sciendo INTERNATIONAL JOURNAL ON SMART SENSING AND INTELLIGENT SYSTEMS

Article | DOI: 10.2478/ijssis-2022-0003 Issue 1 | Vol. 15 (2022)

Fast fourier transform based new pooling layer for deep learning

Aqeel Mohsin Hamad*

University of Thiqar, College of Computer and Mathematics, Computer Department, Nasiriyah, Iraq.

*E-mail: aqeelhamad@utq.edu.iq

The paper was edited by Subhas Chandra Mukhopadhyay.

Received for publication July 1, 2021.

Abstract

Convolution is considered most significant layer in deep learning because it can extract best features of data through the network but it may result in huge volume of data. This problem can be solved by using pooling. In this paper, A novel pooling method is proposed by using discrete Fourier transform (DFT), this method is used DFT technique to transform the data from spatial domain into frequency domain to preserve the most important information from the details coefficients, where the details information of the image is less significant, therefore it can be discarded to down sample the size of dimensions. Its effect will be great with advantage of reducing the eliminated details information as compared with other standard methods. After applying DFT, the most significant coefficients, which represent most important features are cropped while less important details will be discarded then the data are reconstructed by applying inverse DF, therefore the high quality of features are extracted, which solve the problem of losing significant information during the pooling layer. Different methods are proposed based on the scenario of using DFT. The proposed methods are tested by extracting pooled image then the original images were retrieved using only the pooled images. Then the retrieved images are compared with original images by using different measures such as SNR, correlation and SSIM. Then the proposed layers used for image classification for two different datasets. The results proved that the proposed methods outperformed standard methods, thus it can be used for deep learning application.

Keywords

CNN, DL, FFT, FTM and Features.

Introduction

Convolutional neural networks (CNN) have become growing area of interest in last years, because it can be used in different application such as image segmentation (Farabet et al., 2013), image classification (Krizhevsky and Hinton, 2017), object detection (Ren et al., 2017), image capturing (Vinyals et al., 2015), face recognition and more other applications. One of the most important layers of this network is convolution layer. It is considered one of the fundamental component of building blocks of a CNN network due to best extraction of features, which is required for most of application. It is used for extracting features from an input data and then feeding it to the next layer through the network, but the most disadvantage of this layer is that it is produces huge volume of data at its output based on input size, the number of filters and number of used channels, therefore high cost for computation and complexity is required, which may decrease efficiency of the model and therefore decrease the accuracy of model performance. Different methods are used to solve this problem, the most important method of them is using pooling layer to reduce the dimension of the data of convolution or down sampling the size of convolution layer output. The parameters

Open Access. Published by Sciendo. BY-NC-ND
© 2022 Authors. This work is licensed under the Creative Commons Attribution-Non-Commercial-NoDerivs 4.0 License https://creativecommons.org/licenses/by-nc-nd/4.0/

of pooling layer are the filter size and strides values. In order to reduce the data to its one quarter of original side, signal processing methods can be used to process the data by using different techniques such as using threshold method (Kaushik et al., 2019), or applying filter along the signal, the most commonly used pooling size is with filter size 2 and with stride 2. Two most famous types of pooling layers are Max. pooling (Saeedan et al., 2018) and average pooling (Lee et al., 2016), in the Max. method, the maximum value is selected to be the feature map for the pooling size window and average value is taken in average pooling method. Max pooling is used more likely than average pooling. Although, these methods are effective and simple, but they may produce omission of significant information and dilation of important details of the data. Other pooling method are proposed such as mixed and stochastic method, which used the probabilistic calculation to correct some weakness of the previous standard deterministic methods (Farabet et al., 2013). In this paper, A novel pooling method is proposed by using discrete Fourier transform (DFT), this method is used DFT technique to transform the data from spatial domain into frequency domain to preserve the most important information from the details coefficients, where the details information of the image is less significant, therefore it can be discarded to down sample the size of dimensions. Its effect will be great with advantage of reducing the eliminated details information as compared with other standard methods. After applying DFT, the most significant coefficients are cropped while less important details will be discarded then the data is reconstructed by applying inverse DFT, Different methods are proposed based on the scenario of using DFT and select the best coefficients for reconstructing the data (Yaroslavsky, 2014; Taigman et al., 2017).

Literature survey

Xiaym et al. used DCT to compress the image data, which reduced the amount of data through the intermediate networks of hidden layer. The work is accomplished by using small set of most important DCT coefficients rather than all pixels of image. These coefficients are used as input into two layer auto encoder instead. They showed that combining of DCT with deep learning increase model accuracy for image classification by exploiting the DCT compaction properties. The auto encoder was used to generate another a new vector, which reconstructed by decoder, then based on selection of best vector obtained by minimum values of error between original and reconstructed vector, thus, the auto encoder will achieve best result depending on best extracted features (Xioayi et al., 2014).

Xu and Hideki in 2019 was introduced a pooling method based on DCT technique, which is used to process output data of convolution layer. DCT was used to isolate the basic frequencies to isolate the features of input signal based on DCT compaction property. It can extract most significant information. They showed that the method is better than exiting works based on the obtained results with negligible time of consumption (Xu et al., 2019).

