Nexo Revista Científica
Vol. 36, No. 06, pp. 1020-1037/Diciembre 2023

ISSN-E 1995-9516
Universidad Nacional de Ingeniería
http://revistas.uni.edu.ni/index.php/Nexo
https://doi.org/10.5377/nexo.v36i06.17458

Design a 1D-CNN'S to classify surface EMG (SEMG) signals

Diseño de un 1D-CNN para clasificar señales EMG de superficie (SEMG)

Musaab Saleh Dawood *, Mohand Lokman Aldabag

Computer Techniques and Engineering Dept. Engineering, Technical college, Northern Technical University, Mosul, Iraq.
* musaab.dawood@ntu.edu.iq

(recibido/received: 30-julio-2023; aceptado/accepted: 25-octubre-2023)

ABSTRACT

Amputate forearm, finger, or hand is a biggest problem for the disabled subject. Therefore, Prosthetic does a significant role for the amputees to modify the capability and mobility of their systematic activities. Using the EMG signals of hand and finger motion discrimination are continuously growth for numerous hand and finger gestures. The main problem in designing a prosthetic hand is the classification of EMG signals. Machine learning (ML) algorithms present a solution to this problem by providing a way to classify EMG signals with simply and less costly scheme. This study presents more than one experiment on two datasets in order to classify individual fingers (IF) with wrist and victory based on a normative dataset of EMG signals and Deep Learning DL. These experiments show that the overall performance (average accuracy) of the proposed method is 98.83% and the overall error classification rate (error rate) is 1.17%.

Keywords: electromyography EMG; Wrist movement classification, Victory movement classification, Individual Finger movements classification, 1D-CNN.

RESUMEN

Amputar el antebrazo, el dedo o la mano es el mayor problema para el sujeto discapacitado. Por lo tanto, Prosthetic desempeña un papel importante para que los amputados modifiquen la capacidad y movilidad de sus actividades sistemáticas. El uso de señales EMG de discriminación del movimiento de manos y dedos aumenta continuamente para numerosos gestos de manos y dedos. El principal problema al diseñar una prótesis de mano es la clasificación de las señales EMG. Los algoritmos de aprendizaje automático (ML) presentan una solución a este problema al proporcionar una forma de clasificar las señales EMG con un esquema simple y menos costoso. Este estudio presenta más de un experimento en dos conjuntos de datos para clasificar dedos individuales (IF) con muñeca y victoria en función de un conjunto de datos normativos de señales EMG y aprendizaje profundo DL. Estos experimentos muestran que el rendimiento general (precisión promedio) del método propuesto es del 98,83% y la tasa general de clasificación de errores (tasa de error) es del 1,17%.

Palabras claves: electromiografía EMG; Clasificación de movimientos de muñeca, Clasificación de movimientos de victoria, Clasificación de movimientos individuales de los dedos, 1D-CNN.

1. INTRODUCTION

The human disability represented by losing one of the limbs is a biggest problem to the restoration of the disabled subject, especially when the forearm, finger, or hand is amputated (Andrews et al., 2008). Gesture recognition built on electromyographic (EMG) signals is a favorable method for the development of Human-Machine Interfaces (HMIs) with standard control, such as intuitive robot interfaces. The biggest job in designing a prosthetic hand is the classification of EMG signals made by neurons of the arm to distinguish different hand actions. These EMG signals vary in strength from one to one and from movement to movement (Bhatti, 2019).

The using of EMG signals of hand and finger motion discrimination are continuously progressing. Though, little studies have interested on the on individual finger (IF) with wrist and victory gestures, which are considered more interesting than whole other gestures (wrist and whole-hand gestures) to classify because of the complexity of the muscles which responsible for IF movements. Most models of finger gesture classification use a large number of channels to measure the EMG signals, which means that the system (model) is complex, and its cost is high. As a result, classifying wrist and hand movements had been interested in the numerous previous studies. As a result of the current advances in computing world, especially, machine learning (ML) algorithms have reduced the number of channels to classify the IF gestures without losing the accuracy of classification and the response time (Lee et al., 2021).

In current years, DL methods have been effectively applied to a wide range of IF movements recognition (Tsinganos et al., 2018). As a result, new studies in biomedical engineering are focused on towards the application of these methods to EMG-based IF gesture recognition.

1.1. Literature Review

Many studies and researches have been made in the field of sEMG signals classification and processing for individual fingers movements due to its non-invasive nature and availability. P. Tsinganos et al., presented a brief overview of Deep Learning methods along with an analysis of a modified simple model based on CNN's for EMG-based hand gesture recognition. 3% improvement yields on the classification accuracy of the basic model of the proposed network, whereas the analysis helps in exploring new ways to improve performance and understanding the limitations of the presented model (Tsinganos et al., 2018). While (Atzori et al., 2016), a simple CNN architecture based on 5 blocks of convolutional and pooling layers is used to classify a large number of gestures. The classification accuracy is comparable to those obtained with classical methods, but the classification accuracy is relatively low than the best performance achieved on the same problem using an RF classifier. Stephenson et al., the Myo armband is used as an image input into a CNN to indicate how neural networks have been useful to sEMG signal classification i.e. Long Short Term Memory (LSTM) and convolutional neural networks (CNN) networks. The achieved classification on five finger flexion actions in addition to seven gestures which included four combinations of flexion movements involving the thumb and one of the other fingers i.e. index (Stephenson et al., 2018). (Xing et al., 2018) a CNN model is proposed with five parallel convolution layers to remove the disadvantage of conventional classification methods, which is losing valuable information during feature extraction, and rise EMG-based hand gesture accuracy. For this, data from the NinaPro database was used which has EMG data relative to 53 hand movements for 78 subjects divided into three datasets. They exposed that the classification accuracy of the CNN-based method was 83.23% with preprocessing. (Chen et al., 2020) presented CNN's structure for reducing the deep learning hyper parameters and increasing classification accuracy. The CNN model had four convolutional layers and one max pooling layer. MYO armband sensor is used to measure seven hand gestures. As a result, the proposed training CNN model improves the classification accuracy of hand gestures recognition. (Guo et al., 2020), presented Lw-CNN and SVM to classify six various upper-limb movements of eight subjects as a controlled robotic arm in online. Lw-CNN classification is better than SVM in online experiments. Both classifiers have better accuracy offline than online. 88.75% is the average accuracy of the presented study. (Akmal et al., 2021) Support vector machine (SVM) is applied to classify the finger movements at real-time. The proposed classifier achieved mean classification accuracy equal to

78%. The main contribution of this study can be summarized in three points is designing a 1D-CNN layers to classify individual finger movements.

