 and as a result, the model's overall accuracy decreased. The number of epochs was increased to 10 in the final setup, and due to which the training set accuracy got increased to 53.47 percent and as well the recall and precision values increased by 0.52 and 0.57, respectively. The epochs were also early stopped, as the loss function did not change from the first epochs. As a result of the VGGNet model's inability to correctly identify real handwritten signatures, the overall accuracy was not adequate. As a result, the second experiment was undertaken to improve the model's performance by adding one layer of convolutional layer, changing the dropout layer to 0.5, and altering the learning parameter. Three independent trials with different epochs and runtime resets were also carried out. As a result, the model can't be overfitted. Table 1 displays the model's performance in terms of evaluation metrics.

Table 1: Executed Results of Experiment 1.

Epochs	Accuracy	Recall	Precision	Cross Entropy Loss
3	0.5329	0.5	0.5	0.79
5	0.5156	0.48	0.4	0.91
10	0.5347	0.52	0.5	0.78

6.2 Experiment 2

The learning rate parameter was set to 0.0008 in the second experiment, which is higher than the prior one. In this experiment, an additional convolutional layer was added to improve performance, and the dropout value was adjusted to 0.5. The number of epochs was first set to 3, and it was discovered that the training set accuracy was 53.47%, and the Test set accuracy was 50 percent, with precision and recall values of 0.5 and 0.459, respectively. The assessment metrics gathered show that the VGG16 model is ineffective for recognizing handwritten signature images. As the model's performance in recognizing handwritten signatures was not adequate, the number of epochs was increased to 5, and evaluation metrics clearly show that the model's accuracy has not improved. Finally, the number of epochs was adjusted to 10, and it was discovered that the training set's accuracy was still 54.37%, while the test set's was 51.25%, with precision and recall values of 0.5 and 0.459, respectively. As a result, the model is unsuitable for identifying handwritten signatures because its accuracy and other evaluation criteria are inadequate. Therefore, the learning rate parameter will be set to a lower value in the third experiment. It was also discovered that there was no role dropout layer and that adding further convolutional layers did not improve the model's performance, thus these will be deleted from the third experiment. Table 2 shown below described the experiments 2 results.

Table 2: Executed Results of Experiment 2.

Epochs	Accuracy	Recall	Precision	Cross Entropy Loss
3	0.5347	0.5	0.4	0.6915
5	0.5347	0.5	0.4	0.6911
10	0.5347	0.52	0.5	0.6908

6.3 Experiment 3

In the third experiment, the learning rate was reduced to 1e-4, and the drop out layer and extra convolution layer were deleted. In the initial setup, the number of epochs was set to three, and the training set accuracy was calculated to be 93.07, with a loss function value of 0.189. The number of epochs is increases from 3 to 5 in the second batch, and the model's accuracy increased to 96.62%, while the cross entropy loss reduced to 0.1014. It illustrates when, the number of epochs increases, the accuracy of the VGGNet model improves. As a result, the number of epochs was increased to ten in the final setup to improve the model's accuracy and other assessment metrics. When the epochs are set to 10, the training set's accuracy rose to 99.31% , while the cross entropy loss value falls to 0.02 and as well the accuracy achieved in the test set was 95%. There were two class labels,Genuine handwritten signatures images (denoted by 1 label in this case study) and forge signatures (denoted by 0 label), the recall and precision value for class zero is 0.89 and 1, respectively, indicating that the VGGNet model is capable of accurately predicting genuine and forged signatures. It emphasizes the significance of learning rate optimization in terms of improving the model's capacity to detect real signatures from faked signatures. So, in terms of evaluation metrics and cross entropy loss, the third experiment produces a better model than the other two models produced by experiments 1 and 2. Table 3 shows model performance in experiment 3.

Table 3: Executed Results of Experiment 3.

Epochs	**Accuracy**	**Recall**	**Precision**	**Cross Entropy Loss**
3	0.9307	0.89	0.90	0.189
5	0.9662	0.9	0.90	0.1014
10	0.9931	0.89	0.0.9	0.0253

6.4 Conclusion of the Experiments

In this section, I will go through in detail the results of the experiments undertaken for handwritten signature authentication research using VGGNet16. Experiments are being undertaken to increase the VGGNet neural network's efficiency and the experiment was set up by altering the hyper parameters of the VGGNet model. In the first experiment, the learning rate was set to 0.0004 and the model was trained for the first time with three epochs. The accuracy of the VGGNet model was found to be 53.29 percent, which is not excellent because it accurately classified just half of the total input handwritten signatures. The number of epochs was increased to 5 in order to boost performance further. It was discovered that the validation accuracy had declined again, and the cross entropy loss function values had also increased, which indicates the cause for dropped accuracy. To test further, the epochs were increased to 10, which resulted in a little increase in the model's accuracy, which was assessed to be 53.41.