(Ulicny et al., 2020) used CNN model to learn the features, which are extracted from DCT coefficients. First, DCT was applied to compress the data as in image compression, based on compressing the harmonic networks, which have less information by discarding high frequency coefficients or other redundancy. They showed that, their work can compress the data with best performance (Ulicny et al., 2020).

Tavis William and Roberli is proposed pooling layer by using discrete wavelet transform (DWT), this method was used decomposed sub bands of the second level of wavelet transform as approximation coefficients which are used with (details coefficients is replaced by zero value) to reconstruct the data, based on the principle of approximation coefficients provides the approximation shape of the original data. Then best features of the original data are extracted by discarding less important information (Williams and Li, 2016).

Zeiler and Fegus proposed pooling method by elite activation, which selected from distribution multi-dimensional by activation pool size region of pooling. It is accomplished by evaluating probability function of each region, the selected probability will be normalized. The pooling features map will be sample from the distribution depending on determined probability. Different methods were used to elite the probabilities. The elited elements to be pooling feature map may be not the greater value (Zeiler and Fergus, 2013).

Kobayashi used distribution probabilities over activation for extract the feature data based on determining the standard deviation and mean statistics of distribution for Gaussian function for each pooling window. The basic idea work is to concise and summarize distribution of Gaussian function and aggregate activation function into average and standard deviation. This method is applied in stochastic pooling method, which used to determine the mapped features of the data (Kobayashi, 2019).

Hamad and Jasim are proposed a new pooling layer for CNN classification model by using Gaussian density function for determining wavelet transform

coefficients. These coefficients are used as filters weights for wavelet transform to get the approximation subband, which used later by different methods to evaluate the mapped features (Hamad alhussainy and Jasim, 2020).

In this paper, A novel pooling method is proposed by using DFT, this method is used DFT technique to transform the data from spatial domain into frequency domain to preserve the most important information from the details coefficients, where the details information of the image is less significant, therefore it can be discarded to down sample the size of dimensions without discarding significant information from the data. Its effect will be great with advantage of reducing the eliminated details information as compared with other standard methods, which may result in high reduction in the information, therefore this method can overcome most disadvantage of the previous method. This pooling method discarded less information based on isolating less significant component of the data by using fourier transform, while most other methods pool the features by down sample the data in spatial domain, which may produce high loss of data.

Proposed methodology

In this paper, A novel methods layer are proposed based on DFT named as FTM. DFT is isolate the frequency coefficients according its significant, then most important coefficients are cropped to be features of convolution layer output, therefore the amount of discarded information will be reduced and this can overcome disadvantage of standard Max. method or average method, which eliminated discarding of some basic details information of the data. The proposed methods are explained in the block diagram in Figure 1.

The original data (which always is output of convolution layer) are applied to DFT, then shifting and cropping pre-processing are perform to select the most important frequencies of the data, which are used to map the original data while less significant frequencies which is represented low important details are eliminated, the cropping of frequencies are performed based on the required stride of pool size, then inverse DFT is used to reconstruct the down sampled data, which will be input for upper layer as shown the figure. In the feedback, the gradient error is determined in reverse manner for the pooling procedure, first the DFT is applied to transform the data into frequency domain, then padding is used to resample the data to its original size according to the information from feed forward operation and the basic parameters, which are saved at cache, then inverse of DFT is applied to reconstruct the data to be input for lower layer as shown in Figure 1.

Discrete Fourier transform (DFT)

DFT can be applied to image to provide more details description about characteristics of the image and it is determined by Eq. (1) (Yaroslavsky, 2014).

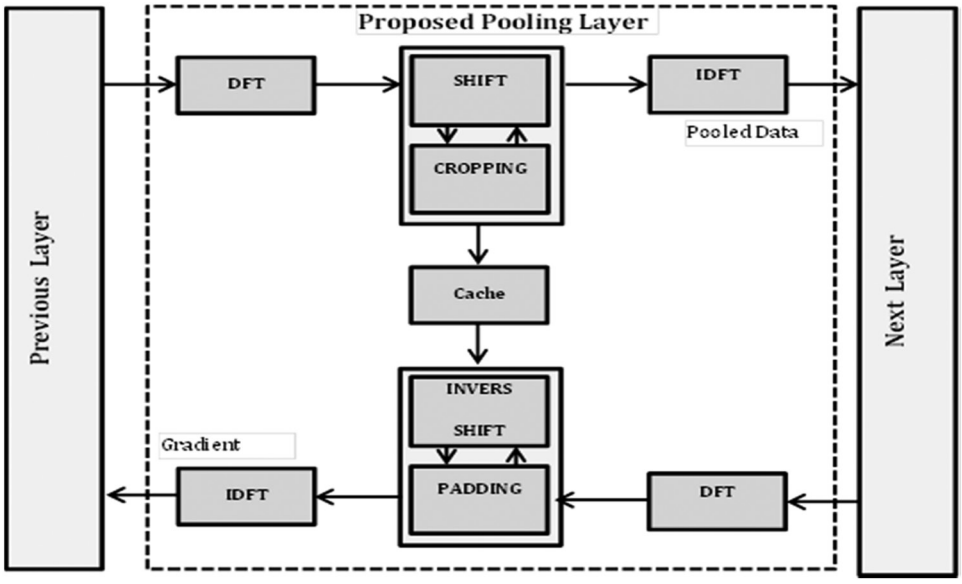


Figure 1: Proposed pooling layer.