2. FEATURE EXTRACTION

Features extraction refers to raw signal data transformation into feature maps by eliminating noise, extracting the important data (Chang et al., 2022), and applying it as inputs to the classification system (Kundu & Naidu, 2021). So, Features extraction is a significant procedure for classifying systems by extracting useful information hidden in the raw signals, which should contain as much information as possible to gesture classification. Features extraction includes taking a short time windows of the EMG signal to obtain a more informative measure.

This feature extraction should be completed to recognize the command signal (Choi, 2023). In general, time TD, or frequency FD or time–frequency domain TFD is adopted to extract the features. Because of TD is fast, easy to implement, representing the transient state of gestures (Lee et al., 2021) and short enough response time which is suitable in real-time recognition (Caceres, 2014). Therefore, TD features is adopted for IF recognition. For this study, the following seven features are used:

2.1. Max

Max is the maximum amplitude value of the input signal that used in non-stationary signal (Caceres, 2014).

2.2. Min

Min is the minimum amplitude (McIntosh et al., 2016).

2.3. Standard deviation (SD)

SD represents the difference between each signal sample and its mean value (Too et al., 2017).

$$\sigma = \frac{1}{N}\sum_{i=0}^{N}(x_i - \bar{x})^2 \tag{2.1}$$

SD represents noise and other interference. It is used in comparison to the mean (Too et al., 2017).

2.4. RMS

RMS is a root mean square of the signal (Venugopal & Ramakrishnan, 2014). It can be expressed as:

$$RMS = \sqrt{\frac{1}{N}\sum_{i=0}^{N} {x_i}^2} \tag{2.2}$$

2.5. MAV

MAV is the total average of the absolute value of the signal (Venugopal & Ramakrishnan, 2014). It can be expressed mathematically, as:

$$MAV = \frac{1}{N}\sum_{i=0}^{N}|x_i| \tag{2.3}$$

2.6. AAC

AAC is the average amplitude change of the waveform (Chang et al., 2022), the mathematical expression is:

$$AAC = \frac{1}{N}\sum_{i=0}^{N-1}|x_{i+1} - x_i| \quad (2.4)$$

2.7. Amplitude First Burst (AFB)

To calculate AFB, the signal is squared. The first maximum point of the resultant signal is used as the feature (BAKIRCIOĞLU & Özkurt, 2020).

3. CONVOLUTION NEURAL NETWORKS (CNN)

CNN is a deep learning multi-layers artificial neural network that is often used in large data sets. CNN structure consists of a set of layers that can extract numerous features of input data (e.g. image). Higher level features are extracted in the initial layers. While the lowest level is extracted in the deeper layers. The basic structure of CNN is shown in figure 2.7 (Sadhu, 2019).

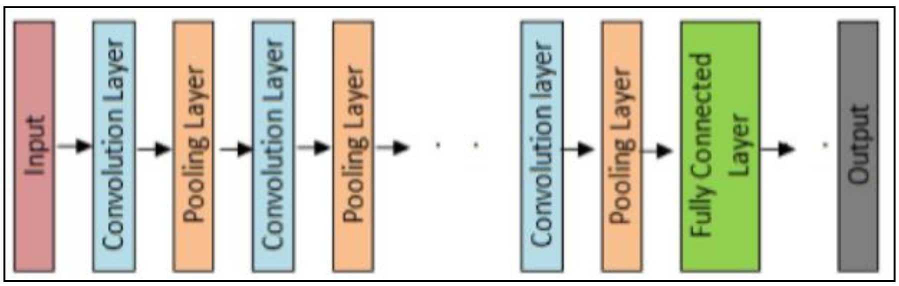

Figure 1. Conceptual model of CNN.

Basically, in CNN's structure, the standard neural network is used in the classification process, while a set of layers different from the conventional neural networks known as hidden layers are used to extract the signal features (Gadekallu et al., 2021).

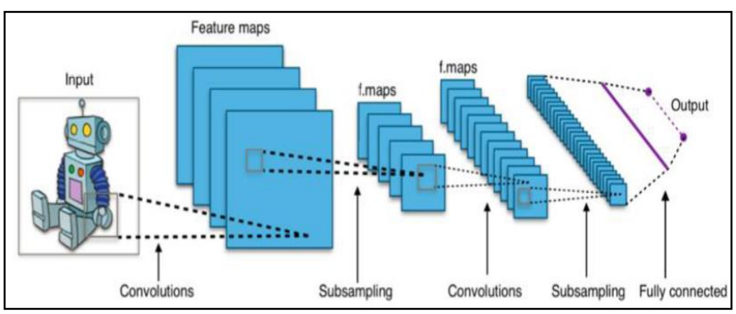

Figure 2. Convolutional and Non-Convolution layers (Bhatti, 2019).

Feature extraction is done by specifying or choosing the size filter in each convolution layer. Depending upon application, CNN's have multiple hidden layers for convoluting the input with the filter. The non-convolution is other layers in CNN's (Bhatti, 2019). CNNs convolute the input data with filter(s) to extract features from them. The convolution and non-convolution layers are shown in figure 2.

3.1. Pooling layer

Pooling-layer decreases the spatial dimensionality of the input matrix. The pooling is classified into average and max pooling layer. Max pool layer deals with peak value of each submatrix in features map. The average pool layer takes average value of each submatrix in feature map (Kiranyaz et al., 2021).

3.2. Fully connected FC-layer

After pooling layer, feed-forward ANN procedures are done using the fully-connected (FC) layer as classifier layer (Amado Laezza, 2018). It is a conventional neural network.
As mentioned above, the last pooling or convolutional layer in the form of flattened vector is the input to FC layer. So, the FC layer output refers to type class for input signal (Ghosh et al., 2020). The smaller datasets lead to CNN overfitting. To solve the overfitting problem, Dropout and Batch normalization layers are added to overcome this problem in CNN's. To modify CNN's, numerous aspects are adopted such as activation, loss and optimization functions (Alzubaidi et al., 2021).

