

# A novel pooling layer based on gaussian function with wavelet transform

Aqeel M. Hamad alhussainy, Ammar D. Jasim

Department of Information and Communication Engineering, Al-Nahrain University, Iraq

---

## Article Info

### Article history:

Received Apr 1, 2020

Revised May 4, 2020

Accepted Jul 1, 2020

---

### Keywords:

CNN

Gaussian

GWT1

GWT2

GWT3

---

## ABSTRACT

Convolution represent basic layer in the convolutional neural network, but it can result in big size of the data, which may increase the complexity of the network. Different pooling methods are used to perform down sample these data. In this paper, we have proposed a novel pooling method by using Gaussian function to determine the wavelet filter coefficients. At first, the basic statistics are determined for each pool size of the signal, then Gaussian probability distribution function is determined. According to the procedure of extracting the features, three methods are proposed, the first method is used the normalized values of basic statistics as wavelet filter to be multiplied by original signal, the second method used the determined statistics as features of the original signal, then multiplied it with constant wavelet filter based on Gaussian, while the third method is similar to first method, except it depend on entire signal instead of each pool size. The proposed methods are combined with other standard methods such as max and pooling. The experiments are performed on different datasets, and the results show that the proposed methods perform or outperform other methods and can increase performance of the CNN.

Copyright © 2020 Institute of Advanced Engineering and Science.  
All rights reserved.

---

### Corresponding Author:

Aqeel M.Hamad,

Department of Information and Communication Engineering,

Al-Nahrain University, Baghdad. Iraq.

Email: aqeel\_alhussainy@utq.edu.iq

---

## 1. INTRODUCTION

Convolutional neural networks (CNN) are used in various applications in recent years, such as recognition, regression and classification. Convolution represent basic layer in the convolutional neural network, but it can result in big size of the data, and this data is increased with increasing of the number of filters, channels and data used for training, which may increase the complexity of the network and reduce the accuracy of the classification. Different studies and researches have been made to improve this network. Pooling layer is key component of the CNN, it can be used for down sampling the features, which are result from convolution layer to increase the efficiency and reduce the complexity of the computation [1, 2]. Different types of pooling methods are used, the most famous is max-pooling, which is the most important method also average pooling method is used in some CNNs, the type of pooling is depended on the dataset type (significant of the value in each element) and the application that is used in it, other pooling methods are proposed to improve the performance of those methods by using some prior models related to pooling based on other functions, which are used to down sample the data [3-5].

Some image processing techniques are used, such as image down sample method [6] detail preserving [7] and other method was based on mixing pooling with gated-pooling [8], while the recent methods are the pooling derived based on entropy principle [9, 10]. Along with those deterministic methods, stochastic method were proposed [10], this method was used to improve the local pooling method by adding randomness to those methods, also stochastic method can be used with mixing of max and average method [11], other methods were

based on multi levels wavelet transform [12]. Unfortunately most of the previous methods have shortcoming. The most famous max pooling layer, which is selected the maximum element only, can delete details from signal especially when there is high contrast between the elements of the signal [9], while average pooling method can discard the effect of the details by averaging the data to lower than significant details [10], this may lead to omission important information from the data and reduce the accuracy of the model. In this paper, we have proposed a new pooling method based on the Gaussian probability density distribution function (pdf), which is avoided the problem of discarding significant information by determine the basic statistics to be used as parameters to select the best features of the signal [13-15]. The main contribution of this method is to use the Gaussian distribution to determine the coefficients of wavelet transform according to some statistics such as mean and standard deviation for each pool size ,thus we can obtain different coefficients for each window depending on its characteristics and significant, and this enabled us to extract the best feature of the signal. According to the procedure of extracting the wavelet coefficients filter, three method are proposed, the first method (GWT1 ) is used the normalized values of basic statistics as weight to be multiplied by original signal, the second method (GWT2) used the determined statistics as features of the original signal and multiply it with constant weights based on Gaussian function, while the third method (GWT3) is worked in a similar way to (GWT1) except that, it depended on entire signal instead of every pool size for calculation the basic statistics.

## 2. METHODOLOGY

In this paper, we have proposed anew pooling layer based on Gaussian function, which is used as a method to determine the coefficients of wavelet filters. The input signal is divided into window (according to the pool size). The basic statistics are computed, which are (mean of the signal and stander d deviation) by equation (1 and 2) respectively, then Gaussian probability density function (pdf) is determine for each pool size. Figure 1 shows the block diagram of the proposed pooling layer [16-18].

For each pool size, the basic statistics are calculated, which are mean and standard deviation, then five values are determined according to the calculated statistics, which are  $\mu_x - 2\sigma_x$ ,  $\mu_x - 1\sigma_x$ ,  $\mu_x$ ,  $\mu_x + \sigma_x$  and  $\mu_x + 2\sigma_x$ , then these values are used as input to the Gaussian function, which is used to compute the Gaussian amplitudes , these amplitudes are used as weights, which are multiplied by the original signal as in (3) to determine the pooled signal [19-21].

$$\mu_x = \frac{1}{|R|} \sum_{(p \in R)} X_p \quad (1)$$

$$\sigma_x^2 = \frac{1}{|R|} \sum_{(p \in R)} (X_p - \mu_x)^2 \quad (2)$$

$$Y = \sum_{i=-2}^2 (\mu_x + i * \sigma_x) * h(i) \quad (3)$$

and the stride value represented down sample factor.