$$F(u,v)=\frac{1}{N*M}\sum_{x=1}^{M-1}\sum_{y=1}^{N-1}f(x,y)\left[cos\left(2\pi\left(\frac{ux}{M}+\frac{vy}{M}\right)\right)\right]-j\left[sin\left(2\pi\left(\frac{ux}{M}+\frac{vy}{M}\right)\right)\right] \quad (1)$$

where $u = 0,1,\ldots..M-1$, also $v = 0,1,2\ldots..N-1$. As shown in the equation above the transform have real and imaginary part. The inverse transform is determined by Eq. (2).

$$f(x,y)=\frac{1}{N*M}\sum_{x=1}^{M-1}\sum_{y=1}^{N-1}F(u,v)\left[cos\left(2\pi\left(\frac{ux}{M}+\frac{vy}{M}\right)\right)\right]-j\left[sin\left(2\pi\left(\frac{ux}{M}+\frac{vy}{M}\right)\right)\right] \quad (2)$$

Algorithm I (FTM1)
Input =*N Input patch Data*;
Output=Y pooled Patch Data

Forward Propagation

```
Initialization
DH: input image Height;
DW: input image Width;
P pool size of pool; S : size of stride;
N input Patch data;
Determine size of pooled signal:
DH=size(N,3);                    %original size of image height
Dw=size(N,4);                    % original size of image width
HH= round((DH+1)/S)              % size of pooled image height
WW= round((DW+1)/S)              % size of pooled image width
While (n in range N)             % loop to number of data
  While (c in range C)           %loop to the number of channel
    Do { im=N(:,:,n,c)           %extract image from patch
      OUT1= Determine DFT(im);
      OUT2 =shift(OUT1)
      RE= OUT2 (1: HH,1:WW);     %Crop the basic coefficients
      Det=zero;                  %delete details coefficients
      OUT3=IDFT(RE);             %determine invers DFT
   Y(n,c,:,)= OUT3;
  End
End
```

Back Propagation

```
Initialization
M size of original Data from Memory Cache;
P pool size of pool; S : size of stride;
N input data from upper Layer;
Determine size of reconstructed signal:
DH=size(M, 3);                   %size of image hight
Dw=size(M, 4);                   % size of image width
HH= (DH*S)                       % size of reconstructed image hieght
WW= (DW*S)                       % size of reconstructed image width
While (n in range N)             % loop to number of data
  While (c in range C)           %loop to the number of channel
    Do { im=N(:,:,n,c)           %extract image from patch
      OUT1= Determine DFT(im);
      OUT2 =inversshift(OUT1)
      OUT2 (DH+1: HH,DW+1:WW)=0;     %padding the required coefficients
      OUT3=IDFT(OUT2);           %determine invers DFT
   Y(n,c,:,)= OUT3;
  End
End
End of Algorithm
```

Figure 2: Proposed first (FTM1) algorithm.

Proposed algorithms

Three pooling methods are proposed by using DFT named as FTM1, FTM2 and FTM3. DFT are used to isolate the different frequency components of the data, then by selecting most important frequencies of DFT transform, which represents most significant features of the data, then IDFT is performed to reconstruct original data. Different methods are proposed based on the procedure of processing of frequency components of the transform, these algorithms are explained in the following sections.

Algorithm 1 (FTM1)

This algorithm is extract features map by applying DFT, which is used to transform the output of convolution layer to frequency components, then shift

Algorithm 2 (FTM2)
Input =*N Input patch Data*;
Output=Y pooled Patch Data

Forward Propagation

```
Initialization
DH: input image Height;
DW: input image Width;
P pool size of pool; S : size of stride;
N input Patch data;
Determine size of pooled signal:
DH=size(N,3);                    %original size of image height
Dw=size(N,4);                    % original size of image width
HH= round((DH+1)/S)              % size of pooled image height
WW= round((DW+1)/S)              % size of pooled image width
While (n in range N)             % loop to number  of data
  While (c in range C)           %loop to the number of channel
    Do { im=N(:,:,n,c)           %extract  image from patch
      OUT1= Determine DFT(im);
      OUT2  =HPF(OUT1)
      OUT3 =LPF(OUT2)
      RE= OUT3 (1: HH,1:WW);     %Crop the basic coefficients
      Det=zero;                  %delete details coefficients
      OUT3=IDFT(RE);             %determine invers DFT
   Y(n,c,:,)= OUT3;
  End
End
Back Propagation
Initialization
M size of original Data from Memory Cache;
P pool size of pool; S : size of stride;
N input data from upper Layer;
Determine size of reconstructed signal:
DH=size(M, 3);                   %size of image hight
Dw=size(M, 4);                   % size of image width
HH= (DH*S)                       % size of reconstructed image hieght
WW= (DW*S)                       % size of reconstructed image width
While (n in range N)             % loop to number  of data
  While (c in range C)           %loop to the number of channel
    Do { im=N(:,:,n,c)           %extract image from patch
      OUT1= Determine DFT(im);
      OUT2  =inversshift(OUT1)
      OUT2 (DH+1: HH,DW+1:WW)=0;     %padding the required coefficients
      OUT3=IDFT(OUT2);               %determine invers DFT
   Y(n,c,:,)= OUT3;
  End
End
End of Algorithm
```

Figure 3: Proposed second (FTM2) algorithm.

operation is performed to select most significant coefficients for reconstructing the original data by inverse DFT. The details of this algorithm is explained in Figure 2.

