CONVOLUTIONAL NEURAL NETWORK FOR OFFLINE SIGNATURE VERIFICATION VIA MULTIPLE CLASSIFIERS

Fadi Mohammad Alsuhimat and Fatma Susilawati Mohamad

Faculty of Informatics and Computing,
Universiti Sultan ZainalAbidin, Terengganu, Malaysia

ABSTRACT

The signature process is one of the most significant processes used by organizations to preserve the security of information and protect it from unwanted penetration or access. As organizations and individuals move into the digital environment, there is an essential need for a computerized system able to distinguish between genuine and forged signatures in order to protect people's authorization and decide what permissions they have. In this paper, we used Pre-Trained CNN for extracts features from genuine and forged signatures, and three widely used classification algorithms, SVM (Support Vector Machine), NB (Naive Bayes) and KNN (k-nearest neighbors), these algorithms are compared to calculate the run time, classification error, classification loss and accuracy for test-set consist of signature images (genuine and forgery). Three classifiers have been applied using (UTSig) dataset; where run time, classification error, classification loss and accuracy were calculated for each classifier in the verification phase, the results showed that the SVM and KNN got the best accuracy (76.21), while the SVM got the best run time (0.13) result among other classifiers, therefore the SVM classifier got the best result among the other classifiers in terms of our measures.

KEYWORDS

CNN, Signature verification, SVM, KNN, NB.

1. INTRODUCTION

A handwritten signature considered as a personal skill which consists a group of symbol and characters written in a specific language, the signature is one of the operations that use to provide persons with authentication to perform many transactions, such as banking transactions and classes attendance, where the signature can ensure the permitted validity of persons and classify the forged signature from the genuine signature [1].

A signature is sketched out as an extraordinarily composed drawing that an individual composes on any record as a sign of character. A person employments it on a normal wish to sign a check, a legitimate instrument, contract, etc. The matter emerges when once some person tries to duplicate its [2].

Signature verification may be a complex design recognizable proof with inadequacy as no two veritable signatures of a person can be absolutely comparative. In case inadvertently it is winning at that point it'll do genuine damage to an individual. One of the ways is to utilize the biometric features of each person [3].

David C. Wyld et al. (Eds): NATL, NETWORKS, Fuzzy - 2021
pp. 107-116, 2021. CS & IT - CSCP 2021 DOI:10.5121/csit.2021.112008

Nowadays signature discovery and other biometric features are playing a fundamental part in nearly all the field, where mystery and security are the most concerns for all people and nations. Moreover, utilizing signature verification can offer assistance to decide the personality of people and their authorization to do a particular work [2].

A signature recognition system could be a way to confirm the signature in order to distinguish any imitation, sometime recently getting the ultimate result from verification stage, the recognition prepare comprises of a set of stages, incorporate normalization, features extraction, and classification, these three phases are exceptionally imperative to confirm signature since the transcribed signature can shift each time depending on the conduct and position of the person. [3]. Figure 1 shows different types of signatures for the same person.

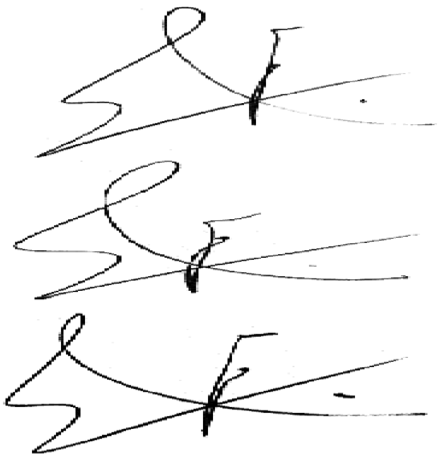


Figure 1. Example of different patterns of signature

The second stage in signature recognition system is features extraction stage, this phase considers a significant phase in signature recognition system because the whole system depends on it in order to verify individuals signatures, where this phase responsible about detecting and determine a group of features in each signature, including number of pixels, width, corner, and length [4].