The purpose of the second experiment was to improve the efficiency of the VGGNet model. The learning rate was set at 0.0008 with an additional convolutional layer and a dropout layer with a value of 0.5 in this set. The number of epochs was first set to three, and it was discovered that the training set accuracy was 53.47 percent, but the test set accuracy was 50 percent, which is unsuitable for recognizing real handwritten signatures. The number of epochs was then increased to 5, and it was discovered that the model's performance did not improve as the number of epochs was increased. The epochs were raised further, and it was discovered that the validation accuracy of the training set was 54.37%, and the test accuracy was 51.25%,while the precision and recall values were 0.5 and 0.49, respectively. This shows that the model is unable to appropriately categorize true positives as legitimate signs and false positives as forgery signatures, which is inefficient in terms of the model's efficiency.

In the third experiment, the learning hyper parameter was set to 1e-4, and the minimum number of epochs was set to 3, resulting in an accuracy of 93.07 percent and a cross entropy loss of 0.189. The number of epochs was increased to 5 to examine the impact of the increased epochs on the accuracy of the VGGNet model, and it was clearly noted that the model's accuracy was enhanced to 96.92 and the cross entropy or loss function value was also dropped to 0.1014. Thus, the number of epochs has been set to 10 in the final state, resulting in the best accuracy for the VGGNet model of 99.31 percent and a cross entropy loss of 0.0253. As previously mentioned, the model was evaluated using two different class labels, so the precision value for class 0, which is a genuine handwritten signature was 0.89, and for class label 1, which is a forge signature, the precision value was 1, indicating that it accurately detects forge signatures from the authentic handwritten signature images.

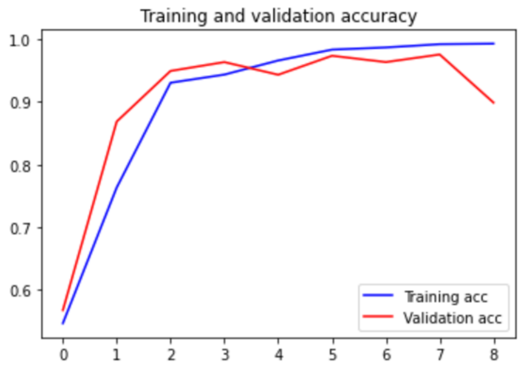

Figure 8: Training and validation accuracy

As a result of the assessment metrics, we can infer that the model acquired from experiment 3 with a learning rate of 1e-4 is the best model in comparison to the other two models generated from experiments 1 and 2. Two graphs have also been plotted, demonstrating that the model's performance is optimal and that the model doesn't overfit and underfit. Three separate trials with various epochs and runtime resets were also conducted to make sure such that model is not underfitting or overfitting.

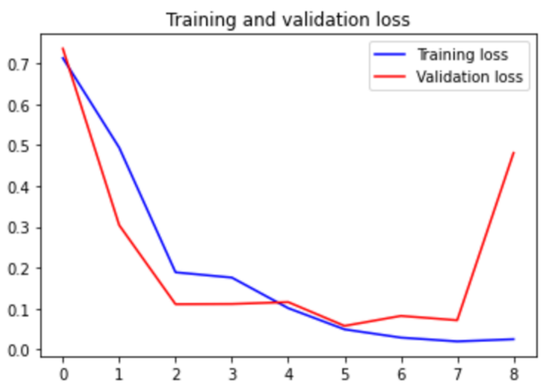

Figure 9: Training and validation loss

The first graph in the figure 8. depicts the relationship between validation and training accuracy. It describes the behavior of the VGG16 model and illustrates that training accuracy will always be larger than validation accuracy because the cross-entropy loss of the model in the training set will be lower than in the test set, as shown in the figure 8. The other graph in the figure 9. describes the relation between training and validation loss. As previously stated, the training set will always have a lower value of loss function than the test set, hence the training set's accuracy will always be higher than the validation set's. Figure 9 shows that the cross entropy loss has been stabilized and reduced after 8 epochs in both the training and test sets. The training-validation loss generalization gap, demonstrating that our model is neither overfit nor underfit.Thus the VGG16 Model obtained from experiment 3 setup delivers the answer for the research question i.e.**To what extent can a VGGNet network be used to detect counterfeit signatures?** This research has improved in terms of accuracy and assessment metrics when compared to other state of the art in the area of handwritten signatures, such as (Moud et al.; 2021).

