Handwritten Signature Forgery Verification using Convolutional Neural Networks

Hassin Da. Khallaf[1]
Email: hassan_dk8080@gmail.com

Abbas Hanon Alasadi[1,2]
Email: abbas.hassin@uobasarh.edu.iq

[1]College of Computer Science and Information Technology, University of Basrah, Basrah, Iraq.
[2]IEEE and ACIT member

Abstract

A behavioral biometric is not based on the individual's physical properties, such as fingerprints or faces, but behavioral ones. Every person has a distinctive signature, primarily used for personal identification and to confirm the authenticity of important papers or legal transactions. In biometric authentication, signature verification is an important study subject. The project aims to create a personal signature-based authentication system. In this work, data extracted from the ICDAR dataset are used, which contains the signatures of Dutch users, both genuine and fraudulent. The data was obtained from the Kaggle website. Two different convolutional neural network strategies have been used to build the proposed model. In the first strategy, the convolutional neural network model was built from scratch; in the second strategy, the pre-trained model, VGG16, was utilized to classify genuine and forged signatures. The findings show that the results of the VGG16 model represent the optimal model for signature forgery verification with an accuracy of 99.8 %, precision of 100%, recall 99.5%, F1-score of 99.4%, and training time of 18 min 52 s.

Keywords: Handwritten Signature; Offline Signature; Convolutional Neural Networks; Deep Learning.
DOI Number: 10.14704/nq.2022.20.11.NQ66075 **NeuroQuantology 2022; 20(11): 770-780**

1. Introduction

Today, digital imaging has grown in popularity since it allows everyone to capture an unlimited number of high-quality photographs fast and free and save them readily on various digital devices or share them on the Internet. Public health services, political blogs, social media platforms, judicial probes, education systems, the military forces, and corporations are examples of how digital photographs are used in nearly every field nowadays [1].

Rapid advancements in digital technology have created and distributed a massive volume of photographs over the last few years. At the same time, it has become relatively easy to change images and videos using photo-editing programs such as Canva, CorelDRAW, PicMonkey, PaintShop Pro, and many other applications. Authenticating the legitimacy of these photographs becomes a severe difficulty for such social media networks. According to cybersecurity experts [2], hackers can access 3-D medical scans of patients and change or erase malignant cell pictures. A recent study found that AI-modified scans misled surgeons, potentially leading to misdiagnosis and insurance fraud[3]. Furthermore, edited political images [4] shared across social media platforms can mislead and affect public perceptions and judgments. According to studies, some types of photographs are more likely to be reused and, in some situations, used in online terrorism communication channels via media sources.

All of these cases fall under photo forgery, so according to Merriam-Webster, digital image forgery is defined as "falsely and fraudulently altering a digital image". Image forgery is not a new concept; it dates back to 1840. Hippolyte Bayard, a French photographer, created the first tampered image titled "Self Portrait as a Drowned Man," in which Bayard professed to commit suicide[5] [6].

Signatures are commonly used as a unique way of identifying and verifying a person's identity. Offline signature verification is one of the more complex tasks of pattern recognition. As a behavioral biometric trait, the signatures are marked with intrapersonal and interpersonal variations. Developing a signature verification system capable of countering these

eISSN 1303-5150

variations is daunting. This study's main objective is to develop an offline handwritten signature verification system that uses a deep convolution neural network (ConvNet) to differentiate between genuine and forged signatures based on features extracted using a convolutional neural network (CNN).

2. Related Works

Recent research on handwritten signature verification has investigated various CNN architectures with and without a separate classifier. This section presents several studies using different methods to define forged signature images.

In 2016, Hafemann*et al.*[7] used two CNN architectures to extract features from an offline signature: Alex Net and VGG Net. The researchers used two datasets: GPDS-960 and PUC-PR, and the experimental results indicate that the features learned by a subset of users are discriminative for the other users.

Alvarez *et al.*[8] proposed automating the signature verification process through convolutional neural networks. Their model is based on the VGG16 architecture, and they train it using transfer learning on the ICDAR 2011 SigComp dataset. Their results achieve a classification accuracy of 97% for Dutch signatures and 95% for Chinese signatures when determining whether a given signature is a forgery or genuine.

In 2018, Hanmandlu*et al.*[9]used two CNN architectures to extract features: LeNetand AlexNet. On the GPDS-960 database, an SVM classifier utilizing the Cubic kernel in conjunction with AlexNet performed very well, with a recognition rate of 96.6%.