4. ACTIVATION FUNCTIONS

Activation functions are computational functions that are used to activate NN performance. It executes complex calculations in the hidden layers and sends them to the output layer. The activation functions generate nonlinear features in hidden layers. The activation functions can activate or deactivate each neuron in the hidden layer. Thus, the output of each neuron is within range 1to 0 or – 1 to +1. The types of the activation functions, are Sigmoid, Tanh, ReLU, Leaky ReLU, Parametric ReLU, Softmax and Swish activation function (Kiranyaz et al., 2021).

4.1. Sigmoid Function (Sigm)

It receives real numbers, their output range between zero and one. Its mathematical expression is:

$$Sigm = \frac{1}{[e^{-x}+1]} \quad (1)$$

Figure 3, shows sigmoid function.

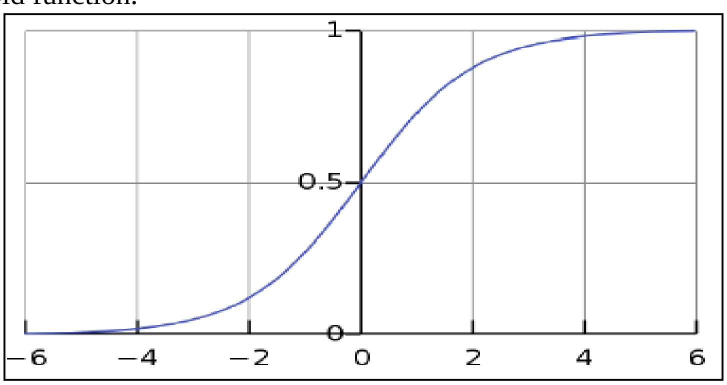

Figure 3. Sigmoid function.

4.2. Tanh Function

Its input is real numbers, but the output range between − 1 and 1, see figure 4. Its mathematical representation is:

$$Tanh = \frac{[e^{x}-e^{-x}]}{[e^{-x}+e^{x}]} \quad (2)$$

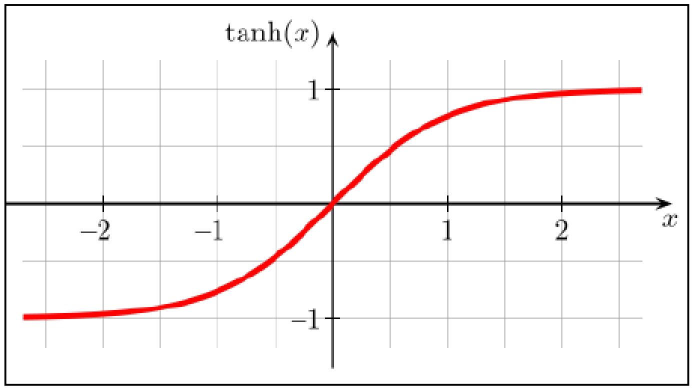

Figure 4. Tanh function.

4.3. ReLU activation function

It is mostly function used in the hidden layers of CNN's to convert input value to positive number, see fig 2.11. Its mathematical expression is:

$$ReLU = max(0, x) \tag{3}$$

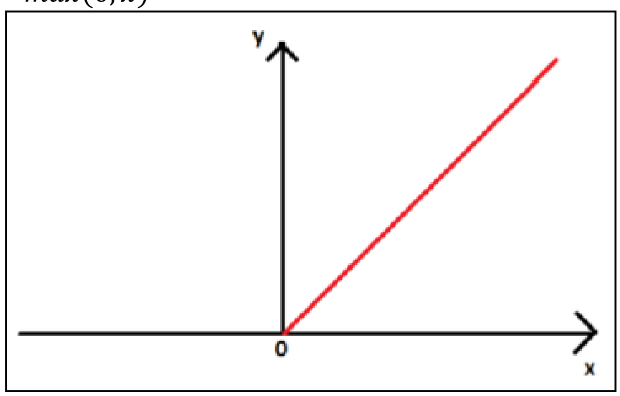

Figure 5. ReLU function.

ReLU is favored for CNN's because of unlike sigmoid and tanh activation functions, ReLU always has a constant derivative which is very small in the saturation region while tanh and sigmoid don't (Bhatti, 2019).

4.4. Softmax

It is used in the FC layer of CNN's (Bhatti, 2019). It turns numbers into probabilities that sum to 1, see figure 6 (Alzubaidi et al., 2021; Ghosh et al., 2020).

Its mathematical expression is:

$$Softmax = \frac{e^{x_i}}{\sum_{i=1}^{N} e^{x_i}} \tag{4}$$

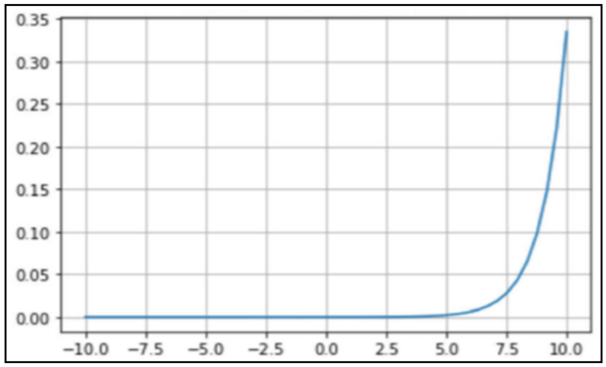

Figure 6. Softmax function.

5. EVALUATION METHODS

Several evaluation techniques are used in ML to evaluate the performance of the classification patterns like confusion matrix.

5.1. Confusion Matrix

The confusion matrix examines the performance of a classification pattern based ML, see (Figure 7). Also, Figure 8 shows a confusion matrix of multi classes.
It is the summary of the performance prediction and also known as the error matrix (Karimi, 2021).
It shows many correct and incorrect predictions per class to understand the confused class by the model as other class.
The confusion matrix rows denote to the predicted values while the columns denote to the actual values.
There are four possiblities:

- True Positive TP cell means that the actual and the predicted value are positive.
- True Negative TN cell means that the actual value is positive while the predicted value is negative.
- False Positive FP cell means that the actual value is negative while the predicted value is positive.
- False Negative FN cell means that the predicted and actual value are negative (Kulkarni & Batarseh, 2021).