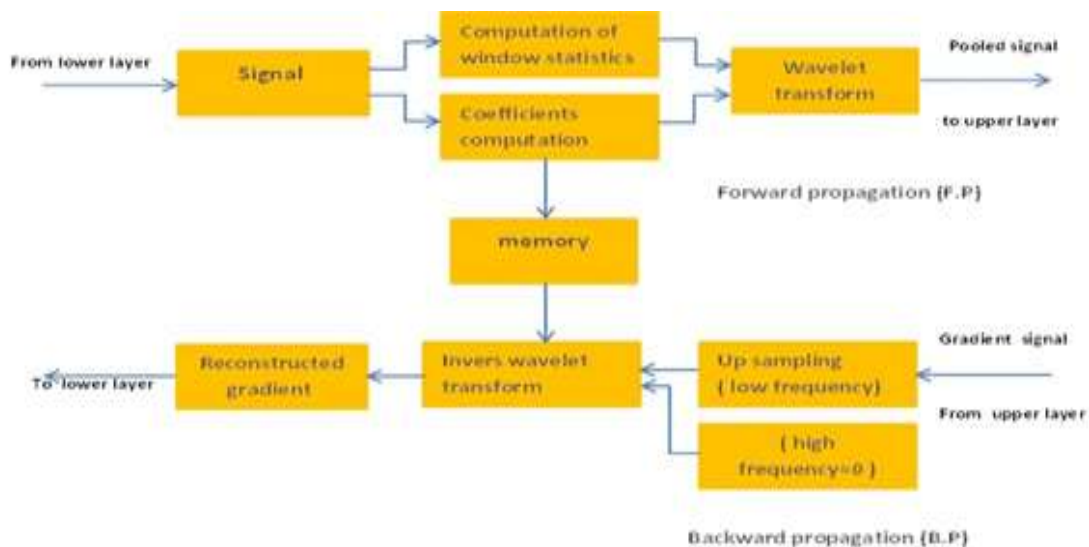


Figure 1. Block diagram of the proposed pooling layer

**2.1. Gaussian probability distribution function.**

The probability density function (PDF) represent normal distribution of variable  $x$  with mean ( $\mu$ ) and standard deviation ( $\sigma$ ), which is described by the following (4, 5) [22, 23]:

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{4}$$

The Gaussian distribution of  $x$  is represented by:

$$X \sim N(\mu, \sigma) \tag{5}$$

it is clear that, there are two parameters characterized the Gaussian (pdf), which are ( $\mu, \sigma$ ) that represent the first and second order moment respectively and can be defined by: [23, 24]:

$$\mu = E[X] = \int_{-\infty}^{\infty} x f(x) dx \tag{6}$$

$$\sigma = E[x - \mu]^2 = \int_{-\infty}^{\infty} (x - \mu)^2 f(x) dx \tag{7}$$

Mean represent the centroid of the pdf and the coordinate of the mean value, while ( $\sigma$ ) is the dispersion of the variable around mean value. Depending on the value of  $\sigma$  the value of Gaussian function is determined, so there are different weight will be used according to the significant of the  $x$  value as shown in Figure 2, which is describe three Gaussians (PDF) for  $\mu=3$  and  $\sigma = 1.0, 0.75$  and  $0.7$ .

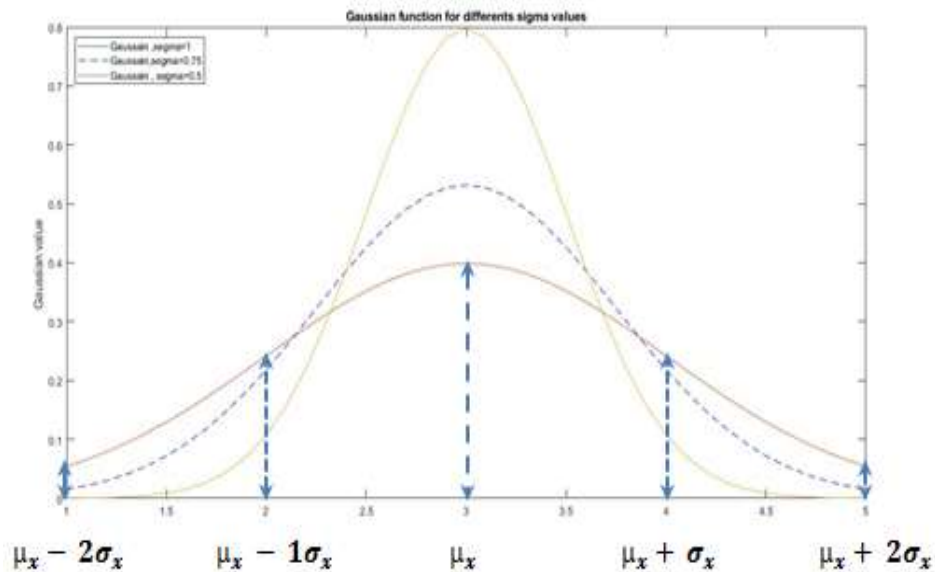


Figure 2. Gaussian PDF for three sigma values ( $\sigma = 1.0, 0.75$  and  $0.5$ )