In 2019, Jahandad*et al.*[10] used GoogLeNet Inception V1 and V3 CNN architectures. The GPDS Synthetic Signature Database was used to classify the signatures of 1000 users, each of which had 24 genuine (or original) signatures and30 forged (or fake) signatures. The experimental results show that the Inception V1 model outperforms the Inception V3.

In 2020, Kao and Wen [11] proposed a signature verification method based on explainable deep learning and local feature extraction on a single reference sample. They used two architectures: VGG19 and Inception V3. The researchers used the open-source dataset, Document Analysis and Recognition (ICDAR) 2011SigComp, to train their system and verify whether a questioned signature was genuine or a forgery. According to the experimental results, they achieve an accuracy of between 94.37 percent and 99.96 percent.

In2021,Kuriakose*et al.*[12]usedaVGG-16convolutionneuralnetwork architecture as a feature extractor. They fedthe datainto variousclassifiers for classification, includingrandom forest,k-nearestneighbors,extratree,andsupportvector machine.Finally,theclassifieroutputisfedinto anartificial neural networkfor final prediction.The experimentalresultsshowthattheproposedalgorithmperformswell, withanaccuracyof 97.3percent.

3. CNN Architecture

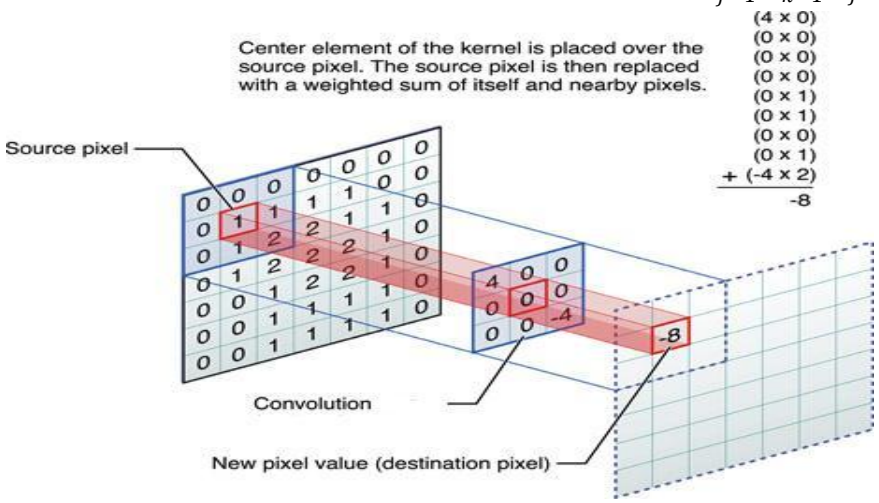
TheConvolutionalNeuralNetworkisa populardeep neuralnetwork modelforcomputer vision.Ontherawinputimage,itappliestrainablefilters and localneighborhoodpoolingprocedures,yieldingahierarchy ofprogressively abstractfeatures.CNNsignificantlyaffectedcomputer visionandimage interpretationingeneral,allowingthemtooutperform competitorsinvisualobject recognition[13].CNNconsistsof the followinglayers.

3.1 Convolution Layer

A convolution layer is usually the first layer of a CNN following the input layer. The image of the input layer can be considered an array of W* H* C pixel values, where W and H are the image's width and height, and C is the image's channel. C = 3 is commonly used in RGB images. The convolution layer will use an N* N matrix as a filter (usually N is 3, 5, or 7) to perform the convolution operation from the upper left corner of the image. The values in the matrix will be multiplied by the corresponding values in the image covered by the filter. All the products will be added to form the convolution value at the filter's location [14]. Figure (1) illustrates the visualization of theConvolution process. The convolution of the filter is denoted asEquation (1)[15].

eISSN 1303-5150

$$F \times I = \sum_{j=1}^{n} \sum_{k=1}^{n} W_j \; K^i jK \quad \ldots (1)$$



Figure(1):Visualizationofthe convolutionprocess[16].

3.2 Pooling Layer

After convolutional layers or nonlinearity, pooling layers are frequently used. Downsamplingis accomplished by pooling. They decrease the number of parameters in the convolution as the spatial scale is reduced [17]. Figure (2) illustrates the visualization of the pooling operation.