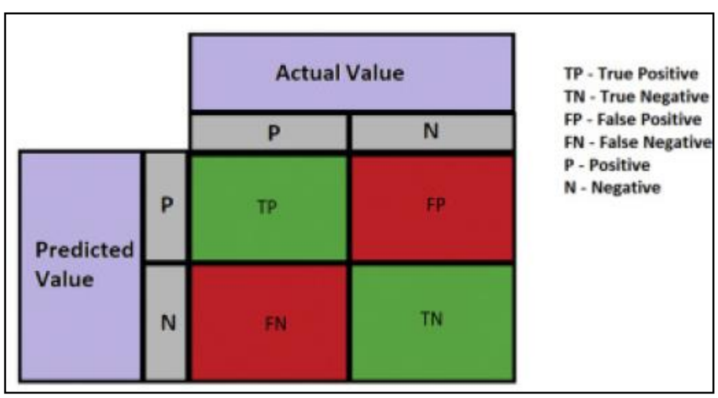

Figure 7. Confusion matrix for two classes.

Gesture	Thumb	Index	Ring	Little	Rest	Accuracy Error
Thumb	40	0	0	0	0	100% 0
Index	3	20	0	5	4	70% 30%
Ring	0	0	20	0	1	97.5% 2.5%
Little	0	6	4	30	0	75% 25%
Rest	11	11	0	10	8	20% 80%
Accuracy Error	74.07% 25.96%	62.22% 37.78%	90.7% 9.3%	66.66% 33.34%	61.54% 34.46%	72.5% 27.5%

Figure 8. Confusion matrix for multi class.

5.2. Precision

Precision aims to quantify the proportion of TP to the actual value.
It is defined as,

$$Precision = \frac{TP}{FP+TP} = \frac{T_P}{Actual\ value} \tag{5}$$

5.3. Recall (Sensitivity)

Sensitivity measures the proportion of TP to the predicted value. For each class, it is defined as,

$$Recall = \frac{TP}{FN+TP} = \frac{TP}{The\ predicted\ value} \tag{6}$$

5.4. F1-Score

F-score determines if the model trade-off between the precision and the recall performance is reasonable. The better score is 1.0 and the worst score is 0.
F1 score is expressed as

$$\mathrm{F1-Score} = \frac{2*\mathrm{Recall}*\mathrm{Precision}}{(\mathrm{Recall}+\mathrm{Precision})} \tag{7}$$

5.5. Accuracy

Accuracy is the ability of the model to determine the correct classification.
Classification accuracy is calculated as:

$$Acc = \frac{(TP+TN)}{(TP+TN+FP+FN)} \tag{8}$$

5.6. Misclassification

$$Misclassification = 1 - ACC \tag{9}$$

(Karimi, 2021; Kulkarni & Batarseh, 2021; Malvuccio & Kamavuako, 2022)

6. MATERIALS and METHODS

The main support of this study is an investigation into accurately discriminating among individual finger, wrist and victory movements using EMG signals.
Deep learning is adopted for classifying EMG signals to predict individual finger flexions. This system tries to make machine mimic human movements via EMG signals.

7. PROPOSED METHOD

The proposed method consists of two stages; feature extraction and classification (CNN) stages. Figure 9 illustrates the proposed method architecture.
The first stage uses MAV, RMS, MIN, MAX, AAC, SD and AFB feature extraction methods that are applied to each window with a length equal to 150 msec of the original signal. The selected features are fed to the CNN model as an input for classifying targeted movements.

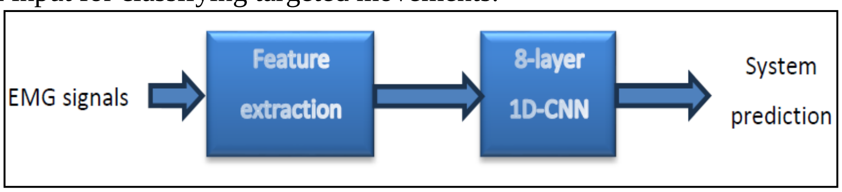

Figure 9. Block diagram of the proposed method.

This model is trained in Python and ran it in a windows-based fourth generation, core i7 PC with 1.8 GHz CPU and 8 GB memory.
Offline Dataset
This dataset is downloaded from Kaggle site which is one of the datasets provided by Google. It contains signals of seven movements (the victory, beside wrist and five individual finger flexions) of eight subjects as in figure 10.

Figure 10. Studied movements for classification.

The information about this dataset illustrated in table 4.1. The 448 dataset divided into two subsets, i.e. training and testing data. The training subset contains 399 sample of each movement and the rest (49) data are used in the testing subsets for eight subjects.

Table 1. The offline dataset.

Specification	Dataset
Number of subjects	8
Number of classes	7
Number of class repetition/subject	56
Total number of repetitions/class	448
Time for each repetition	5 sec
Total repetitions	3136
Sampling rate	200Hz
Window length	150 ms

Each movement has a binary code, table 2, shows the binary code of the class which is inserted into the matrix.

Table 2 The binary class code.

Class	movements	Binary code
1	Little	1000000
2	Index	0100000
3	Middle	0010000
4	Ring	0001000
5	Thumb	0000100
6	Wrist	0000010
7	Victory	0000001

8. THE OFFLINE EXPERIMENTS

In this thesis, a deep learning model was proposed to automate the process of feature extraction, selection, and reduction, in addition to the pattern recognition and classification of selected signals.
As mentioned above, 8-layer convolutional neural networks are used. The figure 11 shows the essential structure of the proposed CNN layers.