Algorithm 2 (FTM2)

In this algorithm, features map are extracted by applying DFT, which is used to transform the output of

Algorithm 3 (FTM3)
Input =*N Input patch Data*;
Output=Y pooled Patch Data

Forward Propagation

Initialization
DH: input image Height;
DW: input image Width;
P pool size of pool; S : size of stride;
N input Patch data;
Determine size of pooled signal:
DH=size(N,3); *%original size of image height*
Dw=size(N,4); *% original size of image width*
HH= round((DH+1)/S) *% size of pooled image height*
WW= round((DW+1)/S) *% size of pooled image width*
While (n in range N) *% loop to number of data*
While (c in range C) *%loop to the number of channel*
Do { im=N(:,:,n,c) *%extract image from patch*
OUT1= Determine DFT(im);
M1=Max(OUT1); *% Max element in(OUT1)*
Pooled1= IDFT(OUT1);
OUT2 =HPF(OUT1)
OUT3 =LPF(OUT2)
RE= OUT3 (1: HH,1:WW); *%Crop the basic coefficients*
Det=zero; *%delete details coefficients*
OUT3=IDFT(RE);
Pooled2= OUT1 *%determine invers DFT*
M2=Max(OUT1); *% Max element in(OUT3)*
$\lambda 1 = M1/(M1 + M2);$
$\lambda 2 = M1/(M1 + M2);$
$Y(n,c) = \lambda 1 * Pooled1 + \lambda 2 * Pooled2;$
End
End

Back Propagation

Initialization
M size of original Data from Memory Cache;
P pool size of pool; S : size of stride;
N input data from upper Layer;
Determine size of reconstructed signal:
DH=size(M, 3); *%size of image hight*
Dw=size(M, 4); *% size of image width*
HH= (DH*S) *% size of reconstructed image hieght*
WW= (DW*S) *% size of reconstructed image width*
While (n in range N) *% loop to number of data*
While (c in range C) *%loop to the number of channel*
Do { im=N(:,:,n,c) *%extract image from patch*
OUT1= Determine DFT(im);
OUT2 =inversshift(OUT1)
OUT2 (DH+1: HH,DW+1:WW)=0; *%padding the required coefficients*
OUT3=IDFT(OUT2); *%determine invers DFT*
Y(n,c,:,)= OUT3;
End
End
End of Algorithm

Figure 4: Proposed third (FTM3) algorithm.

convolution layer to frequency components, then different filters (low pass and high pass filters) are used compress the coefficients into small set, then most significant coefficients are cropped to be used for reconstructing the original data by inverse DFT. The details is explained in Figure 3.

Algorithm 3 (FTM3)

This algorithm is constructed from mixing of the two previous algorithms by applying the two above methods in parallel manner, then the pooling data will be mixed of result of these methods based on determined maximum element for each pooling window to be used as mixing parameter. The description of this method is shown in Figure 4.

Experimental results and discussions

The performance of the proposed methods were evaluated by several tests, including first, the proposed methods were applied to the standard images, and the pooled images were extracted from them, then the original images were retrieved using only the pooled images. Then the retrieved images are compared with original images by using different measures such as SNR, correlation and SSIM to determine the similarity between original and reconstructed image. Also, the proposed pooling methods were used as pooling layers in CNN model for image classification, and it was tested by using two types of databases, which are (MNIST and CIFAR_10), and the results were compared with the results of standard methods (Average pooling and Max. pooling layers). The experiments were implemented using matlab2019a windows10, with CPU 2.7GHz, Intel cori_7, 2400 and RAM 16GB.

Results of test images

The proposed algorithms are applied to standard images to extract best feature to be mapped image with summarized dimensions size. The performance efficiency of the pooling method is evaluated by comparing the reconstructed image after pooling with the original image. The results are determined based on similarity measurements for standard “Lena”, “cameraman” and “Barbara” images. The original images size was (256*256), then these images were mapped to size of (128*128) as shown in Table 1, which showed development of metrics values for SSIM, correlation and SNR values. (FTM1) method achieved (24.1754), (20.29) and (29.38) in SNR for “Lena”, cameraman and Barbara images, which are good improvement compared with (23.28 dB, 20.14 dB and 23.69 dB) for Saeedan et al. (2018) method and (24.01 dB, 21.87 dB and 22.13 dB) for Lee et al. (2016) pooling method. Also in terms of correlation, the proposed method (FTM1) were achieved (0.9846), (0.9952) and

Table 1. Performance of proposed methods.

