IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

HUMAN SIGNATURE VERIFICATION USING CNN WITH TENSORFLOW

Dr.R.Palson Kennedy[1], Nithish Kumar[2], Vinodh Kumar[3], Yogesh[4]

Professor[1],Student[2,3,4]

Department of Computer Science Engineering,

PERI INSTITUTE OF TECHNOLOGY

Abstract

A signature of a person is simply a handwritten sign or marks that resembles to his/ her name often stylized and unique and indicates person's identity, intent, and consent. Mostly used for different purpose like authenticating checks, drafts, certificates, approvals, letters, and other legal documents. Since signature are used in such critical activities, confirmation of their authenticity are significantly important.. Traditionally signature was manually compared with copies of genuine signatures for verification. This simple method may not be sufficient as the technology is becoming more and more advance and with advancing techniques of forgeries and falsification of signature. So, inorder to tackle such problem new efficient tool is needed and this project proposes such signature verification tool which can assist human in correct decision making in authentication of handwritten signature.

For such authentication of signature this project presents an applications of which facilitates the feature of human signature verification using the convolution neural network approach. This software is able to train the network with new dataset of signature and validate the authenticity of new signature of trained class. User can also perform experiment and analyze the training and verification of model with features result analysis feature. This software consists very efficient user interface so that any non-technical person can use the software without any difficulty. The end result of project was very optimistic and iten couragesus for further research and development in this field.

Introduction

To detect the human hand signature, we planned to design deep learning technique so that a person with lesser expertise in software should also be able to use it easily. The proposed system is to predict whether the human hand signature is forged or genuine. Samples of more number of images are collected that comprises of different classes such as genuine and forged signatures. Different number of images are collected for each classes that was classified into input images.

We proposed a Deep Learning (DL) based offline signature verification method to prevent signature fraud by malicious people. The Deep Learning (DL) method used in the study is the Convolutional Neural Network (CNN). It is predicted that the success of the obtained results will increase if the CNN method is supported by adding extra feature extraction methods and successfully classify the human hand signature.

The system of human signature verification works on Deep learning algorithm which contains several "layers" of neural network algorithms, in which signatures passes through each layer giving a simplified representation of the data to the next layer. Most of the machine learning algorithms work well on the datasets which have up to a few hundred features.

However, an unstructured dataset that is an image has a large number of features that this process becomes cumbersome or completely unfeasible. The Deep learning algorithm learns progressively more about the image of the signatures as it goes through each of the neural network layer. The result of the signature which is real or forged is found on the final output layer and displayed on the screen.

Literature Survey

The area of Handwritten Signature Verification has been broad lyre search edin the last decades and still remains as an open research problem. This project focuses on offline signature verification, characterized by the usage of static(scanned)images of signatures, where the objective is to discriminate if a given signature is genuine(produced by the claimed individual), or a forgery (produced by an impostor). Wepresentanoverviewofhowtheproblemhasbeenhandledbyseveralresearchersinthepastfew decades and there centad vancements in the field.

1. BIOMETRIC SIGNATURE VERIFICATION SYSTEM

"Biometric signature verification system based on freeman chain code and k-nearest neighbor" [1], In Stage 1, problem background is analysed where three problems were identified. The first one is related to entire SVS. As signature is a type of biometric that may change with mood, environment and age, some solutions for this problem are defined. A good signature database must be updated in a few specific times so that the database is relevant to be used from time to time. Besides, a person must sign in a consistent manner to construct a series of signatures that are almost similar between each other. The second problem relates to FCC generation that failed to extract from broken parts of signature. Thus, only the largest contiguous part of the signature is chosen to extract the FCC. The third problem related to verification in order to achieve a good result. Earlier processes, namely pre-processing and feature extraction, must work efficiently to achieve the desired results from k-NN.

In Stage 2, an offline signature database which is known as MCYT Bimodal Sub corpus Offline Signature has been selected for use. In Stage 3, there are two parts of feature extraction involved in this research. The first part is regarding to FCC feature. Chain code representation describes the outline for signature image by recording the direction of where is the location of the next pixel and corresponds to the neighbourhood in the image.