As shown in Figure (2), the mean value ( $\mu$ ) have the highest value, while the other elements ( $\mu - \sigma, \mu - 2\sigma, \mu + \sigma, \mu + 2\sigma$ ) have smallest value, so its weight will be small depending on the value of Gaussian function, so the features will be selected in more suitable manner by giving the mean value the highest weight and the weight is decreased as moving from the mean according to the significance of the statistics, as shown in Figure 2, ( $\mu + \sigma$ ) and  $\mu - \sigma$  have weight smaller than  $\mu$ , also ( $\mu + 2\sigma$ ) and  $\mu - 2\sigma$  have lowest values, this can increase the obtained information from using the mean only. The other contributions of this method (GWT) is proposed feature selection method based on statistics of each pool as described in the different Gaussian function in Figure 2, which is described the differences between these different Gaussian function values according to  $\mu$  and  $\sigma$  [24-26].

## 2.2. Proposed algorithms (GWT)

Three different algorithms are proposed according to the procedure of extracting the wavelet coefficients, which are used to reduce the features of the signal to ensure that, the basic information is extracted with little elimination. Also the proposed methods are combined with other standard methods such as max and average pooling.

### 2.2.1. GWT 1 algorithm

This algorithm is based on computation of the wavelet coefficients, by first determine the basic statistics (mean and standard deviation) for each pool size (window), then five values are derived based on these statistics, then the Gaussian magnitudes for those points will be determined by Gaussian function by applying (4) to be used as weights, The original signal (pool window) are multiplied by the weights to determine the pooled signal. Figure 3 show the flowchart of the proposed GWT1 algorithm.

### 2.2.2. GWT 2 algorithm

In this algorithm, the weights are determined based on constant Gaussian function with  $\mu_x = 0$  and  $\sigma_x = 1$ , then five weights are determined by using Gaussian function by (4), while the signal is represented by the basic features, which are  $\mu_x - 2\sigma_x$ ,  $\mu_x - 1\sigma_x$ ,  $\mu_x$ ,  $\mu_x + \sigma_x$  and  $\mu_x + 2\sigma_x$  according to the statistics for each pool size (window), then the pooling features are determined by multiplication of significant features with the weights as shown in Figure 4.

### 2.2.3. GWT 3 algorithm

This algorithm is similar to algorithm 1 (GWT1) except that, the statistics are computed for entire signal instead of each pool size. The feature is computed by multiplying the original signal with weights of the basic statistics of entire signal as described in Figure 5, which shows the flowchart of this algorithm.

---

#### Algorithm 1 (GWT1)

Input= $X_p \in R$  input from lower layer;

Output= $Y$  pooling feature for upper layer

---

1. Initialization  $Z$ : size of signal,  $P$  pool size;  $S$ : stride
2. for  $i=1$  to  $Z$  step  $S$  do
3. signal determination  $X_p \leftarrow$  signal window
4. for  $j=1$  to  $P$  do
5. Determine  $\mu_x$  and  $\sigma_x$   
Where  $\mu_x = \frac{1}{|R|} \sum_{(p \in R)} X_p$ ,  $\sigma_x = \frac{1}{|R|} \sum_{(p \in R)} (X_p - \mu_x)^2$ ,
6. Determine the value  $\mu_x - 2\sigma_x, \mu_x - 1\sigma_x, \mu_x, \mu_x + \sigma_x$  and  $\mu_x + 2\sigma_x$
7. Determine the value of  $f(x)$  for the above values of  $x$  by using:  
$$f(x) = \frac{1}{\sqrt{2\pi\sigma_x^2}} e^{-\frac{(x-\mu_x)^2}{2\sigma_x^2}}$$

To be  $h1, h2, h3, h4$ , and  $h5$  respectively.