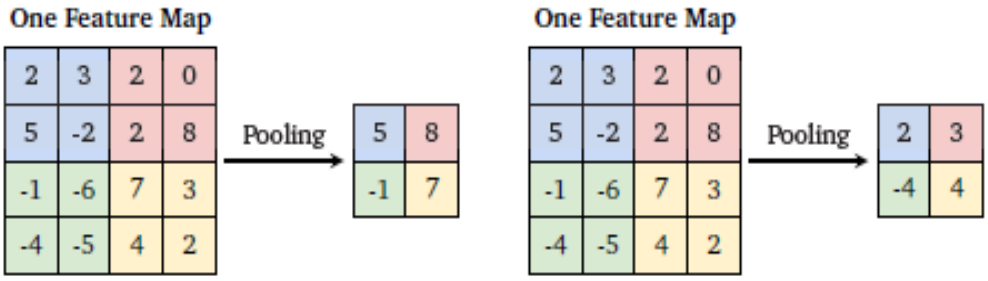


Figure (2):Representationof the Poolingoperation[18].

3.3 Fully Connected Layer

Fully connected (FC) layers use nonlinear algorithms to transform the output of the previous pooling layer to another space. Fully connected layers contain enormous weights compared to other layers. Hence, training time in these layers increases accordingly [19]. The network will employ a Fully Connected Layer to map higher-level activation mappings to the classification of the output layer and construct an n-dimensional vector, where n is the number of output layer classifications, after feature extraction from the preceding layer. This n-dimensional vector [20] represents the probability of the detected image in N classifications.

4. Proposed Method

Inthiswork,theframeworkstepsoftheproposedmethodarepresentedin

Algorithm(1).

Algorithm (1): Proposed Signature Forgery Verification algorithm.

eISSN 1303-5150

Inputs: ICDAR dataset images

Step 1: Read all genuine and forged signatures of an individual.

Step 2: Apply pre-processing methods

- Convert images into RGB
- Labeling the dataset
- Resize the images
- Normalize the images

Step 3: FOR(EPOCHmax)

Step 4: FOR(BACHmax)

Step 5: Shuffle the dataset and select separate input and output signature samples randomly.

Step 6: Train the proposed convolutional neural network

- VGG16
- InceptionV3

Step 7: Extract the features using the trained CNN.

Step 8: END (Inner Loop)

Step 9: END (Outer Loop)

Outputs: Forged or genuine image and the evaluation matrices calculation

4.1 Dataset Description

In this work, data extracted from the ICDAR dataset containing the signatures of Dutch usershave been used, both genuine and fraudulent. The data was obtained from the Kaggle website. The dataset is divided into two main groups: training and testing, and each subset is divided into genuine and forgery. The total number of genuine and forged signatures in the training set is 23,206. The total number of genuine and forged signatures is 5748 in the testing set. Figure (3) displays examplesof genuine and forged signatures.

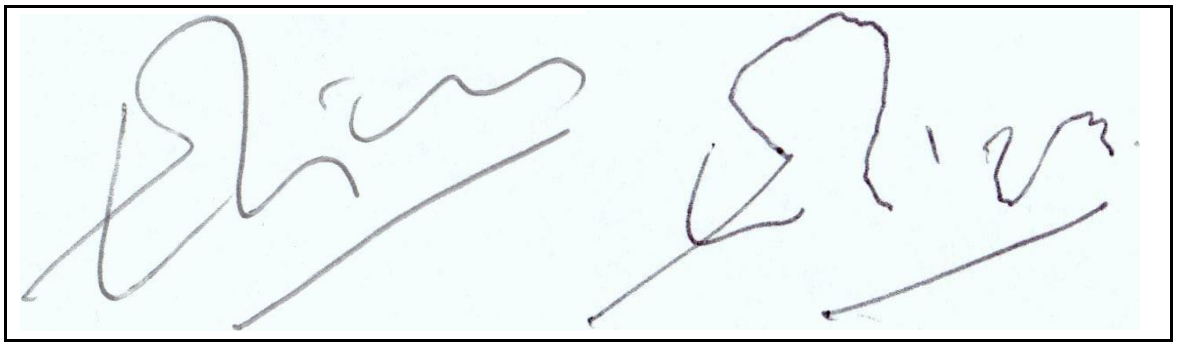


Figure (3):Signaturesample(a)genuinesignatureimage, (b)forgedsignature image.