2. MULTIPLE NEURAL CLASSIFIERS

"Signature verification using multiple neural classifiers" [2] - A prototype recognition system was implemented using C on Sun's Spark System. For experimentation, samples were collected from ten different individuals. Fifteen samples of genuine signatures were collected from each individual. We have used, in addition, 100 random forgeries for evaluating the system. To evaluate effectiveness of the approach a set of experiments were performed. In all the experiments, the classification nets were trained with five randomly selected samples of genuine signature of each person. The number of output nodes of these nets was same as that of the number of persons involved in the experiment (i.e.10).

3. TWO-STAGE NEURAL NETWORK CLASSIFIER

"A new signature verification technique based on a two-stage neural network classifier" [3] - This paper proposes a new off-line signature verification and recognition technique. The entire system is based on 160 features grouped to three subsets and on a two-stage neural network classifier that is arranged in an-one-class-one-network scheme. During the training process of the first stage, only small, fixed-size neural networks have to be trained, while, for the second stage the training process is straightforward. In designing the proposed system, most of our efforts were towards of embodying most of the intelligence to the structure of the system itself. No feature reduction process was used and the basic rule of thumb in deciding which features to include and which not was ''use all features and leave the neural networks decide which of them are important and which are not''.

4. DEEP HSV

"Deep HSV: User-independent offline signature verification using two-channel CNN" [4], In this paper, we proposed a 2-Channel-2-Logit network structure that greatly improves the accuracy of writer-independent off-line handwritten signature verification. The input to the network is the concatenation of reference and query signatures. The output of convolutional layers are two logits that measure the similarity between reference and query signatures. We explicitly add dropout layers and a 2-Logit layer to make the network less prone to overfitting issues. We perform experiments on the widely used databases to show that 2-Channel-2-Logit outperforms SOTA by a large margin. Especially, the accuracy improves from 77.76% to 90.05% on the latest GPDS-Synthetic database. The proposed network is also robust to signature image format and achieves same accuracy on binarized signature images. We also study the generalization ability by conducting cross datasets validation. Furthermore, we investigate that Siamese network is prone to over-fitting on small datasets and find that 2-Channel network is superior to measure image similarity in this case.

5. SIGNATURE IMAGE GENERATION

"Synthetic off-line signature image generation" [5] - This paper proposes a novel methodology to generate static/off-line signatures of new identities. The signature of the new synthetic identity is obtained particularizing the random variables of a statistical distribution of global sig- nature properties. The results mimic real signature shapes and writing style properties, which are estimated from static signature databases. New instances, as well as forgeries, from the synthetic identities are obtained introducing a natural variability from the synthetic individual properties. As additional novelty, an ink deposition model based on a ball- point is developed for realistic static signature image generation. The range of the static signature generator has been established matching the performance obtained with the synthetic databases and those obtained with two public Databases.

6. STATIC SIGNATURE SYNTHESIS

"Static signature synthesis: A neuromotor inspired approach for biometrics" [6],In this paper we propose a new method for generating synthetic handwritten signature images for biometric applications. The procedures we introduce imitate the mechanism of motor equivalence which divides human handwriting into two steps: the working out of an effector independent action plan and its execution via the corresponding neuromuscular path. The action plan is represented as a trajectory on a spatial grid. This contains both the signature text and its flourish, if there is one. The neuromuscular path is simulated by applying a kinematic Kaiser filter to the trajectory plan. The length of the filter depends on the pen speed which is generated using a scalar version of the sigma lognormal model. An ink deposition model, applied pixel by pixel to the pen trajectory, provides realistic static signature images. The lexical and morphological properties of the synthesized signatures as well as the range of the synthesis parameters have been estimated from real databases of real signatures such as the MCYT Off-line and the GPDS960 Gray Signature corpuses. The performance experiments show that by tuning only four parameters it is possible to generate synthetic identities with different stability and forgers with different skills. Therefore it is possible to create datasets of synthetic signatures with a performance similar to databases of real signatures. Moreover, we can customize the created dataset to produce skilled forgeries or simple forgeries which are easier to detect, depending on what the researcher needs. Perceptual evaluation gives an average confusion of 44.06% between real and synthetic signatures which shows the realism of the synthetic ones. The utility of the synthesized signatures is demonstrated by studying the influence of the pen type and number of users on an automatic signature verifier.