8. Determine the pooling signal by:  
 $Y = \sum_{i \in X_p} (X_p * h(i))$
9.  $j=j+1$ ;
10. End loop
11.  $i=i+P$ ;
12. End loop
13. for  $i=1$  to  $Z$  step  $S$  do
14. for  $j=1$  to  $P$  do
15. Chose max., average, approximation
16. Determine pooling as combination of any two or three above methods  
 $Y = (\max + \text{pool}) / 2$ ;
17. End loop
18. End loop
19. Return  $Y$

End algorithm  
End B.W

Figure 3. Flowchart of the proposed GWT1) algorithm

**Algorithm II (GWT2)**

Input= $X_p \in R$  input from lower layer;

Output= $Y$  pooling feature for upper layer

Initialization  $Z$ : size of signal,  $P$  pool size;  $S$ : stride

Read the input signal

While (Pool\_size in signal)

Do{

1. Get window= pool\_size;  $X_p \leftarrow$  signal window

2. Determine mean and standard deviation of the window by:

$$\mu_x = \frac{1}{|R|} \sum_{(p \in R)} X_p, \sigma_x^2 = \frac{1}{|R|} \sum_{(p \in \text{window})} (X_p - \mu_x)^2,$$

3. Give  $x$  initial value  $x = [-2:1:2]$

4. Determine constant weight based on Gaussian function to be:  $w1, w2, w3, w4, w5$

5. Determine the value of

$$\mu_x - 2\sigma_x, \mu_x - 1\sigma_x, \mu_x, \mu_x + \sigma_x \text{ and } \mu_x + 2\sigma_x$$

6. Determine the value of  $f(x)$  for the above values of  $x$  by using:

$$f(x) = \frac{1}{\sqrt{2\pi\sigma_x^2}} e^{-\frac{(x - \mu_x)^2}{2\sigma_x^2}}$$

To be  $h1, h2, h3, h4,$  and  $h5$  respectively.

7. Determine the pooling signal by:

$$Y = \sum_{i \in X_p} (X_p * h(i));$$

8.  $j = j + 1;$

9. End loop

End of algorithm

Figure 4. Flowchart of the proposed GWT2) algorithm

**Algorithm III (GWT3)**

Input= $X_p \in R$  input from lower layer;

Output= $Y$  pooling feature for upper layer

1. Initialization  $Z$ : size of signal,  $P$  pool size;  $S$  : stride

2. Determine  $\mu_x$  and  $\sigma_x$

3. for  $i=1$  to  $Z$  step  $S$  do

4. Determine signal window  $X_p \leftarrow$  signal window

5. for  $j=1$  to  $P$  do

i. Determine

$$\mu_x = \frac{1}{|R|} \sum_{(p \in R)} X_p, \sigma_x = \frac{1}{|R|} \sum_{(p \in R)} (X_p - \mu_x)^2,$$

ii. Determine the value of :

$$\mu_x - 2\sigma_x, \mu_x - 1\sigma_x, \mu_x, \mu_x + \sigma_x \text{ and } \mu_x + 2\sigma_x$$

iii. Determine the value of  $f(x)$  for the above values of  $x$  by using:

$$f(x) = \frac{1}{\sqrt{2\pi\sigma_x^2}} e^{-\frac{(x - \mu_x)^2}{2\sigma_x^2}}$$

To be  $h1, h2, h3, h4,$  and  $h5$  respectively.

iv. Determine the pooling signal by:

$$Y = \sum_{i \in X_p} (X_p * h(i));$$

v.  $j = j + 1;$

6. End loop

End algorithm

Figure 5. Flowchart of the proposed GWT3 algorithm

### 3. RESULTS AND DISCUSSION

The proposed methods are trained and tested with different CNN networks and we used three types of datasets to train the models, these datasets are ECG dataset (which is one dimension signal), while for two dimensions signal, MNIST and CIFAR10 datasets were used we apply all the proposed methods to different convolutional neural network (CNN). The experiments are performed on Mat lab (2019a) by Intel © core™i7-4500CPU@2.40GHz processor, with 8GB of RAM, 64bit windows seven operating system. The results are described as following:

### 3.1. Results of one dimension signal (MIT-BIH ECG dataset)

This database is used for analysis and detection of heart arrhythmia. MIT-BIH database is one dimensional dataset, and their size is (109446). The dataset is splitted into train set (87554) and test set (21892) signal. Each signal contains (188) sample. In our experiments, the model is trained with batch size=128, 10 epochs and 684 iteration. The results of first method (GWT1) are compared with max and average pooling methods as shown in Table 1. The best accuracy (92.57) was achieved by using (GWT1) method, which is outperformed average method and almost equal to max pooling method. It was further observed that combining GWT1 method with average method give worst results because average method eliminates the details of the signal, especially for ECG signal, which is characterized by being oscillating signal. Table 2 shows results of the second method (GWT2), it gives the best results (Accuracy=93.97%), which is outperformed max and average pooling methods, the reason for this improvement in accuracy is the algorithms depend on features as mentioned previously, because ECG is oscillatory signal ,so best features are selected based on statistics for each pool.

Table 1. Results of GWT1 with other methods for MIT\_BIH dataset

Method	Max	average	GWT1	GWT1+Max	GWT1+Average
Accuracy (%)	92.55	91.59	92.57	92.21	92

Table 2. Results of GWT2 with other methods for MIT\_BIH dataset

Method	Max	average	GWT2	GWT2+Max	GWT2+Average
Accuracy (%)	92.55	91.59	93.97	93.25	93.51

Table 3 explains the results of the third proposed method (GWT3), which is achieved the lowest results, the reason is to use entire signal to compute the weights, which is used to select feature of the signal, this produce constant weights, thus the method losses adaptive property. The progress of accuracy for (GWT2) method is shown in Figure 6, while the progress of loss is shown in Figure 7, as shown in figure, the loss rate is reached to less than 0.25 at the final result.