4.2 Dataset Pre-processing

Data pre-processing is done to enhance or make the data more compatible with the model. Many data pre-processing techniques can be applied for data enhancement. Algorithm (2) depicts the pre-processing steps used in this research work, and every step is explained as follows:

a) Dataset Labelling

Genuine signatures are labeled 0, and forgery signatures are labeled 1.

b) Data Resize

The entire signature images matrix was resized and rescaled to a standard resolution of 300 × 300 shapes. Since NN accepts images of identical size, images were resized because there was significant variation in the size of the images in the dataset. Figure (4)shows the result of this operation.

eISSN 1303-5150

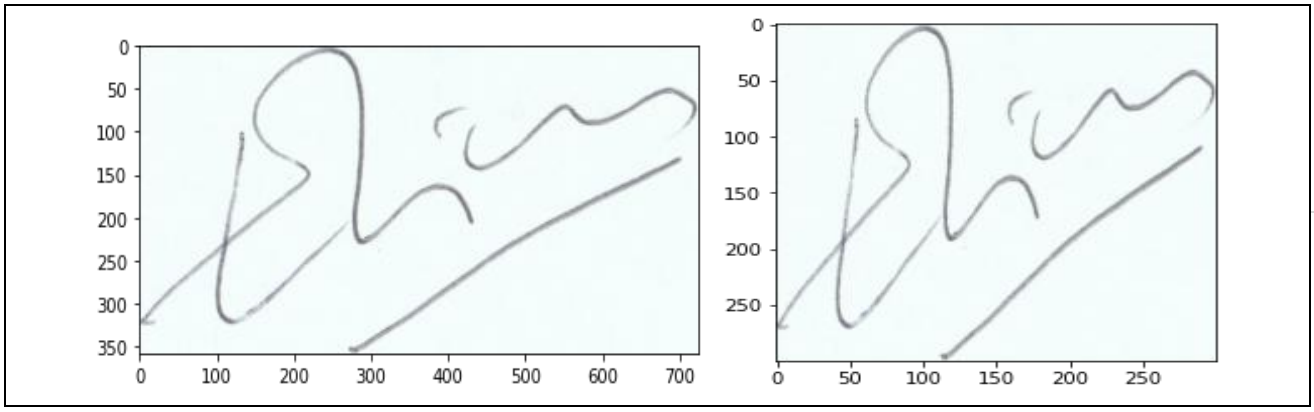


Figure (4): Signature image sample: (a) original image size, (b) resized signatureimage.

c) Data Normalization

Data normalization is a crucial step commonly employed in CNN systems to ensure numerical stability. A CNN model that has been normalized is more likely to train quicker and have steady gradient descent.

Algorithm(2): Pre-processing algorithm.

Inputs: ICDAR dataset images

Step 1: Upload Dataset from G**oogle Drive to Google Colab**

Step 2:

For*i*=1 to N do // N number of images in the ICDAR dataset.
Create Array_Label of Training Set
Create Array_Label of Testing Set
For *i*=1 to Z do // Z number of images in Training Set, Testing Set.
Set the label of the forgery signature image to 1
Set the label of the genuine signature image to 0
Endfor

Step 3: New_Image (x,y) =Image(300,300)
// x is height of image
// y is the width of an image

Step4: New_Image (x,y) =Image((1/255 *(x,y))

Output: Processed image ready for training

4.3 Building the Model

Two different CNN strategies have been used in this phase to build the proposed model.

4.3.1 Build CNN from Scratched

In this step, CNN architecture has been built, as depicted in Table (1).

Table (1): The Layers structures of CNN Architecture.

Feature Extraction Part:	
Layers	**Numbers**
Convolutional layer	3
• Activation Function	• Relu
• Filters:	
in the first layer	16
in the second layer	16
in the third layer	32
Kernel-size	(3,3)
Average pooling	2

eISSN 1303-5150

Pool-size	(2,2)
Classification Part:	
Layers	**Numbers**
Dense	3
First Dense • Units • Activation Function	 64 Relu
Second Dense • Units • Activation Function	 64 Relu
Third Dense • Units • Activation Function	 2 Softmax

4.3.2 Pre-trained CNN Model

Pre-trained model, which is VGG16, was utilized in this work to classify genuine and forged signatures. The VGG16 network contains numerous layers, including 13 convolution layers and three FC layers. The filters used in this model are of sizes 3X3 with a stride of 1 and a padding of 1.