7. FUSION OF LOCAL AND GLOBAL INFORMATION

"An on-line signature verification system based on fusion of local and global information" [7] - Automatic extraction of identity cues from personal traits (e.g., signature, fingerprint, voice, and face image) has given raise to a particular branch of pattern recognition, biometrics, where the goal is to infer identity of people from bio- metric data. The increasing interest on biometrics is related to the number of important applications where an automatic assessment of identity is a crucial point. Within biometrics, automatic signature verification has been an intense research area because of the social and legal acceptance and widespread use of the written signature as a personal authentication method. This work is focused on on-line signature verification, i.e., the time functions of the dynamic signing process (e.g., position trajectories, or pressure versus time) are available for recognition.

8. DISCRIMINATIVE FEATURES MINING

"Discriminative features mining for offline handwritten signature verification" [8],Signatureverificationisanactiveresearchareainthefieldofpatternrecognition. It is employed to identify the particularpers on with the help of his/her signature's characteristics such as pen pressure, loops shape,speed of writing and up down motion of pen, writing speed, pen pressure, shape of loops, etc. Inorder to identify that person.However,in the entire process, features extraction and selection stage is of prime importance. Since several signatures have similar strokes,characteristics and sizes.Accordingly,thispaperpresentscombinationoforientationoftheskeletonandgravitycenterpointtoextractaccuratepatternfeaturesofsignaturedatainofflinesignatureverificationsystem.Promisingresultshaveprovedthesuccessoftheintegrationofthetwomethods.