The features extraction stage depends on detect image highlights with incredible precision through minimizing the measurements of the first picture at that point extricate a group of covered up characteristics within the picture, in arrange to encourage the method of separation between unique and fake marks.

The third stage in the signature recognition system is the classification stage, and this stage is the signature verification stage, in which it is determined whether the signature is false or real in it, through comparing the signature features stored in the database with anyone who wants to verify his/her signature [5].

The classification phase aims at identifying the genuine signature by comparing the enrolled and authenticated signature features. The decision-maker then chooses if the signature should be accepted or denied based on the threshold [6].

Furthermore, the signature is a character trait of individuals used in biometrics systems to verify individuals' identities, as the usage of biometric characteristics in the field of security grows, the signature appears as a biometric feature that provides a secure way of delegating individuals and verifying their identification in legal documents. Furthermore, when compared to other biometric traits like (hand geometry, iris scan, or DNA), the signature has a high level of acceptance by individuals. All these reasons have led to an increase in the proliferation of signature recognition systems and the need for further developments on these systems.

In this paper, our objective is to study the features extraction phase and classification phase for signature images. Therefore, in this research Pre-trained Convolutional Neural Network was used for features extraction phase, then signature image features are classify using (support vector machine (SVM), naive Bayes (NB) and k-nearest neighbor (KNN)), with UTSig dataset [7]. This dataset has (115) classes containing: (27) genuine signatures; (3) opposite-hand signed samples, (36) simple forgeries and (6) skill forgeries; we selected (2475) images as a training group to train the classification algorithms.

2. OVERVIEW OF METHODS

In this section, the features extraction technique and classification algorithms that are used for signature classification and comparison process are described briefly. The suggested signature classification algorithm consists of feature normalization, feature extraction and classification.

2.1. Features Extraction Phase

In this research, a deep learning method was used for offline signature verification. A Convolutional Neural Network (CNN) ad hoc model was used as a deep learning method. A Convolutional Neural Network was firstly proposed by LeCun et al [8] as a method for image processing, where it has consisted of two essential features including spatial pooling and spatially shared weights.

In 1998, they [9] enhanced the CNNs as LeNet-5 which is a pioneering 7-level convolutional network in order to classify digits. At the present time, CNNs considered the most widely utilized deep learning architecture in feature learning, through many successful applications in various areas like autonomous vehicles[10], character recognition [11], video processing [12], medical image processing and object recognition [13].

Figure (2) shows basic structure of CNN.

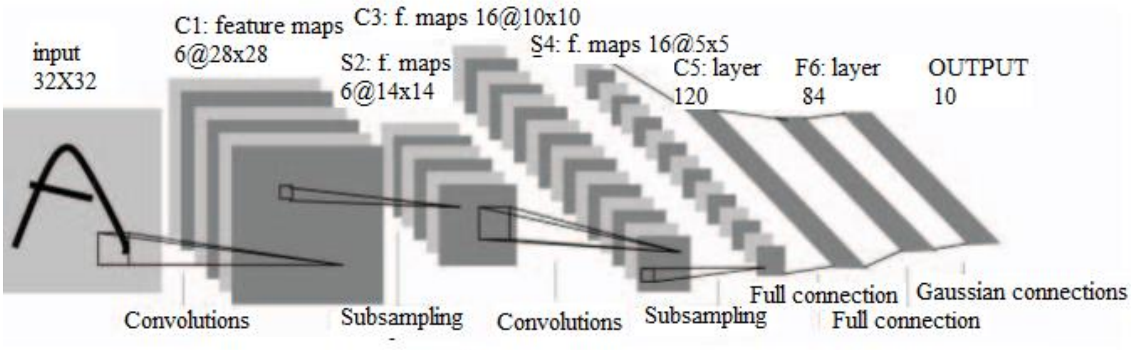


Figure 2. CNN structure

As shown in Figure (2), a CNN has three primary layers: a convolutional layer, a subsampling layer (pooling layer), and a fully-connected layer, that was taken from the study of LeCun et al

[8]. CNN points to define the unique features of pictures utilizing convolutional operations and pooling operations. The features gotten within the first layers identify as edges or colour data, whereas within the final layers they portray parts of shapes and objects [9].