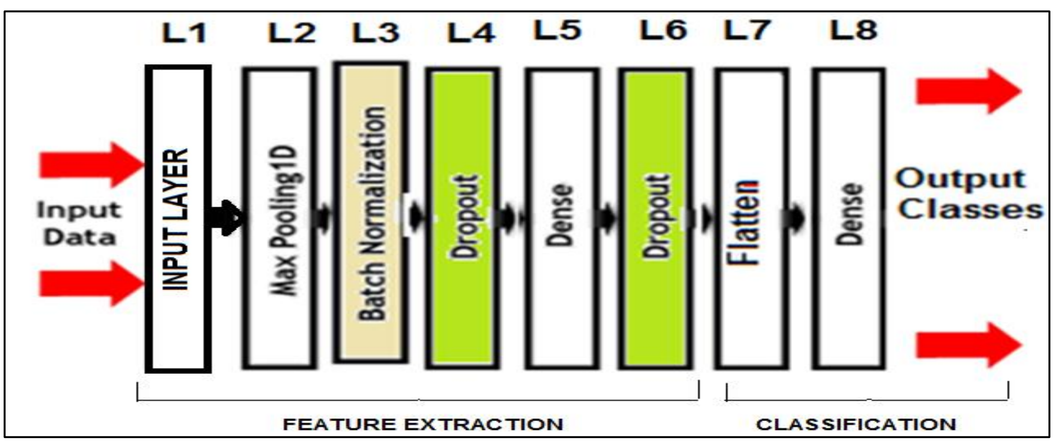

Figure 11. The proposed model block diagram.

Numerous experiments are processed offline. In the first experiment, the filter size is fixed at a value of 4, the window size of the max pooling layer is 2*2, the dropout layers1 and 2 at 0.1 and the patch size is 30.

The activation function of the dense1 layer is ReLU and the activation function of the dense2 layer is Softmax. To find the suitable number of nodes, it is changed from 220 to 160 nodes and the experiment results are in table 3.

Table 3. First experiment results.

EXP.	Dense1	ACC%	Test ACC%
1	220	98	98
2	210	97	95
3	200	98	98
4	190	99	99
5	180	98	99
6	170	97	96
7	160	95	83

From the results of experiment one the best result is at the number of nodes equal to 180. The second experiment adopted the best result (at 180 nodes) and by fixing all previous parameter and changing the filter size. The experiment results are in table 4.

Table 4 Second experiment results.

EXP.	Filter size	ACC%	Test ACC%
1	8	Overfitting	
2	6	98	90
3	4	98	99
4	2	97	89

From the results of experiment two the best result is at the filter size equal to 4. The third experiment adopted the best result (at 180 nodes) and filter size equal to 4 and by fixing all other previous parameter and changing the window size of the maxpooling layer. The experiment results are in table 5.

Table 5. Third experiment results.

EXP.	Maxpooling	ACC%	Test ACC%
1	1	98	90
2	2	98	99
3	4	97	92
4	6	Error	

From the results of the experiment three, the best result is at the window size equals to 2. The fourth experiment adopted the best result (at 180 nodes, filter size 4 and maxpooling dimensions are 2*2) and by fixing all other previous parameter and changing the rate of dropout2 layer. The experiment results are in table 6.

Table 6 Fourth experiment results.

EXP.	Rate of dropout 2 layer	ACC%	Test ACC%
1	0.2	95	97
2	0.15	95	97
3	0.1	98	99

From the results of experiment four, the best rate equals to 0.1. The fifth experiment adopted the best result (at 180 nodes, filter size 4, maxpooling dimensions are 2*2 and rate of dropout is 0.1) and by fixing rest previous parameter and changing the patch size. The experiment results are in table 7.

Table 7 Fifth experiment results.

EXP.	Patch size	ACC%	Test ACC%
1	10	98	94
2	20	95	96
3	30	98	99
4	40	98	98

From the results of the experiment five the best patch size equals to 30. The sixth experiment adopted the best result (at 180 nodes, filter size 4 and maxpooling dimensions are 2*2, rate of dropout is 0.1and the patch size is 30) and by changing the activation functions of dense1 and dense2 layers, the experiment results are in table 8.

Table 8. The sixth experiment results.

EXP.	**Activation function of dense1 layer**	**Activation function of dense2 layer**	**ACC%**	**Test ACC%**
1	ReLU	sigmoid	99	97
2	tanh	softmax	86	77
3	ReLU	softmax	98	99

From the results of the experiment six the best choice for the activation functions are in the third row of the table 8.. According to the previous experiments (experiment1 to experiment 6), the best structure of CNN's is specified in table 9.

Table 9. The best proposed model.

no.	Layer	Filter size	Stride	Method	activation function	No. node	Optimizer
1	Conv	4	1	-	-	-	'
2	Pooling	-	-	Max	-	-	'
3	Batch	-	-	-	-	-	-
4	Dropout						
5	dense				ReLU	180	Adam
6	dense				Softmax	7	Adam

9. OFFLINE EXPERIMENT RESULTS

In this section, the setting of CNN parameters will be explained through implementing various experiments. Offline experiment trains the proposed method with parameters (filter size, activation functions, optimizer type, etc.). The performance results are divided into two types: training and testing.

9.1. The training performance results

In CNN training, the recognizing of the flexion movements for wrist and each finger except thumb and victory movements is 100%. While, the recognizing of the thumb flexion and victory movements are 75% and 98%, respectively. It is clear that the confusion of each movement except thumb and victory is 0%, while the confusion of the thumb flexion and victory movements are 25% and 2%, respectively, as shown in the table 10.

Table 10. Confusion matrix of classifying the studied movements during offline training.

Signal movement	True classification						
	Little	Index	Middle	Ring	Thumb	Wrist	Victory
Little	399	0	0	0	0	0	0
Index	0	399	0	0	0	0	0
Middle	0	0	399	0	0	0	0
Ring	0	0	0	399	0	0	0
Thumb	0	0	0	0	299	100	0
Wrist	0	0	0	0	0	399	0
Victory	0	0	0	0	0	8	391

The results of the performance evaluation of the classification offline training shown in table 11, besides, figures 11 and 13 respectively. It is noticed, that the minimum accuracy and precision of classification at wrist flexion. While, the thumb flexion gives minimum sensitivity of classification. The minimum F1-score is done at the victory gesture. The maximum misclassification results at wrist and thumb movements, while the rest of the studied movements the classification is 100%.

Table 11. Performance evaluation of offline training.