Images	Metrics	Saeedan et al. (2018)	Lee et al. (2016)	FTM1	FTM2	FTM3
Lena	SNR dB	23.28	24.01	24.1754	24.1603	24.16785
	SSIM	0.7882	0.7893	0.79396	0.77886	0.78641
	Correlation	0.9822	0.9833	0.9846	0.9695	0.97705
Cameraman	SNR	20.14	21.87	20.2935	20.2784	20.28595
	SSIM	0.7854	0.7867	0.7891	0.774	0.78155
	Correlation	0.9814	0.9843	0.9952	0.9801	0.98765
Barbara	SNR	23.69	22.13	27.336	27.3209	27.32845
	SSIM	0.7041	0.7065	0.7092	0.6941	0.70165
	Correlation	0.9622	0.9648	0.9612	0.9461	0.95365
Test image	SNR	28.98	28.78	29.38	29.3649	29.37245
	SSIM	0.7832	0.7841	0.7852	0.7701	0.77765
	Correlation	0.9913	0.9922	0.9965	0.9814	0.98895

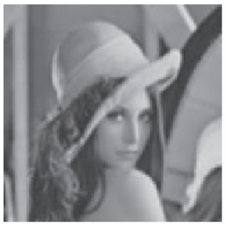
		
Original image (256*256)	Pooled image (128*128)	Reconstructed image (256*256)
		
Original image (256*256)	Pooled image (128*128)	Reconstructed image (256*256)
		
Original image (256*256)	Pooled image (128*128)	Reconstructed image (256*256)
		
Original image (256*256)	Pooled image (128*128)	Reconstructed image (256*256)

Figure 5: Original images, pooled images and reconstructed images by FTM1.

Table 2. Proposed methods results for MNIST dataset.

Method	Saeedan et al. (2018)	Lee et al. (2016)	FTM1	FTM2	FTM3
Accuracy (%)	98.80	98.72	99.95	99.84	99.96

(0.9612) verses (0.9822), (0.9814) and (0.9622) for Saeedan et al. (2018) method for "Lena", cameraman and Barbara images, respectively. Therefore, the pooled features represent most significant characteristics of original image. The details of the results are shown in Table 1, while quality of the extracted pooling image using (FTM1) are shown in Figure 5. It is clear that the reconstructed image look like the original image due to best extraction of features with discarding of less important data through pooling map.

Results of image classification

The proposed pooling methods were used as pooling layer in CNN model for image classification. This model was tested by using two types of databases, which are (MNIST and CIFAR_10).

Results of MNIST classification

The results of MNIST classification by using the proposed pooling methods are shown in Table 2. It is clear that the proposed methods achieved higher accuracy than standard methods (Average and Max.) methods. The proposed method satisfied (Acc. 99.96%), which obtained by reduction of discarding significant frequencies details component of the image. Figure 6 shows comparison between the satisfied accuracy by the proposed methods verses the results of standard methods while sensitivity results is shown in Figure 7 and precision comparison for the proposed methods is described in Figure 8.

The training of the model steps for (FTM1) method is shown in Figure 9. The accuracy is upgraded to greater than (97% accuracy) in 2 epochs iteration only, also the loss function is dropped to less than 0.21, which prove that the proposed pooling method is extracted best characteristics of the input data. The confusion matrix description of the results for matching is listed in Table 3.

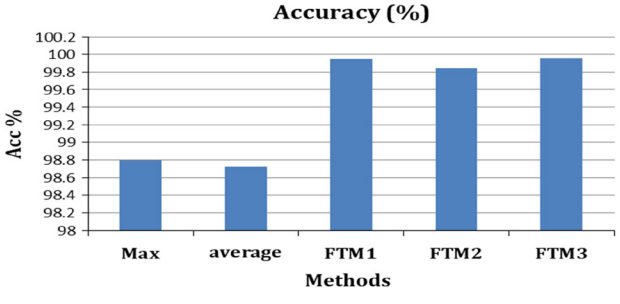


Figure 6: Accuracy of MNIST classification for proposed method.

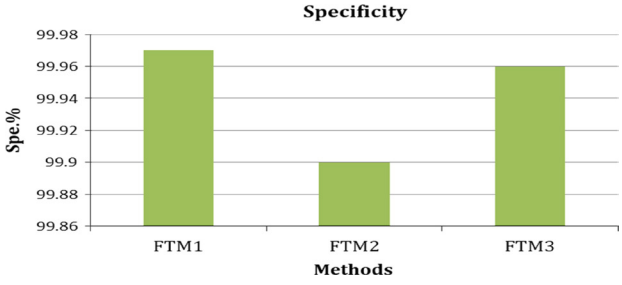


Figure 7: Sensitivity results for proposed method for MNIST classification.

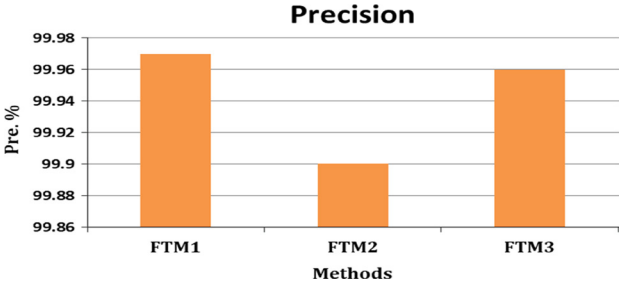


Figure 8: Specificity results for proposed method for MNIST classification.