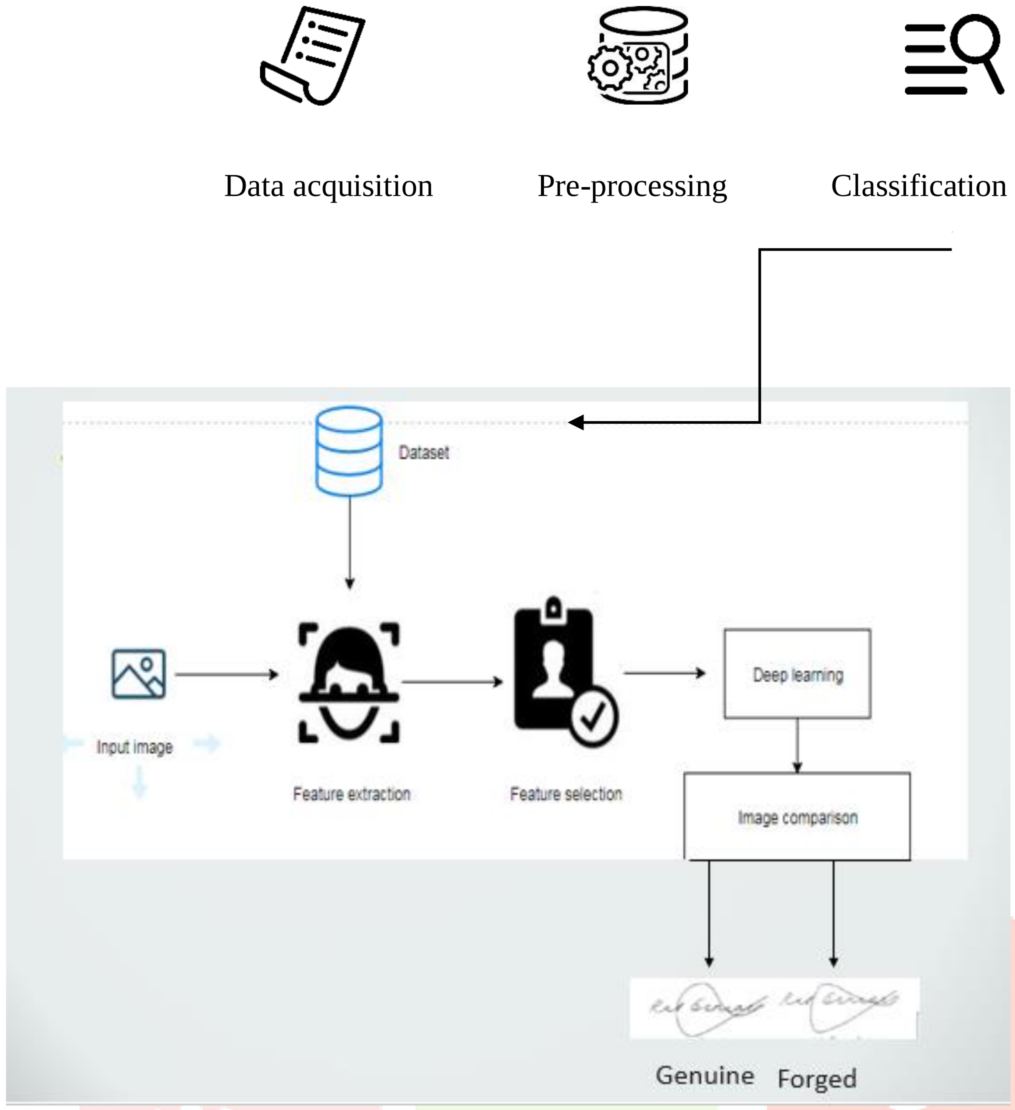

Fig 1. Architecture diagram

LIST OF MODULES

- Manual Net
- AlexNet
- LeNet
- Deploy

MODULE DESCRIPTION

[1] IMPORT THE GIVEN IMAGE FROM DATASET AND TRAINING MANUAL CNN

We have to import our data set using keras preprocessing image data generator function also we create size, rescale, range, zoom range, horizontal flip. Then we import our image dataset from folder through the data generator function. Here we set train, test, and validation also we set target size, batch size and class-mode from this function we have to train using our own created network by adding layers of CNN.

[2] TO TRAIN THE MODULE BY USING ALEXNET

To train our dataset using classifier and fit generator function also we make training steps per epoch's then total number of epochs, validation data and validation steps using this data we can train our dataset. Training the module using Alexnet CNN.

[3] TO TRAIN THE MODULE BY USING LENET

A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, ConvNets have the ability to learn these filters/characteristics. The architecture of a ConvNet is analogous to that of the connectivity pattern of Neurons in the Human Brain and was inspired by the organization of the Visual Cortex. Individual neurons respond to stimuli only in a restricted region of the visual field known as the Receptive Field. Their network consists of four layers with 1,024 input units, 256 units in the first hidden layer, eight units in the second hidden layer, and two output units.

i. INPUT LAYER

Input layer in CNN contain image data. Image data is represented by three dimensional matrixes. It needs to reshape it into a single column. Suppose you have image of dimension 28 x 28 =784, it need to convert it into 784 x 1 before feeding into input.

ii. CONVO LAYER

Convo layer is sometimes called feature extractor layer because features of the image are get extracted within this layer. First of all, a part of image is connected to Convo layer to perform convolution operation as we saw earlier and calculating the dot product between receptive field (it is a local region of the input image that has the same size as that of filter) and the filter. Result of the operation is single integer of the output volume. Then the filter over the next receptive field of the same input image by a Stride and do the same operation again. It will repeat the same process again and again until it goes through the whole image. The output will be the input for the next layer.

iii. POOLING LAYER

Pooling layer is used to reduce the spatial volume of input image after convolution. It is used between two convolution layers. If it applies FC after Convo layer without applying pooling or max pooling, then it will be computationally expensive. So, the max pooling is only way to reduce the spatial volume of input image. It has applied max pooling in single depth slice with Stride of 2. It can observe the 4 x 4 dimension input is reducing to 2 x 2 dimensions.

iv. FULLY CONNECTED LAYER (FC)

Fully connected layer involves weights, biases, and neurons. It connects neurons in one layer to neurons in another layer. It is used to classify images between different categories by training.

v. SOFTMAX / LOGISTIC LAYER

Softmax or Logistic layer is the last layer of CNN. It resides at the end of FC layer. Logistic is used for binary classification and softmax is for multi-classification.

vi. OUTPUT LAYER

Output layer contains the label which is in the form of one-hot encoded. Now you have a good understanding of CNN.