Signal movement	Acc %	***Precision*** %	***Recall*** %	Mis classification%	***F1 score*** %
Little	100	100	100	0	100
Index	100	100	100	0	100
Middle	100	100	100	0	100
Ring	100	100	100	0	100
Thumb	96.4	100	74.9	3.6	85.6
Wrist	96.1	78.7	100	3.87	88.08
Victory	99.7	98	98	0.286	49

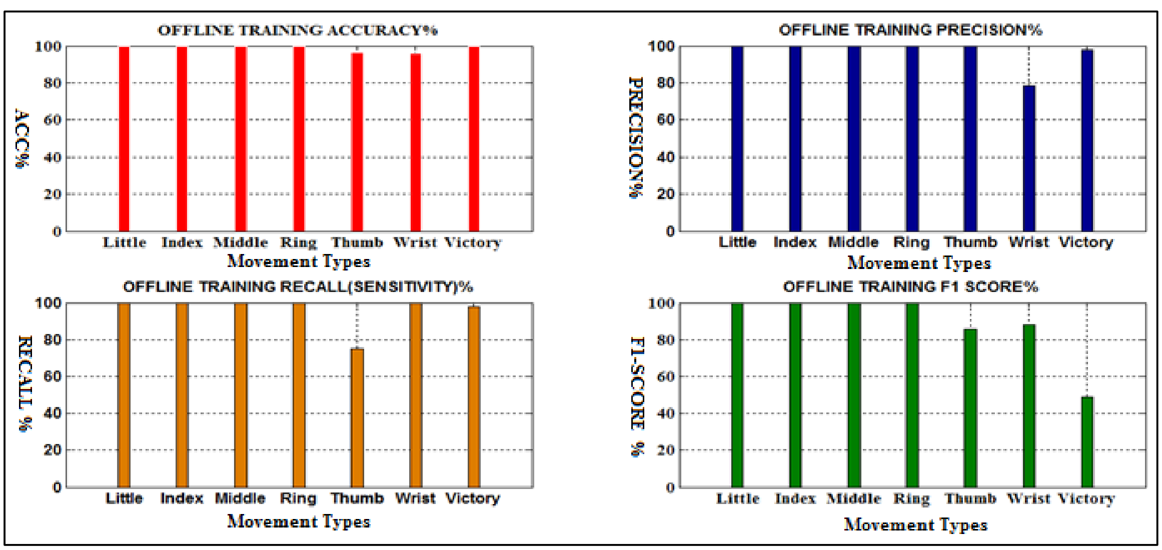

Figure 12. Percentage Accuracy, Precision, Recall (Sensitivity) and F1-Score at offline training.

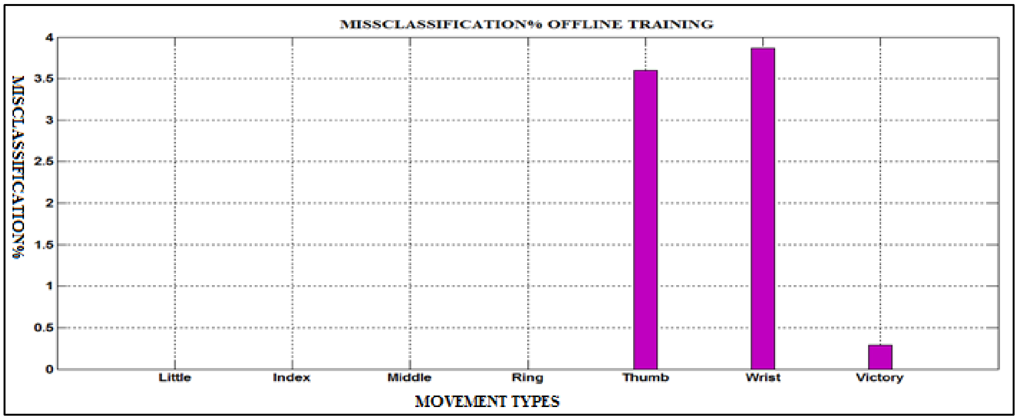

Figure 13. Percentage Misclassification at offline training.

Figure 14., shows that the loss and accuracy curves during training and validation on exercise Dataset. The Loss becomes very low, while the accuracy becomes high at 40 epochs.

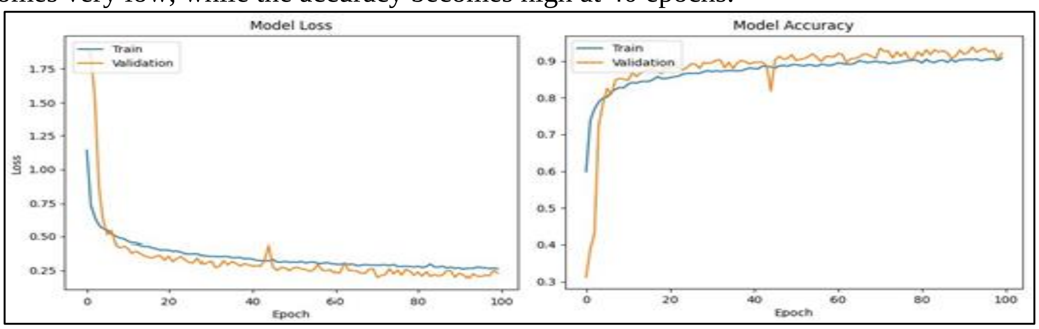

Figure 14. The loss and accuracy curves in offline.

9.1. The testing performance results

In the CNN testing, at 40 epoch the recognizing of the flexion movements for wrist and each finger except thumb and victory movements is 100%. While, the recognizing of the thumb flexion and victory movements are 73.5% and 98%, respectively.
It is clear that the confusion of each movement except thumb and victory is 0%, while the confusion of the thumb flexion and victory movements are 26.5% and 2%, respectively, as shown at table 12. It is clear that the thumb training and testing give the worst result.

Table 12. Confusion matrix of classifying the studied movements during offline testing.

Signal movement	True classification						
	Little	Index	Middle	Ring	Thumb	Wrist	Victory
Little	49	0	0	0	0	0	0
Index	0	49	0	0	0	0	0
Middle	0	0	49	0	0	0	0
Ring	0	0	0	49	0	0	0
Thumb	0	0	0	0	36	13	0
Wrist	0	0	0	0	0	49	0
Victory	0	0	0	0	0	1	48

The results of the performance evaluation of the classification offline testing shown in table 13 besides, figures 15 and 16 respectively. It is noticed, that the minimum accuracy of classification at wrist and thumb flexion.