CIFAR_10 dataset results

The proposed methods also used for classification of second dataset, which is CIFAR10 with size of (60000*32*32*3), where 3 represents (RGB) image. The classification model trained using 50,000 images, while other 1,000 images used to test the system efficiency. The model has been trained up to 10 epochs by 390 iteration per epoch with batch size 128 and the initial learning rate (0.01). The obtained results of the proposed methods were compared with results of (Max. method and Average method), which are shown in Table 4. The proposed methods achieved better results compared to the results of the standard methods, as their accuracy rate were (73.88), (73.82), (73.76) (FTM1, FTM 2 and FTM 3) methods, respectively, while it were (72.59%), (72.4%) for the standard (Saeedan et al., 2018; Lee et al., 2016) methods, respectively. Superior classification results were achieved with FTM1 method due to its best isolation of important frequency and therefore eliminate less important frequencies, depending on the shift process of frequencies which leads to the loss of the least possible information from the original

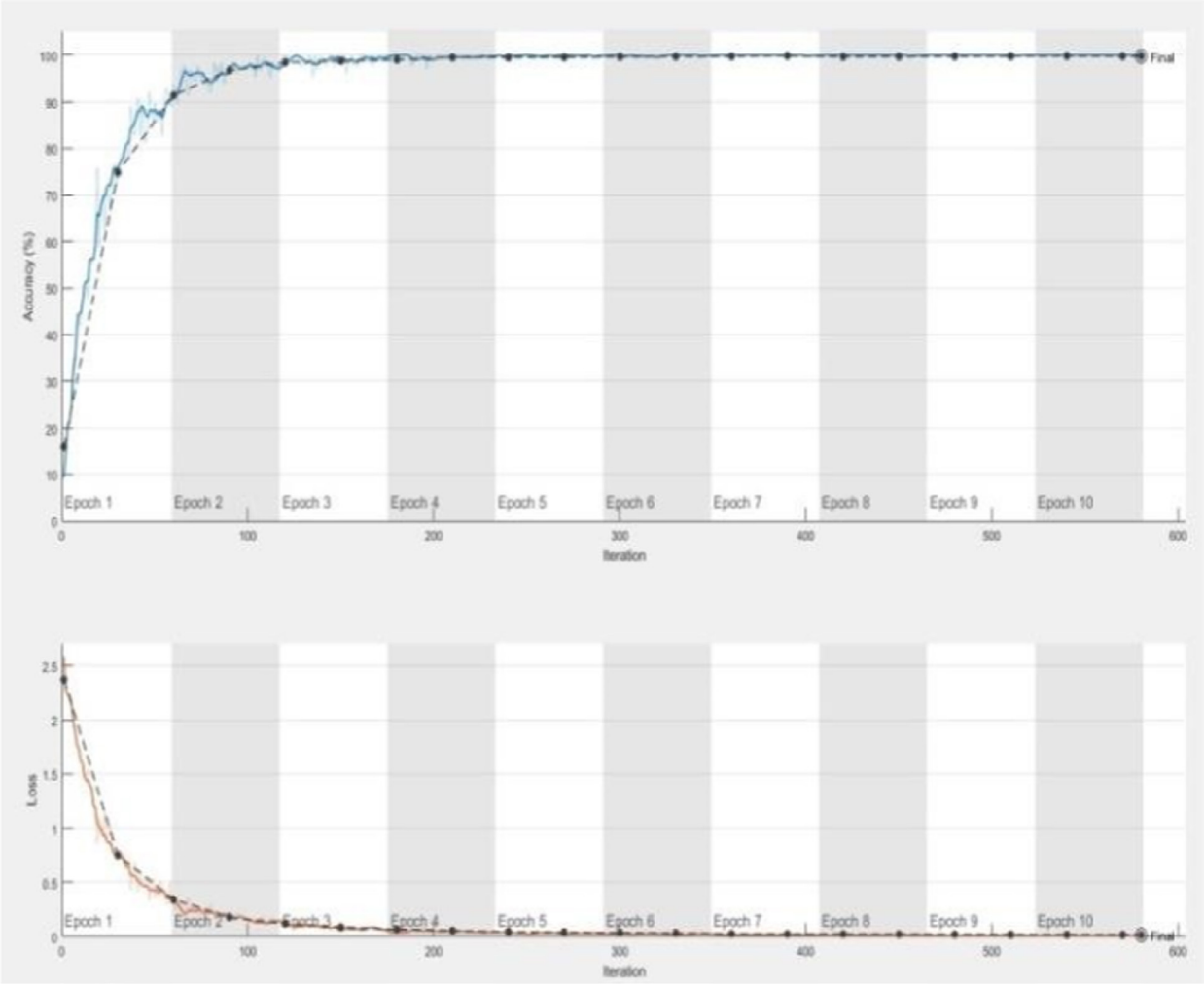


Figure 9: Training steps for MNIST classification by (FTM1) method.

data and recovery of the best characteristics of the information. The performance descriptions are shown in Figures 10, 11 and 12, for accuracy, specificity and precision respectively. The matching between classification results and actual results are shown in Table 5. As shown in this table, there is high matching between the predicted and actual classes, also the it achieved high rate of sensitivity and specificity for all classes, which prove that this method can be applied for different types of images.