In the convolution layer, the convolution operation is implemented by shifting the filter data matrix on the input data matrix and adding a bias to the multiplication of these matrixes. Basic convolution process represents in Figure. 3, Basic formulation of the convolution operation has been given in equation (1). In the equation, pixels of the output image, pixels of the input image, pixels of the filter (kernel) and bias term were represented by y, x, w and b respectively.

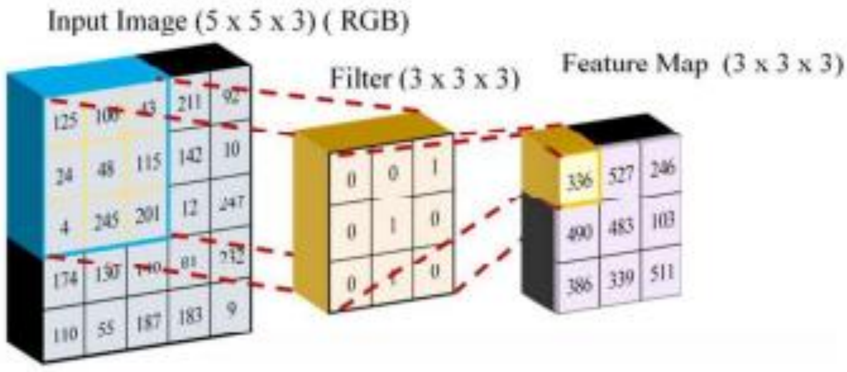


Figure 3. Basic convolution operation

$$y_n = \sum_{n=1}^{9}(x_n . w_n + b_0) \qquad (1)$$

Another tool using by CNNs is called pooling, the pooling tool [58] is utilized to spatially down-sample the activation of the previous layer by propagating the maximum activation of the previous neuron groups. The most objective of the pooling layers is diminishing the computational complexity of the model by continuously diminishing the dimensionality of the representation [9]. If preferred, a rectified linear unit (*ReLU*) activation function can be utilized at the conclusion of each layer for normalization. The main operation of (*ReLU*) was depicted in equation (2).

$$\mathrm{ReLU(x)} = f(x) = \begin{cases} 0 & if\ x < 0 \\ x & \quad if\ x \geq 0 \end{cases} \qquad (2)$$

Fully Connected Layers (FC), which are the primary building components of classical neural networks, are the final layer in CNN. Fully Connected layers are shaped by the association of neurons to each neuron within the following layer. It is at that point normalized to a probability dispersion employing a Soft-Max layer. Moreover, it points to require the high-level sifted pictures and interpret them into votes. These votes are communicated as weights, or association qualities, between each esteem and each category [9], [11].

2.2. Signature Classification

In this paper, we used various algorithms for classification: KNN, SVM, and SVR.

K-nearest Neighbor (KNN): This is a procedure of gathering parameters based on closest tests of the range of inner features [14]. KNN is one of the popular and clear classification calculations.

Learning approach as it joined sparing characteristic vectors and marks of the learning pictures, inner gathering operations.