[4] DEPLOYING THE MODEL IN DJANGO FRAMEWORK AND PREDICTING OUTPUT

In this module the trained deep learning model is converted into hierarchical data format file (.h5 file) which is then deployed in our Django framework for providing better user interface and predicting the output whether the given signature is real or forged.

Implementation

General

Requirements are the basic constrains that are required to develop a system. Requirements are collected while designing the system. The following are the requirements that are to be discussed.

- Functional requirements
- Non-functional requirements
- Environment requirements

FUNCTIONAL REQUIREMENTS

The software requirements specification is a technical specification of requirements for the software product. It is the first step in the requirements analysis process. It lists requirements of a particular software system. The following details to follow the special libraries like tensorflow, keras, matplotlib.

NON-FUNCTIONAL REQUIREMENTS

Process of functional steps,

1. Problem define
2. Preparing data
3. Evaluating algorithm
4. Improving results
5. Prediction the result

ENVIRONMENT REQUIREMENT

Software requirements

Operating System : Windows / Linux

Simulation Tool : Anaconda with Jupyter Notebook

Hardware requirements

Processor : Pentium IV/III

Hard disk : Minimum 80 GB

RAM : Minimum 2GB

LIBRARIES REQUIRED

Tensorflow : Just to use the tensor board to compare the loss and adam curve our result data or obtained log. TensorFlow is a Python library for fast numerical computing created and released by Google. It is a foundation library that can be used to create Deep Learning models directly or by using wrapper libraries that simplify the process built on top of TensorFlow. TensorFlow is a software library or framework, designed by the Google team to implement machine learning and deep learning concepts in the easiest manner. It combines the computational algebra of optimization techniques for easy calculation of many mathematical expressions.

Keras : To pre-process the image dataset. Keras runs on top of open source machine libraries like TensorFlow, Theano or Cognitive Toolkit (CNTK). Theano is a python library used for fast numerical computation tasks. TensorFlow is the most famous symbolic math library used for creating neural networks and deep learning models. TensorFlow is very flexible and the primary benefit is distributed computing. CNTK is deep learning framework developed by Microsoft. It uses libraries such as Python, C#, C++ or standalone machine learning toolkits. Theano and TensorFlow are very powerful libraries but difficult to understand for creating neural networks. Keras is based on minimal structure that provides a clean and easy way to create deep learning models based on TensorFlow or Theano. Keras is designed to quickly define deep learning models. Well, Keras is an optimal choice for deep learning applications.

Matplotlib : To display the result of our predictive outcome. Matplotlib is one of the most popular Python packages used for data visualization. It is a cross-platform library for making 2D plots from data in arrays. Matplotlib is written in Python and makes use of NumPy, the numerical mathematics extension of Python. It provides an object-oriented API that helps in embedding plots in applications using Python GUI toolkits such as PyQt, WxPythonotTkinter. It can be used in Python and IPython shells, Jupyter notebook and web application servers also.

OS : To access the file system to read the image from the train and test directory from our machines.The OS module in Python comes with various functions that enables developers to interact with the Operating system that they are currently working on. In this article we'll be learning mainly to create and delete a directory/folder, rename a directory and even basics of file handling. Python OS module provides the facility to establish the interaction between the user and the operating system. It offers many useful OS functions that are used to perform OS-based tasks and get related information about operating system. The OS comes under Python's standard utility modules. This module offers a portable way of using operating system dependent functionality.

Performance analysis

Performance evaluation metric

An evaluation metric quantifies the performance of a predictive model. This typically involves training a model on a dataset, using the model to make predictions on a holdout dataset not used during training, then comparing the predictions to the expected values in the holdout dataset.

A classifier is only as good as the metric used to evaluate it. If you choose the wrong metric to evaluate your models, you are likely to choose a poor model, or in the worst case, be misled about the expected performance of your model. Choosing an appropriate metric is challenging generally in applied machine learning, but is particularly difficult for imbalanced classification problems. Firstly, because most of the standard metrics that are widely used assume a balanced class distribution, and because typically not all classes, and therefore, not all prediction errors, are equal for imbalanced classification.