4.4 Model Training

In the training process, we used a backpropagation algorithm for training the models (explained in detail in chapter two). A loss function (or cost function) called binary cross-entropy (BCE) has been used to estimate the deviation between the predicted value and the actual label and then train the network to minimize the loss value. The lower the loss value, the closer the expected result is to the accurate label. Adam optimizer has been used for adjusting the weight of the layers. The training process for each CNN model passes throw two steps:

- The first step is **Feature Extraction:**A convolution neural network algorithm isused to perform the feature extraction process. During the feature extraction process, the system examines a given pattern and records certain features to submit the structured data as an observation sequence. In addition, this process recognizes and discriminates a person's signature from one another. A feature extraction process is essential in improving a system's accuracy.
- The second step is the Classification process: The fully connected layer has been used to perform the Classification process. The output layer is a softmax function that handles the probability problems.

The training process is depicted in Algorithm (3). Training the model has two strategies:

4.4.1 Training CNN Model from Scratch

In this strategy, we have trained five differentarchitectures from scratch. Training propertiesfor each model architecture are shown in Table (2). The early stopping technique has been used to prevent the overfitting problems with patience = 30 and mini-delta = 0:

Table (2): Training Properties for the Scratch Model

Model	Bach-size	Epochs	Learning rate
Architecture C	32	30	0.001

4.4.2 Training the Pre-Trained Model

In this step, we only have to per-train the lastthree layers for the **VGG16**and freeze all previous layers. This prevents the early training process from undermining valuable initial weights. The training properties for each pre-trained model are shown in Table (3).

Table (3): Training Properties for the Pre-trained Model9.

eISSN 1303-5150

Model	Bach-size	Epochs	Learning rate
VGG16	64	30	0.0001

Algorithm(3): Proposed Signature Forgery Verification algorithm.

Inputs: Processed image

Step 1: Splitting Dataset into 70% training_set& 30% testing _ set
Step 2: Splitting training_set& testing _ set into Batches
Step 3: **For***i* = 1 to N // N number of training Batches
The training_set batches are entirely passed through the CNN network in a forward phase.
Step 4: calculate the loss using cross-entropy loss
Step 5: A backward phase, where gradients are backpropagated and the optimizer will update the weights and biases
Step 6: **Endfor**

Outputs: accuracy and loss for training_set, testing_set to check the performance of the CNN model

4.4.3 Model Evaluating

After the network training and validation, to further test whether an utterly unknown author can use the networks, we test the networks with new authors' signatures that are not present in the training and verification dataset stage. Then the following metrics are used to evaluate the performance of our network: training, validation, and testing: Accuracy, Precision, and F1- score.

5. Results and Discussion

All experiments were carried out on the Collab platform, provided by Google for free, using the Python language and the Keras, Pandas, NumPy, and Matplotlib libraries. The accuracy curve's trend from the first to the 30th iteration of model training and evaluation is depicted in Figure (5). At the same time, Figure (6) showsthe loss function curve's trend from the first to the 150th iteration of model training and evaluation. Figure (7) illustrates the confusion matrix.