Table 3. Results of GWT3 with other method MIT\_BIH dataset

Method	Max	average	GWT3	GWT3+Max	GWT3+Average
Accuracy (%)	92.55	91.59	92.50	<b>92.37</b>	92.10

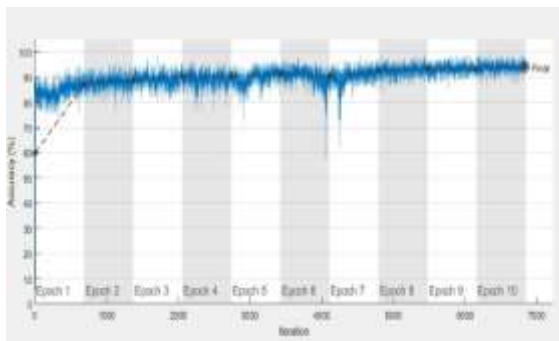


Figure 6. Accuracy progress for (GWT2) method

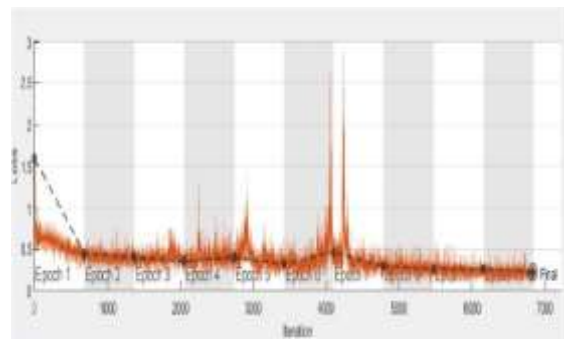


Figure 7. Loss progress for (GWT2) method

### 3.2. Results of MNIST dataset

It is one of the most common type of database, which is used to train two-dimensional model, this dataset contains (28\*28) gray scale image, and its total size is 60000 image, the Training part (50000) image, while test part 10000 image. The CNN was trained with initial learning rate 0.01, 10 epochs and 58 iteration per epoch. The performance of the first method is shown in Table 4, the best accuracy was achieved by (GWT1) method, which is achieved (99.84%). Because MNIST database is grayscale image, and almost black background images, so maximum element represents important information, this is what happened by max pooling method, but our proposed method gives better accuracy (99.84%), this is due to a reduction in eliminated details of

the image. Table 5 shows the different performance metrics for (GWT1) method, it is achieved best sensitivity (99.97%) and lowest FPE (0.03%).

Table 4. Results of (GWT1) for MNIST dataset

Method	Max	average	GWT1	GWT1+Max	GWT1+Average
Accuracy (%)	98.80	98.72	99.84	<b>99.84</b>	99.76

Table 5. Performance metrics for (GWT1) for MNIST dataset

Method	GWT1	GWT1+Max	GWT1+Average
Accuracy (%)	99.84	<b>99.72</b>	99.76
Sensitivity (SN%)	99.97	<b>99.72</b>	99.76
False Error Rate FER (%)	0.03	<b>0.28</b>	0.24
Specificity (%)	99.52	<b>99.68</b>	99.78
ERR (%)	0.16	0.16	0.24

The results of second method (GWT2) is shown in Table 6. (GWT1+Max) is achieved best results (acc=99.96%) because this method is depended on feature of the signal, so the combination of it with Max pooling can reduce the dropping of details information to minimum extent possible and achieved high accuracy and improvement in all performance metrics as shown in Table 7, since it is achieved high accuracy (99.96) with high sensitivity (99.96%) and lowest FPR (0.04%). Figure 8 shows the accuracy progress for (GWT2) method, it is clear that, the model is needed just 3 epochs to reach to more than (95% accuracy ) and to minimize the loss rate to less than 0.25 as shown in Figure 9, which shows the loss curve for the same method. The confusion matrix details of this method is described in Table 8, as we note, there is only one value is not correct, while all the other diagonal element of the matrix are 250, which means there are high match between the target and actual classes, also the specificity and sensitivity are 100% for all classes except for one mistake. The results of third method (GWT3) are presented in Table 9, the best accuracy is achieved by (GWT3+average) method, because this method is depended on the entire signal statistics, average method can provide the basic features and give better results after combined it with proposed GWT3 as shown in the detail description results in Table 10.