Conclusions

Convolution layer considered the most important layer in deep learning due to its ability to extract basic features of the data through the network, therefore it is used in most D.L models such as CNN model, but because this layer is used more number of filters and different numbers of channels, the size of the data will be increased, which may need to huge volume of computation and lead to reduce the model efficiency. Thus, different methods are used to solve this problem, one of them is using pooling layer. Also pooling layer may discard important information from the data, therefore best features of data must be selected to reduce the loosing of information. In this paper, A new methods layer are proposed based on DFT. DFT isolated the frequency coefficients according to its significant, then most important coefficients are cropped to be features of convolution layer output, therefore the amount of discarded information will be reduced and this can overcome most disadvantage of standard Max. method or average method, which were eliminated some of basic details information of

Table 3. Confusion matrix for (FTM1) method for MNIST Classification.

	Target class										
	Class 0	**Class 1**	**Class 2**	**Class 3**	**Class 4**	**Class 5**	**Class 6**	**Class 7**	**Class 8**	**Class 9**	**Sensitivity**
Output class											
Class 0	250	0	0	0	0	0	0	0	0	0	100
Class 1	0	250 10%	0	0	0	0	0	0	0	0	100 %
Class 2	0	0	250 10%	0	0	0	0	0	0	0	100 %
Class 3	0	0	0	250 10%	0	0	0	0	0	0	100 %
Class 4	0	0	0	0	250 10%	0	0	0	0	0	100 %
Class 5	0	0	0	0	0	250 10%	0	0	0	0	100 %
Class 6	0	0	0	0	0	0	249 10%	0	0	0	100 %
Class 7	0	0	0	0	0	0	0	250 10%	0	0	100 %
Class 8	0	0	0	0	0	0	1	0	250 10%	0	99.9 6%
Class 9	0	0	0	0	0	0	0	0	0	250 10%	100 %
	100 %	100 %	100 %	100 %	100 %	100 %	100 %	99.9 6%	100 %	100 %	100 %

Table 4. Accuracy results for CIFAR10 dataset classification.

Method	Saeedan et al. (2018)	Lee et al. (2016)	RFT1	RFT2	RFT3
Accuracy (%)	72.59	72.4	73.88	73.82	73.76

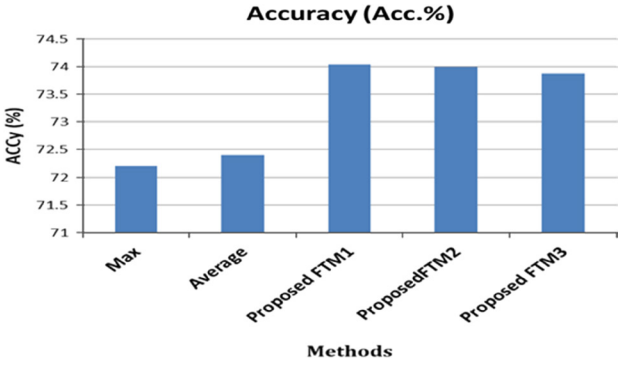


Figure 10: Accuracy results for proposed method for CIFAR 10 classification.

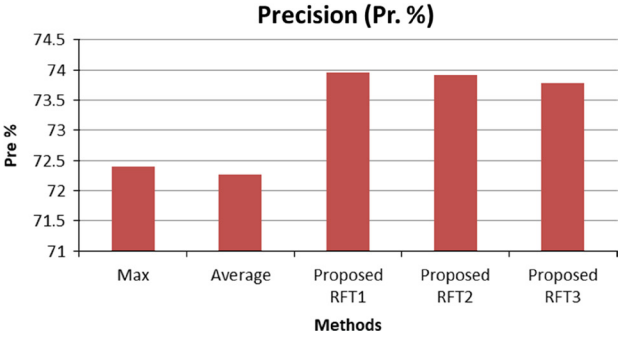


Figure 12: Precision results for proposed method for CIFAR 10 classification.

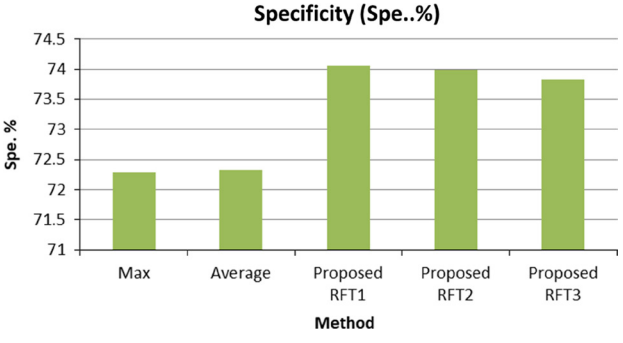


Figure 11: Specificity results for proposed method for CIFAR 10 classification.

the data, therefore, this pooling method discarded less information based on isolating less significant component of the data by using Fourier transform, while most other methods pool the features by down sample the data in spatial domain, which may produce high loss of data.