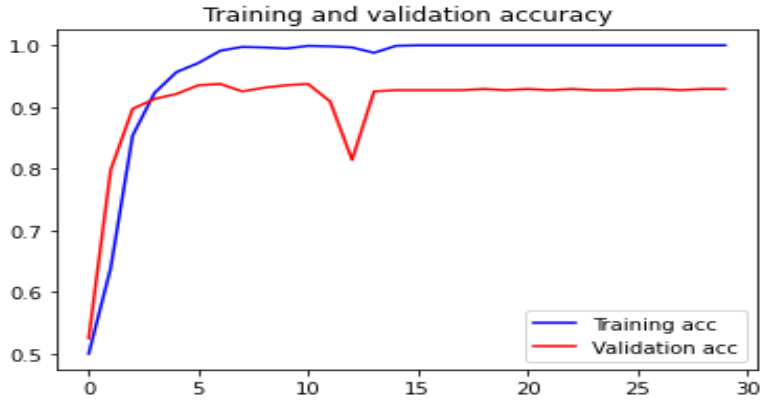


Figure (5): The Accuracy of Training and Validation across Epochs

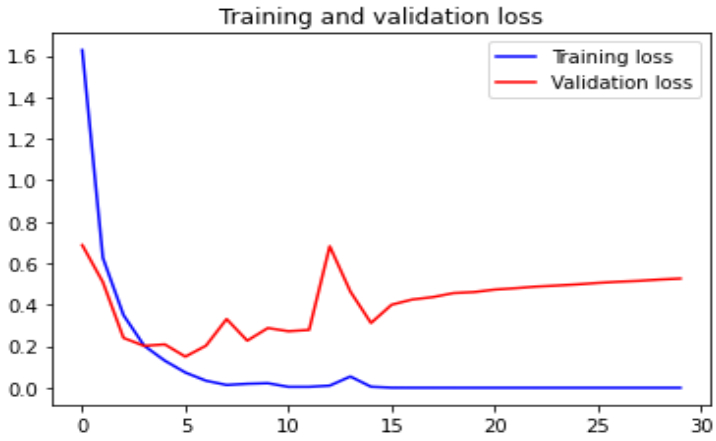


Figure (6): The Training Loss and Validation Loss across Epochs.

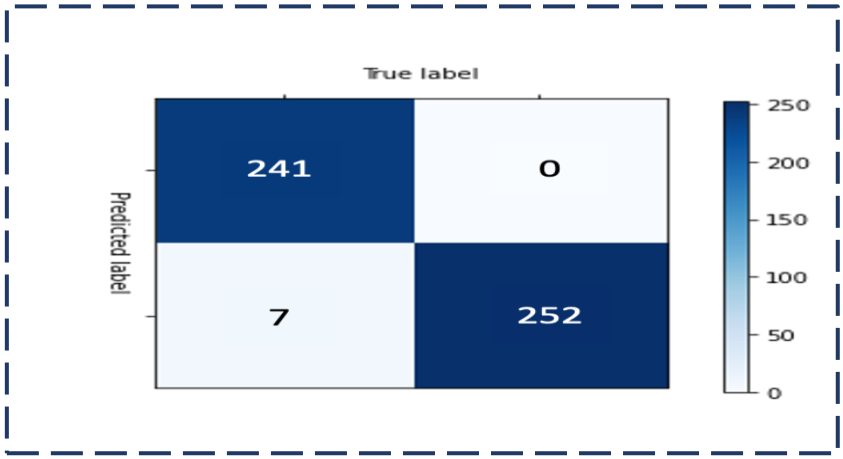


Figure (7): The confusion matrix for scratch CNN *model*.

The VGG16 model's accuracy curves trend from the first to the 30th iteration of model training and evaluation is depicted in Figure (8). At the same time, the loss function curve's trend from the first to the 30th iteration of model training and evaluation is shown in Figure (9). Figure (10) illustrates the confusion matrix.