PROPOSED SYSTEM RESULT

After training the images of the signatures using CNN, the database consisting of the signatures are tested and the result displays whether the corresponding signature is genuine or forged. Convolutional Neural Networks are a very strong and efficient algorithm that may be implemented on an embedded device. The aforementioned tests can be used to verify the algorithm's efficacy. The results of all of these tests are remarkably similar. According to algorithm tests, the training of signature datasets acquired from various angles is a critical parameter to consider. This intelligent human signature verification system will assist the people in making the most secure and efficient use of their signatures.

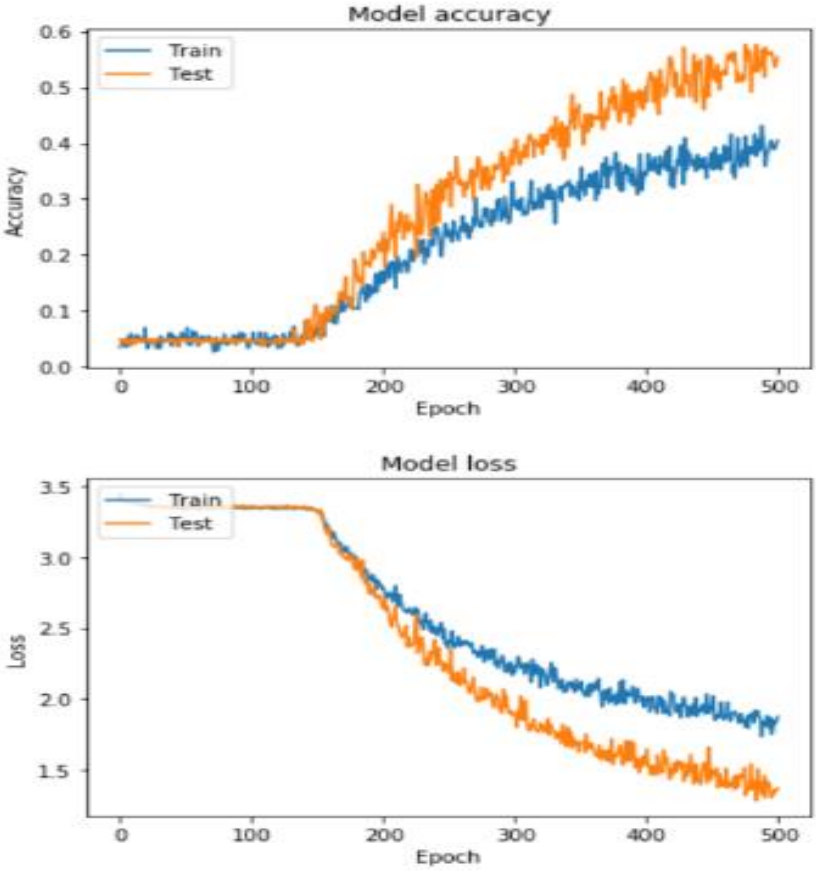

Fig 2. Proposed system graph

EXISTING SYSTEM RESULT

The samples were divided randomly into the learning set and the testing set. All forged samples were included in the testing set. Fig.5.3 shows the correct verification rate obtained through the experiments. The experiment (learning and testing) results reveal the following interesting facts about the system: All the features used for verification had almost the same degree of effectiveness whether they were used to compare the sample to prototype or vice versa. Central line features contain the largest number of points per image. Some of these points are useless because they are concentrated in the central region. Corner curve features are not highly effective in the verification process. It is quite sufficient to use the corner line features, central circle features and critical point features in the discriminating feature set.