Table 6. Results of (GWT2) for MNIST dataset

Method	Max	average	GWT1	GWT1+Max	GWT1+Average
Accuracy (%)	98.80	98.72	99.64	<b>99.96</b>	99.80

Table 7. Performance metrics for (GWT2) for MNIST dataset

Method	GWT2	GWT2+Max	GWT2+Average
Accuracy (%)	<b>99.52</b>	<b>99.96</b>	99.80
Sensitivity (SN%)	<b>99.97</b>	<b>99.96</b>	99.84
False Error Rate FER (%)	<b>0.03</b>	<b>0.04</b>	0.16
Specificity (%)	<b>99.52</b>	<b>99.84</b>	99.80
ERR (%)	<b>0.48</b>	0.16	0.20

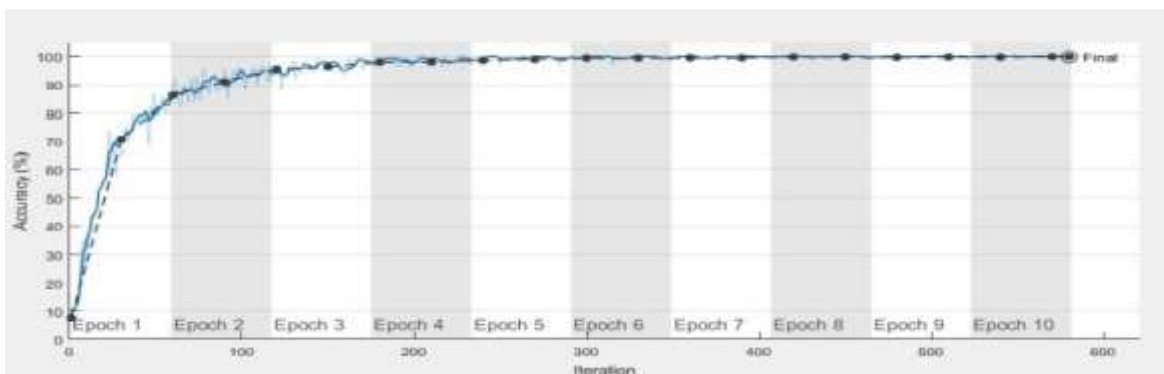


Figure 8. Accuracy progress for (GWT2) method for MNIST dataset

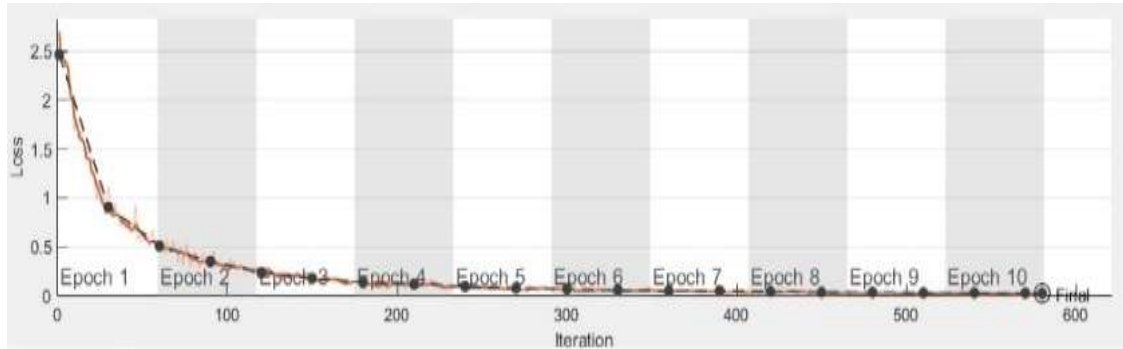


Figure 9. Loss progress for (GWT2) method for MNIST dataset

Table 8. Confusion matrix of (GWT2) method for MNIST dataset

Target Class	Class 0	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Class 8	Class 9	sensitivity
Class 0	<b>250</b> 10%	0	0	0	0	0	0	0	0	0	100%
Class 1	0	<b>250</b> 10%	0	0	0	0	0	0	0	0	100%
Class 2	0	0	<b>250</b> 10%	0	0	0	0	0	0	0	100%
Class 3	0	0	0	<b>250</b> 10%	0	0	0	0	0	0	100%
Class 4	0	0	0	0	<b>250</b> 10%	0	0	0	0	0	100%
Class 5	0	0	0	0	0	<b>250</b> 10%	0	0	0	0	100%
Class 6	0	0	0	0	0	0	<b>250</b> 10%	0	0	0	100%
Class 7	0	0	0	0	0	0	0	<b>250</b> 10%	0	0	99.96%
Class 8	0	0	0	0	1	0	0	0	<b>250</b> 10%	0	100%
Class 9	0	0	0	0	0	0	0	0	0	<b>250</b> 10%	100%
	100%	100%	100%	100%	99.96%	100%	100%	100%	100%	100%	100%

Table 9 results of (GWT3) for MNIST dataset

Method	Max	average	GWT3	GWT3+Max	GWT3+Average
Accuracy (%)	98.80	98.72	99.60	<b>99.64</b>	99.80

Table 10. Performance metrics for (GWT3) for MNIST dataset

Method	GWT3	GWT3+Max	GWT3+Average
Accuracy (%)	99.60	<b>99.64</b>	99.80
Sensitivity (SN%)	99.6	99.64	99.80
False Error Rate FER (%)	0.04	<b>0.036</b>	0.2
Specificity (%)	99.60	99.96	99.80
ERR (%)	0.04	0.036	0.2

### 3.3. Results of CIFAR 10 dataset

The third type of database used in this paper is CIFAR10 database, the size of this database is (60000\*28\*28\*3), where 3 represents RGB image, while (28\*28) represents the size of image. There are 60000 image, 50000 of them are used for training the model, while the rest are used for testing the model accuracy. The model is trained with initial learning rate (0.01), 10 epoch, 390 iteration per epoch and minimum batch used is 128.