This unmarked position may be really assigned the title for its *k* closest neighbor's. Regularly, this thing will be categorized based on the marks of its *k* closest neighbor's by utilizing overwhelming portion surveying. On *k*=1, those parameters are categorized based on the power of the parameter closest to it. If there is a need for only two segments, then k should make an odd number. *K* may be an odd number when showing up multiclass arrangement. This stage used the famous distance equation, Euclidean distance, as a related point separation capacity for KNN after changing each image to a vector from claiming fixed-length for true numbers:

$$d(x, y) = (\sum_{i=1}^{m}((x_i - y_i)^2))^{1/2} \qquad (3)$$

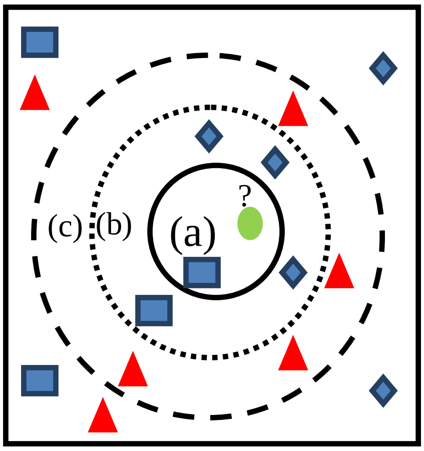


Figure 4. KNN Classification

Support Vector Machine (SVM): This is prepared to assess signature among specific signature qualities [15]. Through applying a classification algorithm to particular features for signature images, during the training procedure, we trained a signature classifier, used every last one of the preparation data. An outline of signature prediction utilized SVM algorithm indicated in Fig. 5 to classify the input signature image with training procedures. The inputs xi is the characteristic vectors. To configure the SVM parameters, we used Gaussian kernel *K*:

$$f(x) = \sum_{i=1}^{N_s} a_i y_i K(s_i, x) + b \qquad (4)$$

$$K(x_i, x_j) = e^{\frac{1}{2\sigma^2}|x_i - x_j|^2}$$

Naive Bayes: Naive Bayes learning refers to the construction of a Bayesian probabilistic model that assigns a posterior class probability to an instance*: P(Y = yj |X = xi)*. The simple naive Bayes classifier uses these probabilities to assign an instance to a class. Applying Bayes' theorem (Eq. 7) [16], and simplifying the notation a little, as shown in equation 5.

$$P(y_i|x_i) = \frac{P(x_i|y_i)P(y_i)}{P(x_i)} \qquad (5)$$

3. EXPERIMENTAL RESULT

This section shows the results of our classifiers, through three mean sections, section (3.1) describes the database which was used. Section (3.2) shows the receiver operating characteristic (ROC) and run-time for each classifier, while section (3.3) indicates the performance of each classifier by calculating (accuracy, classification error, classification loss, and run-time).

3.1. Database

The process of comparing three algorithms implemented on a set of signature images from the (UTSig) dataset. As illustrated in Figure (5), this dataset has "(115) classes containing: (27) authentic signatures; (3) opposite-hand forgeries, (36) easy forgeries, and (6) skill forgeries." Each lesson is assigned to a single actual person. UTSig contains (8280) photos taken from undergraduate and graduate students at the University of Tehran and Sharif University of Technology, where signatures images were scanned at 600 dpi and saved as 8-bit Tiff files" [7, p1].

In this paper, a total of (2475) signature images were chosen to train the set, and (660) signature images were chosen to test our classification algorithms.

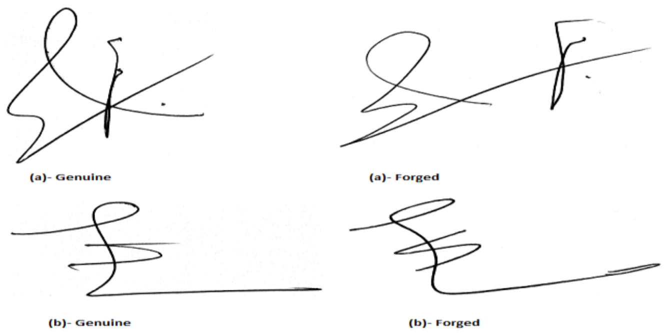


Figure 5. Forger and Genuine signature examples from UTSig dataset.