The proposed methods were applied to the standard images, and the pooled images were extracted from them, then the original images were retrieved using only the pooled images. Then the retrieved images are compared with original images by using different measures such as MSE, PSNR and SSIM to determine the similarity between original and reconstructed image. (FTM1) achieved (32.868 dB), (24.19 dB) and (26.58 dB) in SNR for "Lena", cameraman and Barbara images, verses (24.28 dB, 23.14 dB and 21.69 dB), respectively, for [Saeedan et al. (2018) method] and (24.01 dB, 22.87 dB and 22.13dB), respectively, for [Lee et al. (2016)] pooling method. The proposed layers used for image classification by using two different datasets, which are MNIST and CIFAR_10. Specifications results of classification are evaluated using different metrics like accuracy, specificity and precision. The classification of MNIST achieved (Acc: 99.96%) more than (Saeedan et al. (2018) 98.8%), (Lee et al. (2016) 98.72%). For classification of CIFAR_10, the proposed method satisfied (73.88%) better than (Saeedan et al. (2018) method 72.59%) and (Lee et al. (2016) method 72.4%). The results proved that the proposed methods outperformed standard methods, thus it can be used for deep learning application. The most limitation of this method is that it required more calculation cost due to more computation operation, which may lead to increase required time for training, therefore batch normalization layer may be used in future depending on the third dimension of the data to reduce the execution time Figures 6 and 7.

Table 5. Confusion matrix for (FTM1) for CIFAR_10 classification.

	Target class										
	airplane	*auto mobi*	bird	cat	deer	dog	frog	hour	ship	truc	Sens
Output class											
airpla	796	29	68	32	25	13	10	22	57	3	73
autom	10	829	6	6	4	2	3	4	15	7	86
bird	44	10	599	62	69	42	37	32	13	8	66
cat	18	3	61	533	36	16	57	43	10	9	56
deer	24	3	75	60	708	38	26	55	5	1	71
dog	4	4	109	201	55	69	22	97	4	3	58
frog	9	9	46	66	54	13	83	5	7	1	78
hours	7	1	16	12	38	25	3	72	2	4	86
ship	48	36	9	14	9	6	4	6	86	2	82
truck	38	75	8	13	4	6	2	8	18	80	83
specifi	79.8	82.3	59.8	53.3	71.6	69	83	73	86	82	74

Literature Cited

Farabet, C., Couprie, C., Najman, L. and LeCun, Y. 2013. Learning hierarchical features for scene labeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35(8): 1915–1929.

Hamad alhussainy, A. M. and Jasim, A. D. 2020. A novel pooling layer based on gaussian function with wavelet transform. *Indonesian Journal of Electrical Engineering and Computer Science* 20(3): 1289–1298.

Kaushik, M., Gupta, S. H. and Balyan, V. 2019. Evaluating threshold distance by using eigen values and analyzing its impact on the performance of WBAN. *2019 6th International Conference on Signal Processing and Integrated*.

Kobayashi, T. 2019. Global feature guided local pooling. *IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, Korea (South), pp. 3365–3374.

Krizhevsky, A. and Hinton, G. E. 2017. Using very deep autoencoders for content-based image retrieval. *ESANN*, Belgium, pp. 1–7.

Lee, C.-Y., Gallagher, P. W. and Tu, Z. 2016. Generalizing pooling functions in convolutional neural networks: Mixed, gated, and tree. *In AISTATS* 51(17): 464–472.

Ren, S., He, K., Girshick, R. and Sun, J. 2017. Faster RCNN: towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39(6): 1137–1149.

Saeedan, F., Weber, N., Goesele, M. and Roth, S. 2018. Detail-preserving pooling in deep networks. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, pp. 9108–9116.

Taigman, Y., Yang, M., Ranzato, M. and Wolf, L. 2017. DeepFace: closing the gap to human-level performance in face verification. *IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, pp. 1701–1708.

Ulicny, M., Krylov, V. A. and Dahyot, R. 2020. Harmonic convolutional networks based on discrete cosine transform. arXiv:2001.06570v1.

Vinyals, O., Toshev, A., Bengio, S. and Erhan, D. 2015. Show and tell: a neural image caption generator. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3156–3164.

Williams, T. and Li, R. 2016. Advanced image classification using wavelets and convolutional neural networks. *15th IEEE International Conference on Machine Learning and Applications*, pp. 233–239.

Xioayi, Z., Xiangmin, X., Chunmei, Q. and Xiaofen, X. 2014. High speed deep networks based on discrete cosine transformation. *Image Processing (ICIP), 2014 IEEE International Conference on image processing*, pp. 5921–5925.

Xu, Y. and Hideki, N. 2019. *DCT Based Information-Preserving Pooling for Deep Neural Networks.* (ICIP), Taipei, Taiwan.

Yaroslavsky, L. P. 2014. Fast Transforms in Image Processing: Compression, Restoration, and Resampling, Hindawi Publishing Corporation Advances in Electrical Engineering Volume.

Zeiler, M. and Fergus, R. 2013. Stochastic pooling for regularization of deep convolutional neural Networks. In *ICLR*.