7 Conclusion and Future Work

This research adds novelty by using convolution layers as feature extractors and the VGGNet model as a classifier network to detect handwritten signatures. The study was conducted using a Kaggle dataset of Dutch handwritten signature images. To recognize genuine handwritten signatures, the VGGNet neural network was used. The optimization of hyper parameters employed in the VGGNet network is the key learning outcome of this study. While developing a model for handwritten signature identification, it was discovered that the learning rate, as well as the number of epochs, is critical. Also, as this research has shown, a faster learning rate does not imply that the model will be more efficient, and the same can be said for the number of epochs. In the field of handwritten signature image authentication, research has outperformed other state of the art methods.

In this study, it was ensured that the model was not overfitted, hence runtime resets were performed after the experiment setup was completed, ensuring that the model should not overfit. The model should be able to authenticate all forms of handwritten signatures made in various languages for future research purposes. Future researchers should also try to test larger batch sizes in the future, which was not doable in this study because of GPU availability from colab and due to memory usage, as image processing requires more GPU. More training data of different images in various languages is needed in the handwritten signature authentication domain, hence the information should be stored in a database so that it may be used for further research.

8 Acknowledgement

I'd want to express my gratitude to Dr. Hicham Rifai for his unwavering support and direction during this research. His input encouraged me to finish the research on time and ahead of schedule. Finally, I'd like to express my gratitude to my family and friends for their continued support in helping me finish my thesis throughout the months.

References

Arora, N., Kumar, A. and Jain, C. (2014). Gmm for offline signature forgery detection, *Proceedings of the 5th International Conference on Confluence 2014: The Next Generation Information Technology Summit* pp. 576–581.

Batool, F. E., Attique, M., Sharif, M., Javed, K., Nazir, M., Abbasi, A. A., Iqbal, Z. and Riaz, N. (2020). Offline signature verification system: a novel technique of fusion of glcm and geometric features using svm, *Multimedia Tools and Applications 2020* pp. 1–20.
URL: *https://link.springer.com/article/10.1007/s11042-020-08851-4*

Bhat, A. R., Nerella, P., Baburaj, P. and S, S. K. (2). A comparative study of transfer learning models for offline signature verification and forgery detection.

Chahi, A., khadiri, I. E., merabet, Y. E., Ruichek, Y. and Touahni, R. (2018). Block wise local binary count for off-line text-independent writer identification, *Expert Systems with Applications* **93**: 1–14.

Coetzer, J., Herbst, B. M. and du Preez, J. A. (2004). Offline signature verification using the discrete radon transform and a hidden markov model, *EURASIP Journal on Advances in Signal Processing 2004 2004:4* **2004**: 1–13.
URL: *https://asp-eurasipjournals.springeropen.com/articles/10.1155/S1110865704309042*

Dey, S., Dutta, A., Toledo, J. I., Ghosh, S. K., Llados, J. and Pal, U. (2017). Signet: Convolutional siamese network for writer independent offline signature verification.
URL: *https://arxiv.org/abs/1707.02131v2*

Fakhiroh, L. A., Fariza, A. and Basofi, A. (2021). Mobile based offline handwritten signature forgery identification using convolutional neural network, *2021 International Electronics Symposium (IES)* pp. 423–429.
URL: *https://ieeexplore.ieee.org/document/9594019/*

Foroozandeh, A., Hemmat, A. A. and Rabbani, H. (2020). Offline handwritten signature verification and recognition based on deep transfer learning, *Iranian Conference on Machine Vision and Image Processing, MVIP* **2020-January**.

Ghosh, R. (2021). A recurrent neural network based deep learning model for offline signature verification and recognition system, *Expert Systems with Applications* **168**: 114249.

He, S. and Schomaker, L. (2020). Fragnet: Writer identification using deep fragment networks, *IEEE Transactions on Information Forensics and Security* **15**: 3013–3022.