3.2. Experimental setup

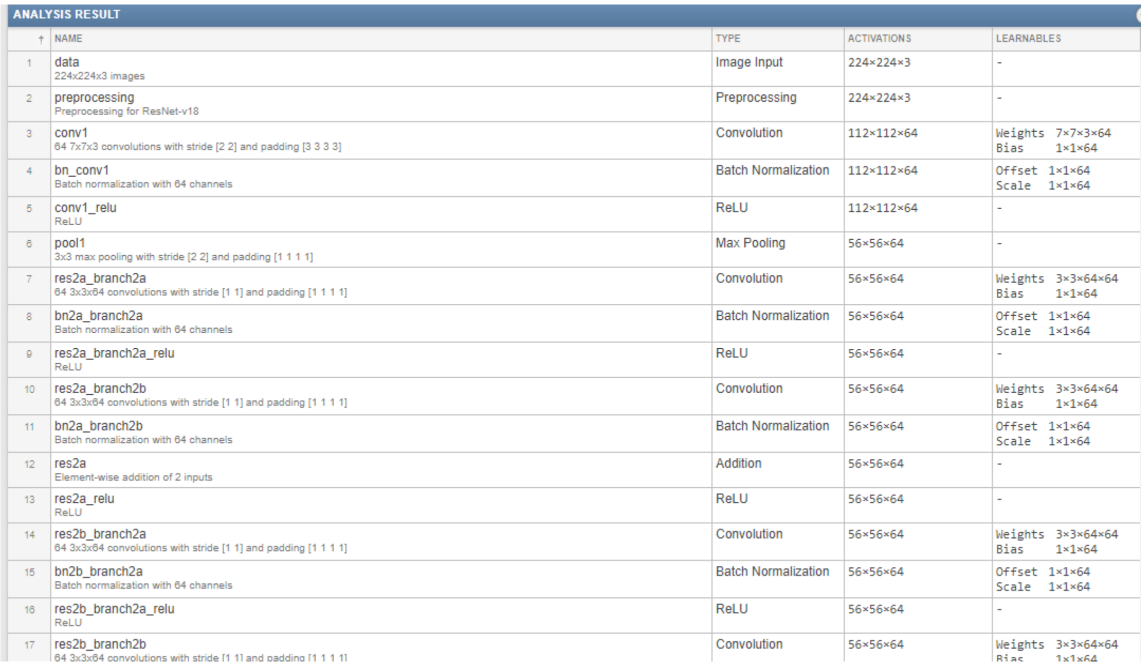
Features were extracted from a pre-trained CNN and then classified in original-forgeries through three classifiers, SVM, KNN and NB. In the first model, CNN was trained via a set of signatures for (75) persons, where each person has 33 signatures which include 27 genuine and 6 forgeries were used, the pre-trained CNN used AlexNet for features extraction process, where AlexNet uses layers property that comprises of 25 layers. There are 8 layers for learnable weights, 5 convolutional layers and 3 fully connected layers. Fig. 5 shows the details of all the layers of AlexNet.

Table I shows the experimental results using (ROC) by calculating the area under the curve for the estimated values of *X* and *Y*. Also, calculate the run-time for each classifier. We discovered that KNN performed better than other classifier algorithms, which include SVM and NB according to ROC values, where the NB classifier run-time was better than other classifier algorithms.

Table 1. Run-Time and AUC values for each classifier

Method	Run-Time	AUC
SVM	70.1	0.998
KNN	1.89	0.999
NB	1.52	0.782

Figure. 6 showed the ROC values for each classifier, where KNN produces better ROC values for higher thresholds, SVM is also got good ROD values and almost equal to KNN values. While the ROC curve for naive Bayes is often lowers than the other two ROC curves, this suggests that the other two classifier algorithms perform better in-sample.