Wrist flexion gives minimum precision. While, the thumb flexion gives minimum sensitivity of classification. The minimum F1-score is done at thumb flexion movement.

The maximum misclassification results at wrist and thumb movements, while the classification of the rest of the studied movements is 100%.

Table 13. Performance evaluation of the classification offline testing.

Signal movement	Acc %	***Precision*** %	***Recall*** %	Mis- Classification %	***F1 score*** %
Little	100	100	100	0	100
Index	100	100	100	0	100
Middle	100	100	100	0	100
Ring	100	100	100	0	100
Thumb	96.2	100	73.47	3.94	84.7
Wrist	95.9	92.45	100	4.1	96.07
Victory	99.7	100	97.96	0.29	98.97

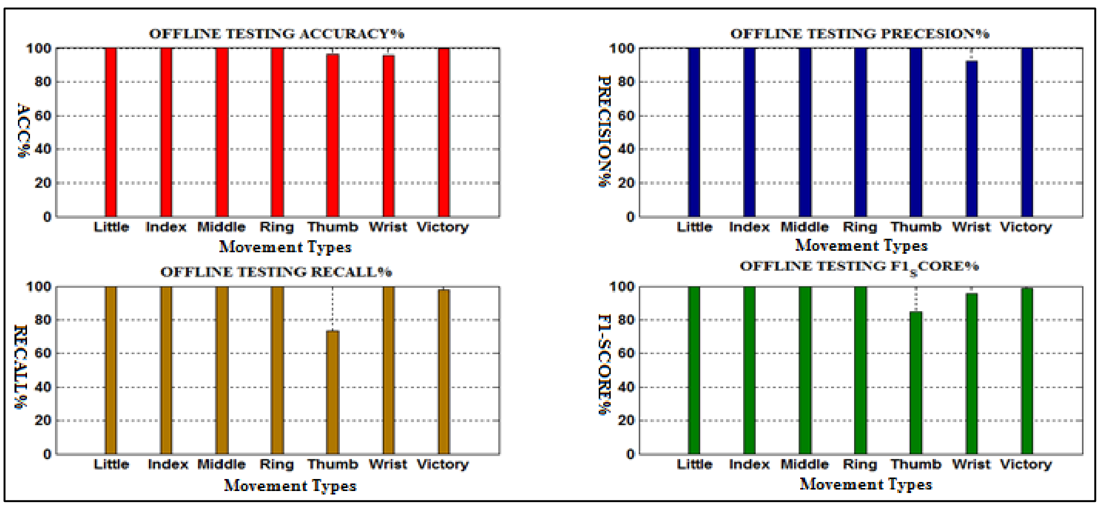

Figure 15. Percentage Accuracy, Precision, Recall (Sensitivity) and F1-Score at offline testing.

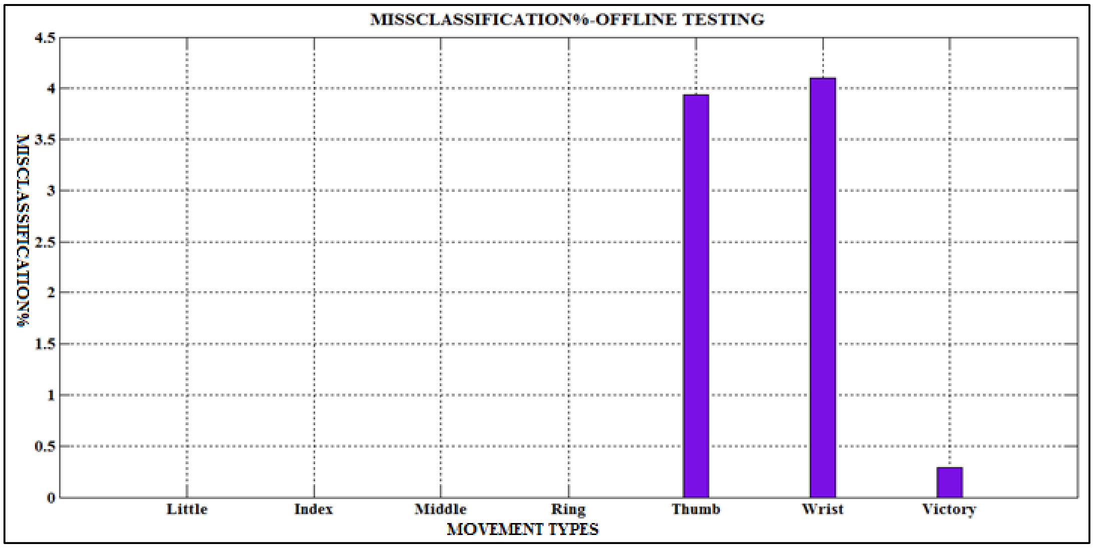

Figure 16. Percentage Misclassification at offline testing.

10. CONCLUSIONS

A CNN-based classification algorithm is proposed for five finger, wrist and victory gestures. The proposed method needs several experiments to tune the CNN layers parameter. This study presents more than one experiment on the selected Kaggle datasets in order to implement a human-computer interface based on EMG signal and DL. One of these datasets is a benchmark dataset. These experiments show that the overall performance (average accuracy) of the proposed method according to Table 9 is 98.83% and the overall average misclassification (error rate) is 1.17%. The overall sensitivity is 95.918%. the overall precision is 98.92% and overall F1-score is 97.105%.

Based on the obtained results and a number of experiments that were achieved in this study, the parameters of the CNN affects the performance of the proposed method so several experiments have been performed to find the proper parameter (such as filter size, pooling method, etc) that provide the best classification of the EMG signal. The model is able to learn the features and pattern of sEMG signals with relatively high accuracy when applied in both offline and online modes. The parameters found in the offline experiments based on the benchmark dataset have good performance. Likewise, the system can be used by any person, whether disabled or healthy, and it can also be used in the Internet of Things to control a specific system

It is concluded that the model is able to learn the features and patterns of sEMG signals with relatively high accuracy when tested. So, according to this study, the CNN is suitable to increase the success of the classification of the daily used finger/hand movements.