The result of the proposed methods GWT1, GWT2 and GWT3 are presented in Tables 11, 12 and 13 respectively. As we note, the proposed methods are achieved the best results, which are (72.68%), (73.12%) at GWT1 and GWT2 respectively, while GWT3 gives accuracy less than Max and average method. The best result is achieved with GWT2 method ,because this database have RGB image with different characteristics, which means that there are some image have spatial redundant in neighboring pixels, GWT2 method can extract the most significant features from the signal according to the basic statistics, which represents most important features and gives best results as shown in the detail description of this method in Table 14 ,which describes confusion matrix between satisfied result and actual result.

Table 11. Results of (GWT1) method for CIFAR10 dataset

Method	Max	average	GWT1	GWT1+Max	GWT1+Average
Accuracy (%)	72.59	72.4	<b>72.68</b>	72.18	72.12

Table 12. Results of (GWT2) method for CIFAR10 dataset

Method	Max	average	GWT2	GWT2+Max	GWT2+Average
Accuracy (%)	72.59	72.4	<b>73.12</b>	72.62	72.44

Table 13. Results of (GWT3) method for CIFAR10 dataset

Method	Max	average	GWT3	GWT3+Max	GWT3+Average
Accuracy (%)	72.59	72.4	<b>72.35</b>	72.29	71.57

Table 14. Confusion matrix of (GWT2) method for CIFAR10 dataset

		Target Class										
		airplane	Sensitivity %	truck	ship	hours	frog	dog	deer	cat	bird	automobile
Output Class	airplane	<b>798</b> <b>8.0%</b>	25	75	25	25	17	5	19	83	39	71.85
	automobile	10	<b>803</b> <b>8.0%</b>	05	6	4	5	3	4	18	57	87.8
	bird	44	1	<b>592</b> <b>5.9%</b>	78	62	62	42	40	15	4	63.0
	cat	18	11	52	<b>561</b> <b>5.6%</b>	60	161	70	42	14	9	55.1
	deer	24	4	110	79	<b>726</b> <b>7.3%</b>	63	64	75	5	4	62.9
	dog	4	4	67	131	25	<b>599</b> <b>6.0%</b>	22	55	4	3	65.5
	frog	9	8	56	55	40	27	<b>770</b> <b>7.7%</b>	5	7	3	78.6
	hours	7	2	13	25	43	32	3	<b>731</b> <b>7.3%</b>	0	2	85.2
	ship	48	30	14	12	10	4	5	4	<b>826</b> <b>8.3%</b>	26	84.4
	truck	38	112	16	28	5	10	16	25	28	<b>853</b> <b>8.5%</b>	75.4
	specifity	79.8	80.3	59.2	56.1	72.6	59.9	77.0	73.1	82.6	<b>85.3</b>	<b>72.6</b>

#### 4. CONCLUSION

Convolution is the basic layer of the convolutional neural network (CNNs), but it can result in huge data in their output, which may increase the complexity of the network and reduce the accuracy of the network, so different studies and researches are performed to down sample and reduce the size of these data to increase the (CNN) efficiency and robustness, in this paper, a new pooling methods were proposed based on Gaussian probability distribution function (PDF), this function is used to determine the basic features of the signal, then different statistics with different weights were extracted and used as a coefficient for wavelet transform filter in different proposed methods. According to the procedure of extracting wavelet coefficients filter. Three methods are proposed, the first method (GWT1) is used the normalized values of basic statistics as weight to be multiplied by original signal, the second method (GWT2) used the determined statistics as features of the original signal

and multiply it with constant weights based on Gaussian function, while the third method (GWT3) is work in similar way to (GWT1) except that, it depended on entire signal instead of every pool size for calculation the basic statistics. Other methods are proposed by combined the previous methods with other standard methods such as max and average pooling, by first apply GWT method without down sampling (stride=1), then applying max or average or mixing of them. The experiments are performed on different data sets, which are two dimensions (MNIST and CIFAR10) and one dimension signal (ECG signal MIT-BIH dataset), the results are evaluated in terms of accuracy, FPR, and EER. For ECG signal MIT-BIH dataset, the proposed method achieved (Acc=93.97%), which is outperformed standard methods (Max (92.55%) and average (91.59%)), while for MNIST database, the proposed method is achieved (Acc=99.96%) greater than Max (98.8%) and average (98.72%), also for CIFAR10 database, the proposed method is achieved (73.12%) more than max (72.59%) and average (72.4%). Also the combinations of the proposed method with max and average method are achieved good improvement (for ECG database GWT2+Max satisfied 93.51%, MNIST database GWT2+Max satisfied 99.80% and for CIFAR10 GWT2+average achieved 72.62%). The proposed methods are perform or outperform previous methods and can be used as pooling methods in classification application.