ANALYSIS RESULT

↑	NAME	TYPE	ACTIVATIONS	LEARNABLES
1	data 224x224x3 images	Image Input	224×224×3	-
2	preprocessing Preprocessing for ResNet-v18	Preprocessing	224×224×3	-
3	conv1 64 7x7x3 convolutions with stride [2 2] and padding [3 3 3 3]	Convolution	112×112×64	Weights 7×7×3×64 Bias 1×1×64
4	bn_conv1 Batch normalization with 64 channels	Batch Normalization	112×112×64	Offset 1×1×64 Scale 1×1×64
5	conv1_relu ReLU	ReLU	112×112×64	-
6	pool1 3x3 max pooling with stride [2 2] and padding [1 1 1 1]	Max Pooling	56×56×64	-
7	res2a_branch2a 64 3x3x64 convolutions with stride [1 1] and padding [1 1 1 1]	Convolution	56×56×64	Weights 3×3×64×64 Bias 1×1×64
8	bn2a_branch2a Batch normalization with 64 channels	Batch Normalization	56×56×64	Offset 1×1×64 Scale 1×1×64
9	res2a_branch2a_relu ReLU	ReLU	56×56×64	-
10	res2a_branch2b 64 3x3x64 convolutions with stride [1 1] and padding [1 1 1 1]	Convolution	56×56×64	Weights 3×3×64×64 Bias 1×1×64
11	bn2a_branch2b Batch normalization with 64 channels	Batch Normalization	56×56×64	Offset 1×1×64 Scale 1×1×64
12	res2a Element-wise addition of 2 inputs	Addition	56×56×64	-
13	res2a_relu ReLU	ReLU	56×56×64	-
14	res2b_branch2a 64 3x3x64 convolutions with stride [1 1] and padding [1 1 1 1]	Convolution	56×56×64	Weights 3×3×64×64 Bias 1×1×64
15	bn2b_branch2a Batch normalization with 64 channels	Batch Normalization	56×56×64	Offset 1×1×64 Scale 1×1×64
16	res2b_branch2a_relu ReLU	ReLU	56×56×64	-
17	res2b_branch2b 64 3x3x64 convolutions with stride [1 1] and padding [1 1 1 1]	Convolution	56×56×64	Weights 3×3×64×64 Bias 1×1×64