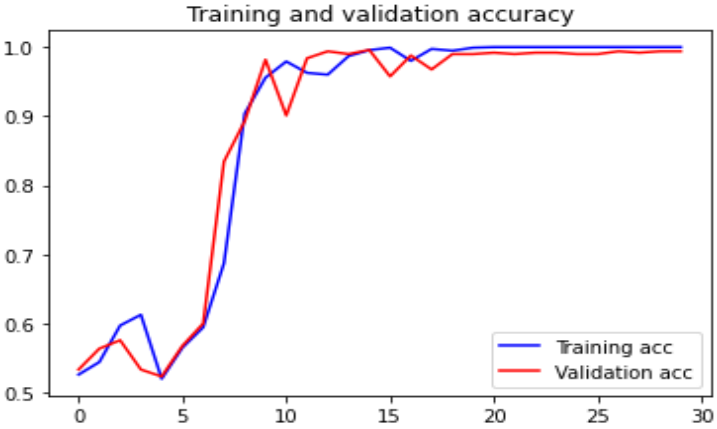


Figure (8): The Accuracy of Training and Validation across Epochs

eISSN 1303-5150

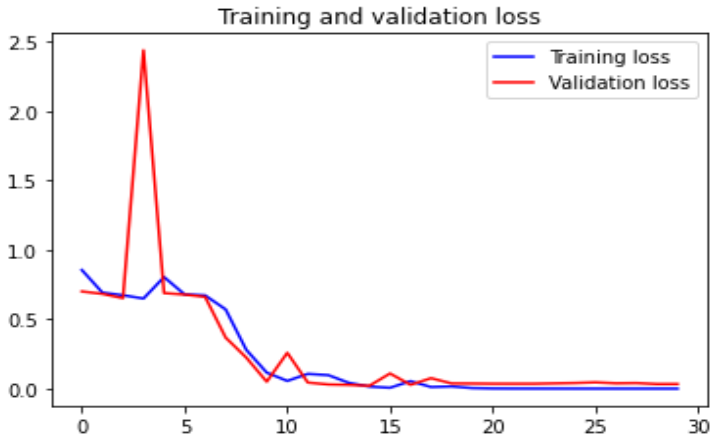


Figure (9): The Training Loss and Validation Loss across Epochs.

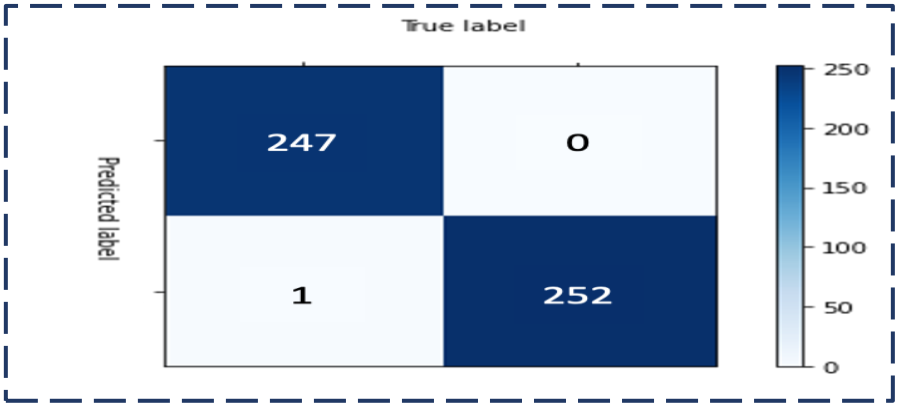


Figure (10): The confusion matrix for pre-trained model - VGG16.

The accuracy, precision, recall, and F1-Score were utilized to compare the models used in this work. Another critical parameter to consider is the time required for a system to learn. Table (4) shows the evaluation matrices for all models. The best model for the verification process is the one that was obtained from the second research strategy that gave us a VGG16 network, which is a good model for the operation of building an intelligent system that can distinguish between fake and original signatures that will help reduce forgery of essential papers and documents. Comparing the significance model with previous studies that used the same dataset and different architectures of deep learning is displayed in Table (5). The proposed model has achieved high performance via all evaluation criteria.

Table (4): The evaluation matrices comparison for all models.

Matrices	Scratch CNN MODEL	VGG16 MODEL
Accuracy	98.6 %	99.8 %
Precision	100%	100%
Recall	97.1%	99.5%
F1-Score	98.5%	99.4%
Training Time	2min 12s	18min 52s

Table (5): Comparing the Suggested Model with the Previous Study.

Matrices	Alvarez *et al.*[8]	The proposed model
Validation Accuracy	97 %	99%
Test Accuracy	94%	99.4%

eISSN 1303-5150

6. Conclusion

This work aims to develop an offline handwritten signature verification system that uses a deep convolution neural network (ConvNet) to differentiate between genuine and forged signatures based on features extracted using CNN using two strategies. The first is training a convolution neural network from scratch, while the other is using the pre-trained convolution neural network models. Our findings show that the results obtained from the VGG16 model represent the optimal model for signature forgery verification with high accuracy. The reason for that is the practical structure of the pre-trained model.

7. References

[1] Kadam,K.,Ahirrao,S.andKotecha,K.,2022.EfficientApproachtowards DetectionandIdentificationofCopy MoveandImageSplicingForgeriesUsing MaskR-CNNwithMobileNetV1.Computational IntelligenceandNeuroscience, 2022, pp.1-21.

[2] Kumar Y, Koul A, Singla R, Ijaz MF. Artificial intelligence in disease diagnosis: a systematic literature review, synthesizing framework and future research agenda. Journal of Ambient Intelligence and Humanized Computing. 2022 Jan 13:1-28.

[3] Jaafar RH, Rasool ZH, Alasadi AH. New copy-move forgery detection algorithm. In2019 International Russian Automation Conference (RusAutoCon) 2019 Sep 8 (pp. 1-5). IEEE.