REFERENCIAS

Akmal, M., Qureshi, M. F., Amin, F., Rehman, M. Z. U., & Niazi, I. K. (2021). SVM-based real-time classification of prosthetic fingers using myo armband-acquired electromyography data. *2021 IEEE 21st International Conference on Bioinformatics and Bioengineering (BIBE)*, 1–5.

Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M. A., Al-Amidie, M., & Farhan, L. (2021). Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data*, *8*, 1–74.

Amado Laezza, R. (2018). *Deep neural networks for myoelectric pattern recognition-An implementation for multifunctional control*.

Andrews, A., Morin, E., & McLean, L. (2008). *Finger movement classification using forearm EMG signals*.

Atzori, M., Cognolato, M., & Müller, H. (2016). Deep learning with convolutional neural networks applied to electromyography data: A resource for the classification of movements for prosthetic hands. *Frontiers in Neurorobotics*, *10*, 9.

BAKIRCIOĞLU, K., & Özkurt, N. (2020). Classification of EMG signals using convolution neural network. *International Journal of Applied Mathematics Electronics and Computers*, *8*(4), 115–119.

Bhatti, S. A. (2019). *Finger Movement Classification via Machine Learning using EMG Armband for 3D Printed Robotic Hand*. University of Minnesota.

Caceres, C. A. (2014). *Machine learning techniques for gesture recognition*. Virginia Tech.

Chang, K.-M., Liu, P.-T., & Wei, T.-S. (2022). Electromyography Parameter Variations with Electrocardiography Noise. *Sensors*, *22*(16), 5948.

Chen, L., Fu, J., Wu, Y., Li, H., & Zheng, B. (2020). Hand gesture recognition using compact CNN via surface electromyography signals. *Sensors*, *20*(3), 672.

Choi, H.-S. (2023). Electromyogram (EMG) Signal Classification Based on Light-Weight Neural Network with FPGAs for Wearable Application. *Electronics*, *12*(6), 1398.

Gadekallu, T. R., Alazab, M., Kaluri, R., Maddikunta, P. K. R., Bhattacharya, S., & Lakshmanna, K. (2021). Hand gesture classification using a novel CNN-crow search algorithm. *Complex & Intelligent Systems*, *7*, 1855–1868.

Ghosh, A., Sufian, A., Sultana, F., Chakrabarti, A., & De, D. (2020). Fundamental concepts of convolutional neural network. *Recent Trends and Advances in Artificial Intelligence and Internet of Things*, 519–567.

Guo, B., Ma, Y., Yang, J., Wang, Z., & Zhang, X. (2020). Lw-CNN-based myoelectric signal recognition and real-time control of robotic arm for upper-limb rehabilitation. *Computational Intelligence and Neuroscience*, *2020*.

Karimi, Z. (2021). Confusion Matrix. *Encycl. Mach. Learn. Data Min., No. October*, 260.

Kiranyaz, S., Avci, O., Abdeljaber, O., Ince, T., Gabbouj, M., & Inman, D. J. (2021). 1D convolutional neural networks and applications: A survey. *Mechanical Systems and Signal Processing*, *151*, 107398.

Kulkarni, A., & Batarseh, F. A. (2021). Confusion Matrix—an overview| ScienceDirect Topics. *URL Https://Www. Sciencedirect. Com/Topics/Engineering/Confusion-Matrix*.

Kundu, B., & Naidu, D. S. (2021). Classification and feature extraction of different hand movements from EMG signal using machine leaning based algorithms. *2021 International Conference on Electrical, Communication, and Computer Engineering (ICECCE)*, 1–5.

Lee, K. H., Min, J. Y., & Byun, S. (2021). Electromyogram-based classification of hand and finger gestures using artificial neural networks. *Sensors*, *22*(1), 225.

Malvuccio, C., & Kamavuako, E. N. (2022). The Effect of EMG Features on the Classification of Swallowing Events and the Estimation of Fluid Intake Volume. *Sensors*, *22*(9), 3380.

McIntosh, J., McNeill, C., Fraser, M., Kerber, F., Löchtefeld, M., & Krüger, A. (2016). EMPress: Practical hand gesture classification with wrist-mounted EMG and pressure sensing. *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, 2332–2342.

Sadhu, S. R. (2019). *Classification of sEMG signals and Motion data using learning algorithms with application to post-stroke impairment*.

Stephenson, R. M., Chai, R., & Eager, D. (2018). Isometric finger pose recognition with sparse channel SpatioTemporal EMG imaging. *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 5232–5235.

Too, J., Abdullah, A. R., Zawawi, T. N. S. T., Saad, N. M., & Musa, H. (2017). Classification of EMG signal based on time domain and frequency domain features. *International Journal of Human and Technology Interaction (IJHaTI)*, *1*(1), 25–30.

Tsinganos, P., Cornelis, B., Cornelis, J., Jansen, B., & Skodras, A. (2018). Deep Learning in EMG-based Gesture Recognition. *PhyCS*, 107–114.

Venugopal, G., & Ramakrishnan, S. (2014). Differentiating sEMG signals under muscle fatigue and non-fatigue conditions using logistic regression classifiers. *Biomedical Sciences Instrumentation*, *50*, 314–321.

Xing, K., Ding, Z., Jiang, S., Ma, X., Yang, K., Yang, C., Li, X., & Jiang, F. (2018). Hand gesture recognition based on deep learning method. *2018 IEEE Third International Conference on Data Science in Cyberspace (DSC)*, 542–546.

SEMBLANZA DE LOS AUTORES

Musaab Saleh Dawood: was born on August 30, 1979 in Mosul, Iraq and obtained his Bachelor's degree in Computer Technology Engineering from the College of Al-Hadbaa University / Mosul, University of Al-Hadbaa in 2012. He is currently a graduate student / researcher in Computer Technology Engineering from the College Technical Engineering Department/Mosul/ Northern Technical University. He can be contacted via e-mail: musaab.dawood@ntu.edu.iq .

Mohand Lokman Aldabag: completed his PhD. Program in Yasar University in 2019. He received his B.E. in computer engineering degree from technical college\Mosul and his MSc. Degree in computer engineering at Mosul University, in 1998 and 2002 respectively. He is working at northern technical university\computer engineering technology dept. as assistant prof. He published six international articles in signal processing and two national articles. His region of interest signal processing, image processing, data mining and AI. His email= mohandaldabag@ntu.edu.iq .