Figure 5. shows the details of all the layers of AlexNet

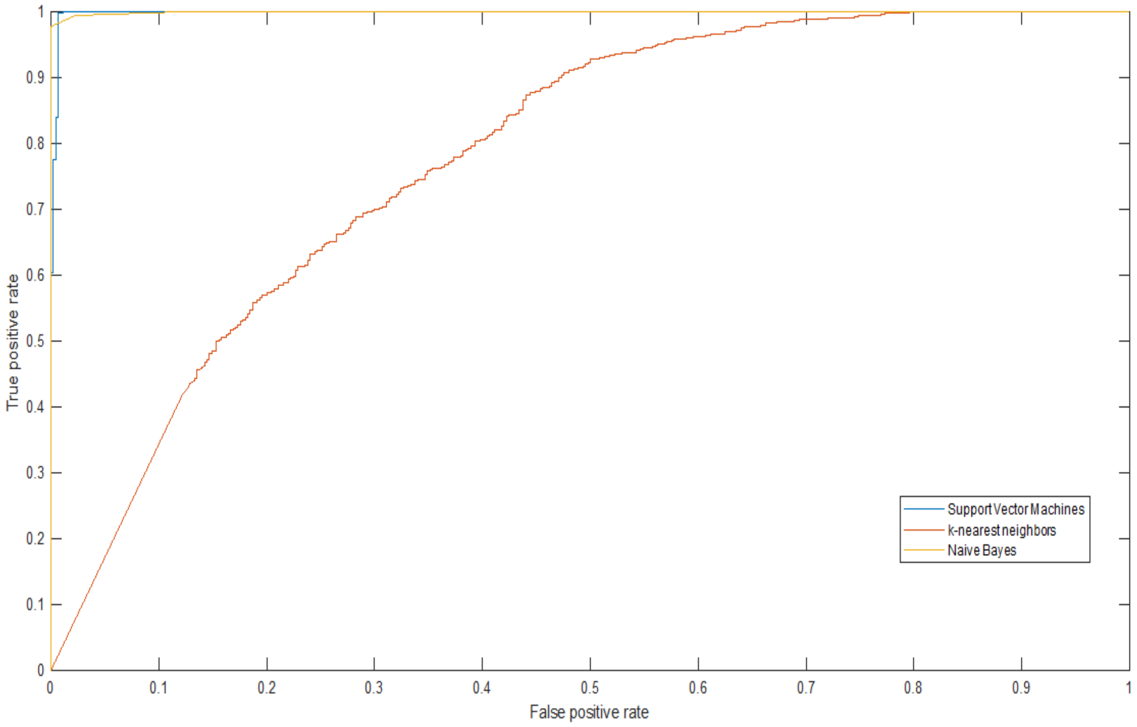


Figure 6. Receiver operating characteristic (ROC) curve

3.3. Efficiency

The efficiency was taken regarding run time, classification error, classification loss and accuracy measurements for each classifier on 660 images sequentially.

Table 2. shows all measures for each classification algorithms

Measures	Methods		
	SVM	KNN	NB
Run-Time	0.13	0.77	0.18
Classification Error	0.24	0.24	0.29
Classification Loss	0.01	0.01	0.26
Accuracy	76.21	76.21	71.36

Data in the above table showed that, for the run time we can note that the best run time was for SVM classifier. Following by NB classifier, and finally KNN classifier, while for classification error we note that, SVM and KNN misclassifies approximately (24%) of the test sample, while NB misclassifies approximately (29%) of the test sample. Besides that, classification loss values indicated that, SVM and KNN classifiers have better value (0.01) than NB classifier (0.26), finally the accuracy value for both classifier SVM and KNN achieved (76.21) which better than the accuracy value for NB classifier.

4. Conclusion and Future Work

In this research, the SVM, KNN, and NB classification algorithms were compared on a set of signature images from the (UTISG) dataset to assess performance by calculating the run time, classification error, classification loss, and accuracy metrics for each algorithm. The three methods described here are popular classification algorithms, with computing complexity and accuracy being the most important factors in selecting a better classification technique.

The comparison process is done between the train set consist of (2475) signature images through pre-trained CNN for features extraction, then the result trained using three classifiers SVM, KNN and NB. After that the run time, classification error, classification loss and accuracy measurements calculated for each algorithm in order to find the best classification algorithm. The experimental results showed that, the best run time was for SVM classifier, following by NB classifier, and finally KNN classifier, while for classification error SVM and KNN got same misclassifies approximately and better than NB misclassifies approximately. In addition, SVM and KNN classifiers have same classification loss values and better than NB classifier, finally the accuracy value for both classifier SVM and KNN was same and better than the accuracy value for NB classifier.

For future work other classification algorithms will be test with the same and different dataset, also using full deep learning system for both phases (extract features and classification) will help in build an accurate signature verification system.

References

[1] F. Alsuhimat, F. S. Mohamad, and M. Iqtait, “Detection and Extraction Features for Signatures Images via Different Techniques,” IOP Conf. Series: Journal of Physics: Conf. Series 1179 (2019) 012087.

[2] F. S. Mohamad, F. Alsuhimat, M. Mohamed, M. Mohamad, and A. Jamal, “Detection and Feature Extraction for Images Signatures,” International Journal of Engineering & Technology, vol. 7, no. 3, pp. 44-48, 2018.

[3] J. Poddar, V. Parikh, S. K. Bharti, “Offline signature Recognition and Forgery Detection using Deep Learning,” The 3rd International Conference on Emerging Data and Industry 4.0 (ED140), April 6-9, Warsw, Poland, 2020.