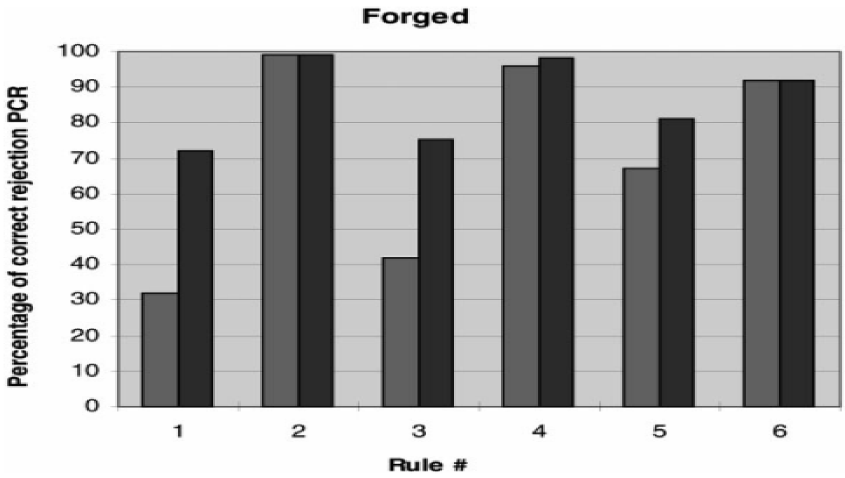

Fig. 11. The experimental results, with degree of certainty more than 85%. ■ Without threshold; ■ with threshold.

Fig 3. Existing system graph

Conclusion

Thus we have successfully developed a model using the AlexNet algorithm of one of the most powerful deep learning models, CNN (Convolutional Neural Network) to achieve an higher accuracy in terms of recognizing a signature and classifying it as to whether it is the corresponding person's original signature or a fake one. This can be useful in various sectors which involves collecting authentic information of the customer, employee or any other person. Some of the sectors include banking, database related fields, healthcare etc.

Reference

[1] Xinyu Li, Jing Xu, Xiong Fan, Yuchen Wang and Zhenfeng Zhang, "Puncturable Signatures and Applications in Proof-of-Stake Blockchain Protocols", July 16,2020; DOI 10.1109/TIFS.2020.3001738, IEEE.

[2] Jacopo Pegoraro , Francesca Meneghello , Michele Rossi , "Multiperson Continuous Tracking and Identification From mm-Wave Micro-Doppler Signatures", September 13,2020; https://www.ieee.org/publications/rights/index.html

[3] Anamika Jain1 · Satish Kumar Singh1 · Krishna Pratap Singh1, " Handwritten signature verification using shallow convolutional neural network ", April 07 ,2020 ; https://doi.org/10.1007/s11042-020-08728-6

[4] Azmi AN, Nasien D, Omar FS (2017) Biometric signature verification system based on freeman chaincode and k-nearest neighbor. Multimed Tools Appl 76(14):15341–15355

[5] Bajaj R, Chaudhury S (1997) Signature verification using multiple neural classifiers. Pattern Recogn 30(1):1–7

[6] Baltzakis H, Papamarkos N (2001) A new signature verification technique based on a two-stage neuralnetwork classifier. Eng Appl Artif Intell 14(1):95–103. https://doi.org/10.1016/S0952-1976(00)00064-6.

http://www.sciencedirect.com/science/article/pii/S0952197600000646

[7] Bouamra W, Djeddi C, Nini B, Diaz M, Siddiqi I (2018) Towards the design of an offline signature verifier based on a small number of genuine samples for training. Expert Syst Appl 107:182–195

[8] Berkay Yilmaz M, Ozturk K (2018) Hybrid user-independent and user-dependent offline signature verification with a two-channel cnn. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp 526–534

[9] Ferrer MA, Vargas JF, Morales A, Ordonez A (2012) Robustness of offline signature verification based on gray level features. IEEE Trans Inf Forensic Secur 7(3):966–977. https://doi.org/10.1109/TIFS.2012.2190281

[10] Fierrez-Aguilar J, Nanni L, Lopez-Penalba J, Ortega-Garcia J, Maltoni D (2005) An on-line signature verification system based on fusion of local and global information. In: Kanade T, Jain A, Ratha NK (eds) Audio- and video-based biometric person authentication. Springer, Berlin, pp 523–532

[11] Hadjadji B, Chibani Y, Nemmour H (2017) An efficient open system for offline handwritten signature identification based on curvelet transform and one-class principal component analysis. Neurocomputing 265:66–77. https://doi.org/10.1016/j.neucom.2017.01.108. http://www.sciencedirect.com/science/article/pii/S0925231217310159. New Trends for Pattern Recognition: Theory and Applications

[12] Ismail,M.A.;Gad,Samia(Oct2000). "Off-linearabicsignaturerecognitionandverification".PatternRecognition.**33**(10):1727–1740.