## REFERENCES

- [1] K. He, X. Zhang, et al., "Deep residual learning for image recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, pp. 770-778, 2016.
- [2] A. G. Howard, et al., "Efficient convolutional neural networks for mobile vision applications," *arXiv, pp. 1704.04861*, 2017.
- [3] M. Riesenhuber and T. Poggio, "Just one view: Invariances in inferotemporal cell tuning," *Advances in neural information Processing Systems*, pp. 215-221, 1998.
- [4] M. Riesenhuber and T. Poggio, "Hierarchical models of object recognition in cortex," *Nature Neuroscience*, vol. 2, no. 11, pp. 1019-1025, 1999.
- [5] T. Serre and T. Poggio, "A neuromorphic approach to computer vision," *Communications of the ACM*, vol. 53, no. 10, pp. 54-61, 2010.
- [6] T. Williams and R. Li, "Wavelet pooling for convolutional neural networks," *Conference ICLR*, pp. 1-12, 2018.
- [7] F. Saedan, et al., "Detail-preserving pooling in deep networks," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, pp. 9108-9116, 2018.
- [8] C.-Y. Lee, P. W. Gallagher, and Z. Tu., "Generalizing pooling functions in convolutional neural networks: Mixed, gated, and tree," In *AISTATS*, vol. 51, pp. 464-472, 2016.
- [9] D. Yu, H., et al., "Mixed pooling for convolutional neural networks," *International Conference on Rough Sets and Knowledge Technology*, vol. 8818, pp. 364-375, 2014.
- [10] M. Zeiler and R. Fergus, "Stochastic pooling for regularization of deep convolutional neural Networks," In *ICLR*, 2013.
- [11] Takumi Kobayashi, "Gaussian-Based Pooling for Convolutional Neural Networks," *NIPS Proceedings*, 2019.
- [12] P. Liu, H. Zhang, et al., "Multi-level wavelet convolutional neural networks," *arXiv:1907.03128v1*, 2019.
- [13] Takumi Kobayashi, "Global Feature Guided Local Pooling," *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, Korea (South), pp. 3365-3374, 2019.
- [14] Pengju Liu, et al., "Multi-level Wavelet Convolutional Neural Networks," *arXiv:1907.03128v1*, 2019.
- [15] Shin Fujieda, et al., "Wavelet convolutional neural networks for texture classification," *arXiv:1707.07394*, 2017.
- [16] Chiraz Ben Chaabaneetal, "Wavelet convolutional neural networks for handwritten digits recognition," *International Conference on Hybrid Intelligent Systems*, vol. 734, pp. 305-310, 2017.
- [17] A. Pewsey, "Large-sample inference for the general half-normal distribution," *Communications in Statistics-Theory and Methods*, vol. 31, no. 7, pp. 1045-1054, 2002.
- [18] Timothy Dozat, "Incorporating Nesterov Momentum into Adam," *ICLR*, vol. 1, pp. 2013-2016, 2016.
- [19] Allison M Rossetto and Wenjin Zhou, "Improving Classification with CNNs using Wavelet Pooling with Nesterov-Accelerated Adam, EPiC Series in Computing," *Proceedings of 11th International Conference on Bioinformatics and Computational Biology*, vol. 60, pp. 84-93, 2019.
- [20] Travis Williams, Robert Li, "An Ensemble of Convolutional Neural Networks Using Wavelets for Image Classification," *Journal of Software Engineering and Applications*, vol. 11, no. 2, pp. 69-88, 2018.
- [21] S. Xie, R. Girshick, et al., "Aggregated residual transformations for deep neural networks," *CVPR*, pp. 5987-5995, 2017.
- [22] Ajay Kumar Boyat and Brijendra Kumar Joshi, "A rewiw paper: Noise models in digital image processing, Signal & Image Processing," *An International Journal (SIPIJ)*, vol. 6, no. 2, 2015.
- [23] Yaakov Bar-Shalom, X. Rong Li, "Estimation with Applications to Tracking and Navigation," *John Wiley & Sons*, 2001.
- [24] Guanping Lu, Jinsong Wu and Robert C. Qiu, "Analysis on the Empirical Spectral Distribution of Large Sample Covariance Matrix and Applications for Large Antenna Array Processing," in *IEEE Access*, vol. 7, pp. 30135-30141, 2019.
- [25] Vaibhav Arora and Ravi Kumar, "Probability Distribution Estimation of Music Signals in Time and Frequency Domains," *2014 19th International Conference on Digital Signal Processing*, Hong Kong, pp. 409-414, 2014.
- [26] Danilo Pena, Carlos Lima, Matheus Doria, Luan Pena, Allan Martin and Vicente Sousa, "Acoustic Impulsive Noise Based on Non-Gaussian Models: An Experimental Evaluation," *Sensors*, vol. 19, no. 12, pp. 2827-2844, 2019.