[4] Hosny, K.M., Mortda, A.M., Fouda, M.M. and Lashin, N.A., 2022. An Efficient CNN Model to Detect Copy-Move Image Forgery. IEEE Access, 10, pp.48622-48632.

[5] K. Meena and V. Tyagi, "Image Forgery Detection: Survey and Future Directions", Data, Engineering and Applications, pp. 163-194, 2019. Available: 10.1007/978-981-13-6351-1_14 [Accessed 30 April 2022].

[6] Alasadi AH, Jaffar FH. Fingerprint verification system based on active forgery techniques. International Journal of Computer Applications. 2018;180(11):6-10.

[7] L.Hafemann,R.SabourinandL.Oliveira, "Writer-independentfeature learningforOffline SignatureVerificationusingDeepConvolutionalNeuralNetworks", *2016InternationalJoint Conferenceon Neural Networks (IJCNN)*, 2016.Available: 10.1109/ijcnn.2016.7727521.

[8] G.,Alvarez,B.,Sheffer,& M.Bryant,"Offlinesignatureverificationwithconvolutional neural networks", *2016 Technical report, Stanford University,*2016.

[9] M.Hanmandlu,A.SronotharaandS.Vasikarla,"DeepLearning-basedOffline Signature Verification",*20189thIEEE AnnualUbiquitousComputing,Electronics& Mobile Communication Conference(UEMCON)*, 2018.Available: 10.1109/uemcon.2018.8796678.

[10] Jahandad,S.Sam, K.Kamardin,N.AmirSharifandN.Mohamed, "OfflineSignatureVerification using Deep Learning Convolutional Neural Network (CNN) ArchitecturesGoogLeNetInception-v1andInception-v3",*ProcediaComputerScience*,vol.161 ,pp.475-483,2019. Available: 10.1016/j.procs.2019.11.147.

[11] H.KaoandC.Wen,"AnOfflineSignature VerificationandForgeryDetectionMethod Based on a Single Known Sample and an Explainable Deep Learning Approach",*Applied Sciences*, vol. 10, no. 11,p. 3716, 2020. Available:10.3390/app10113716.

[12] Y.Kuriakose,V.Agarwal,R.DixitandA. Dixit,"ANovelTechniqueforFakeSignature DetectionUsingTwo-

TieredTransferLearning", *ProceedingsofInternationalConferenceon Computational Intelligence*, pp. 45-58, 2021. Available: 10.1007/978-981-16-3802-2_4.

[13] Y.E.Abdalla, "DetectionOfCopy-MoveForgeryIn DigitalImagesUsing Different Computer Vision Approaches", Ph.D., Memorial University of Newfoundland,2020.

[14] H.AhnandC.Yim,"ConvolutionalNeural NetworksUsingSkipConnectionswithLayerGroups forSuper-ResolutionImageReconstructionBasedonDeepLearning",*AppliedSciences*,vol. 10,no.6, p. 1959,2020. Available:10.3390/app10061959.

[15] S.Aleshin-GuendelandS.Alvarez,"Examining theStructureofConvolutionalNeuralNetworks", Master's degree, Boston College, 2017.

[16] N. Toyonaga, "Convolutional NeuralNetworksForPhasePredictionInDeep Tissue Microscopy", Masterdegree, Stanford University, 2017.

[17] A. Khan,A.Sohail,U.ZahooraandA.Qureshi,"Asurveyoftherecentarchitecturesof deep convolutionalneuralnetworks", *ArtificialIntelligenceReview*,vol.53,no. 8,pp.5455-5516,2020. Available:10.1007/s10462-020-09825-6.

[18] P.Fürnkranz,H.DangandS.Luthardt,"UsingConvolutionalNeuralNetworkstodistinguish vehicleposeand vehicleclass", master,TechnischeUniversitätDarmstadt, 2016.

[19] Q.Xu,M.Zhang,Z.GuandG.Pan,"Overfittingremedybysparsifyingregularizationonfully- connected layers of CNNs",*Neurocomputing*, vol. 328, pp. 69-74, 2019. Available:10.1016/j.neucom.2018.03.080.

[20] Z. Chen, Z. Xie, W. Zhang andX. Xu, "ResNetandModelFusionforAutomaticSpoofingDetection",*Interspeech2017*,2017.Available:10.21437/interspeech.2017-1085[Accessed20June2022].