Horn, Z. C., Auret, L., McCoy, J. T., Aldrich, C. and Herbst, B. M. (2017). Performance of convolutional neural networks for feature extraction in froth flotation sensing, *IFAC-PapersOnLine* **50**: 13–18.

Hu, J. and Chen, Y. (2013). Offline signature verification using real adaboost classifier combination of pseudo-dynamic features, *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR* pp. 1345–1349.

Iranmanesh, V., Mumtazah, S., Ahmad, S., Azizun, W., Adnan, W., Yussof, S., Arigbabu, O. A. and Malallah, F. L. (2014). Online handwritten signature verification using neural network classifier based on principal component analysis.
URL: *http://dx.doi.org/10.1155/2014/381469*

Ishikawa, C., Marasigan, J. A. U. and Caya, M. V. C. (2020). Cloud-based signature validation using cnn inception-resnet architecture, *2020 IEEE 12th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management, HNICEM 2020* .

Jagtap, A. B., Sawat, D. D., Hegadi, R. S. and Hegadi, R. S. (2020). Verification of genuine and forged offline signatures using siamese neural network (snn), *Multimedia Tools and Applications 2020 79:47* **79**: 35109–35123.
URL: *https://link.springer.com/article/10.1007/s11042-020-08857-y*

Jampour, M., Abbaasi, S. and Javidi, M. (2021). Capsnet regularization and its conjugation with resnet for signature identification, *Pattern Recognition* **120**: 107851.

Javidi, M. and Jampour, M. (2020). A deep learning framework for text-independent writer identification, *Engineering Applications of Artificial Intelligence* **95**: 103912.

Junior, C. A., Silva, M. H. M. D., Bezerra, B. L. D., Fernandes, B. J. T. and Impedovo, D. (2020). Fcn+rl: A fully convolutional network followed by refinement layers to offline handwritten signature segmentation, *Proceedings of the International Joint Conference on Neural Networks* .

Moud, D., Sharma, P. and Kushwaha, R. C. (2021). A comparative analysis on offline signature verification system using deep convolutional neural network, *Lecture Notes in Networks and Systems* **166**: 609–618.
URL: *https://link.springer.com/chapter/10.1007/978-981-15-9689-6$_6$6*

Nathwani, C. (2020). Online signature verification using bidirectional recurrent neural network, *Proceedings of the International Conference on Intelligent Computing and Control Systems, ICICCS 2020* pp. 1076–1078.

Quraishi, M. I., Das, A. and Roy, S. (2013). A novel signature verification and authentication system using image transformations and artificial neural network, *2013 World Congress on Computer and Information Technology, WCCIT 2013* .

Rateria, A. and Agarwal, S. (2018). Off-line signature verification through machine learning, *2018 5th IEEE Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering, UPCON 2018* .

Rodrigues, B., Chaudhari, A., Sakhare, P. and Modi, D. (2016). Prototype for signature verification system using euclidean distance, *Proceedings of the 2015 International Conference on Green Computing and Internet of Things, ICGCIoT 2015* pp. 1624–1627.

Ruiz, V., Linares, I., Sanchez, A. and Velez, J. F. (2020). Off-line handwritten signature verification using compositional synthetic generation of signatures and siamese neural networks, *Neurocomputing* **374**: 30–41.

Sharif, M., Khan, M. A., Faisal, M., Yasmin, M. and Fernandes, S. L. (2020). A framework for offline signature verification system: Best features selection approach, *Pattern Recognition Letters* **139**: 50–59.

Tang, Y. and Wu, X. (2016). Text-independent writer identification via cnn features and joint bayesian, *Proceedings of International Conference on Frontiers in Handwriting Recognition, ICFHR* **0**: 566–571.

Xing, L. and Qiao, Y. (2016). Deepwriter: A multi-stream deep cnn for text-independent writer identification, *Proceedings of International Conference on Frontiers in Handwriting Recognition, ICFHR* **0**: 584–589.

Yapici, M. M., Tekerek, A. and Topaloglu, N. (2019). Convolutional neural network based offline signature verification application, *International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism, IBIGDELFT 2018 - Proceedings* pp. 30–34.

Zheng, Y., Iwana, B. K., Malik, M. I., Ahmed, S., Ohyama, W. and Uchida, S. (2021). Learning the micro deformations by max-pooling for offline signature verification, *Pattern Recognition* **118**: 108008.