[4] K. Daqrouq, H. Sweidan, A. Balamesh, and M. Ajour, "Off-Line Handwritten Signature Recognition by Wavelet Entropy and Neural Network", Entropy., vol. 19, no. 6, pp. 1.20, 2017.

[5] T. Jahan., S. Anwar, and A. Al-Mamun, "A Study on Preprocessing and Feature Extraction in offline Handwritten Signatures", Global Journal of Computer Science and Technology: F Graphics & Vision., vol. 15, no. 2, pp. 1.7, 2015.

[6] S. Gunjal, B. Dange, and A. Brahmane, "Offline Signature Verification using Feature Point Extraction", International Journal of Computer Applications., vol. 141, no. 14, pp. 6.12, 2016.

[7] Soleimani, K. Fouladi, and B. Araabi, "UTSig: A Persian offline signature dataset", IET Biometrics., vol. 6, no. 1, pp. 1.8, 2016.

[8] S. Singh, M. Gogate, and S. Jagdale, “Signature Verification Using LDP & LBP with SVM Classifiers,” International Journal of Scientific Engineering and Science, vol. 1, no. 11, pp. 95-98, 2017.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ Images classification with deep convolutional neural networks,” In Advance in neural information processing systems, pp. 1097-1105, 2012.

[10] Y. LeCun et al.,”Handwritten digit recognition with a back-propagation network,” in Advances in neural information processing systems, pp. 396-404, 1990.

[11] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffiner, ‘”Gradient-based learning applied to document recognition,” Proc. IEEE, vol. 86, no. 11, pp. 2278-2324, 1998.

[12] M. F. Yahya and M. R. Arshad, “Detection of markers using deep learning for docking of autonomous underwater vehicle,” in 2017 IEEE 2nd International Conference on Automatic Control

and Intelligent Systems (I2CACIS), pp. 179 184, 2017.
[13] C. Boufenar, A. Kerboua, and M. Batouche, "Investigation on deep learning for off-line handwritten Arabic character recognition," Cogn. Syst. Res., 2017.

[14] R. Olmos, S. Tabik, and F. Herrera, "Automatic handgun detection alarm in videos using deep learning," Neurocomputing, vol. 275, pp. 66-72, 2018.

[15] V. D. Nguyen, H. Van Nguyen, D. T. Tran, S. J. Lee, and J. W. Jeon, "Learning Framework for Robust Obstacle Detection, Recognition, and Tracking," IEEE Trans IntellTranspSyst, vol. 18, no. 6, pp. 1633-1646, 2017.

[16] T. Cover, and P. Hart, "Nearest neighbor pattern classification," IEEE transactions on information theory, vol. 13, no. 1, pp. 21-27, 1967.

[17] C. J. Burges, " A tutorial on support vector machines for pattern recognition," Data mining and knowledge discovery, vol. 2, no. 2, pp. 121-167, 1998.

[18] D. Berrar, "Bayes theorem and naive Bayes classifier," Encyclopedia of Bioinformatics and Computational Biology, vol. 1, pp. 403-412, 2018.

AUTHORS

Received the B.S. degree in computer information system from Alhussien Bin Talal University, Ma'an, Jordan, in 2007, the M.S. degree in computer science from Utara University Malaysia (UUM), Kedah, Malaysia, and now Ph.D. student in pattern recognition, deep learning at University Sultan ZainalAbidin, Kuala Terengganu (UNISZA), interesting in machine and deep learning, data science and artificial intelligence.

B.Sc degree in information system management from Oklahoma, USA, master degree in computer science from University Kebangsaan Malaysia, and Ph.D in computer science from University Teknologi Malaysia, now work as Associate Professor at Faculty of Informatics and Computing, University Sultan ZainalAbidin, Kuala Terengganu, Malaysia. Current research on Statistical and Biometric Pattern Recognition.