Received: May 24, 2023. Revised: June 3, 2023.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Robust Object Recognition with Deep Learning on a Variety of Datasets

Samar Antar[1] **Hussein Karam Hussein[1]*** **Mohammad H. Abdel-rahman[1]** **Fayed Ghaleb[1]**

[1]*Ain Shams University, Faculty of Science, Mathematics & Computer Science Department, Abbassia 11566, Cairo, Egypt*
* Corresponding author's Email: h_karam@sci.asu.edu.eg

Abstract: For many intelligent applications, object recognition is a critical issue. Developing an effective feature extraction method is one of the most difficult issues in object recognition. For this process, a variety of algorithms were developed, including self-organizing maps (SOMs), support vector machines (SVM), principal component analysis (PCA), and modern deep learning techniques, particularly convolutional neural networks (CNNs). In CNNs, the two layers that most significantly contribute to the network bottleneck are the convolution layer and the fully connected layer. While the later layer is memory-intensive, the convolution one is computationally expensive. So, optimizing these two layers is crucial for executing efficient convolution operations. The primary objective of this paper is a detailed discussion of two approaches for optimizing the CNNs architecture. In the first approach, CNNs were enhanced using the self-organizing maps (SOMs) topology space in the convolution layer and the KNN classifier instead of the conventional fully connected layer. The second approach employed the KNN classifier in the fully connected layer and used an improved SOMs technique called "cyclic convolution SOMs" rather than convolution structures to process CNNs more quickly. The efficiency of the proposed approaches has been evaluated on four wide benchmark datasets: AHDBase for Arabic digits, MNIST for English digits, CMU-PIE for faces, and CIFAR-10 for objects. The experiment using these datasets provided the following findings in comparison to other approaches (e.g., standard CNN, CSOMs, LSTMs, SVM, SOMs, PCA, and cyclic SOM): the first approach produced results of 97.7%, 98.2%, 98.51%, and 93.8%; the second approach produced results of 96.57%, 95.4%, 97%, and 89.23%. Our results indicate that when applied to a variety of datasets, the proposed methods offer promising outcomes with higher accuracy than the existing ones.

Keywords: Object recognition, Self-organizing maps, Convolutional neural networks, Learning rate, Deep learning, Network optimization, Feature extraction.

1. Introduction

Object recognition is a subject that is becoming more important in both industry and academia. For a robust object recognition task, a number of related computer vision tasks are required, including: (1) pre-processing, (2) features segmentation and extraction, and (3) classification. When we refer to the "method of feature extraction", we actually mean the "methodology of dimensional reduction" [1]. The two primary strategies used for dimensionality reduction are feature extraction (FE) and feature selection (FS). Since data is created continually at an increasing rate, FS is viewed as a key approach because it may successfully minimize serious dimensionality difficulties, such as removing unnecessary data. Additionally, finding the most distinct, enlightening, and condensed collection of characteristics to improve the efficiency of data processing and storage is a concern addressed by FE. There are two types of FS algorithms: nonlinear and linear [2]. However, the optimum dimensionality reduction methods based on feature extraction are isometric mapping [3], principal component analysis, linear discriminant analysis [1], clustering methods [4], and more recently deep learning (DL), in particular convolutional neural networks (CNNs) [5, 6]. CNNs is one of the most significant networks in the domain of DL, which is applied successfully in a variety of research areas, including, internet of things

Received: May 24, 2023. Revised: June 3, 2023.

(IoT) [7], Twitter Sentiment Analysis [8], sign language recognition [9], Speech Recognition [10], medical imaging [11-13], object detection [14] and more. Convolution and fully connected (FC) layers in CNNs are the two layers that most significantly affect network performance. While the later layer (FC) is memory-intensive, the convolutional one is computationally expensive, consequently, limits its practical implementation. Therefore, speeding up these two layers is crucial for executing efficient and quick convolution operations to enhance the computational speed of CNNs. This paper addressed these issues via two approaches for CNN's architecture optimization. This paper is organized as follows. The problem statement, objectives, and prior research on object recognition application is highlighted in section 2. Section 3 summarizes the background, the basic terminology, and notions that will be used throughout this paper. The methodology, which includes the two methods we have proposed for CNNs architecture optimization, is introduced in section 4. The description of the databases used is demonstrated in section 5. Some experimental results with discussion and analysis of our proposed framework compared with the existing one are discussed in section 6. Section 7 concludes this paper with some directions for future work.

2. Problem statement, objectives and related work

The main advantage of CNNs is that it automatically detects relevant features without any human supervision, which has made it extensively applied in a range of different fields. However, CNNs have millions of parameters, often requires a lot of data for learning, and training CNNs is laborious and time-consuming. Before training, the model can be configured with different parameters (e.g., weights, biases, number of layers, etc.) and a group of parameters called hyper-parameters which are associated with a convolution layer (e.g., processing units (neurons), filter size, activation function, stride, zero-padding, learning rate, etc.) [6]. Hyper-parameters are constants, which need their values to be predefined before the models could be constructed. According to [15] these hyper-parameters affect the overall performance of the network. The stochastic gradient descent (SGD) optimization algorithm [16] and its variants are the most widely used algorithms for the training of CNNs. The goal of any optimization issue when training and fine-tuning neural networks is to determine the optimum weights and biases that will return the lowest cost, also known as loss or error, with a higher cost indicating a less efficient network. The mean square error (MSE), which is determined as follows, is the most used loss function (L):

$$MSE(L) = \sum \frac{1}{2}(target - output)^2 \tag{1}$$

Now, our goal is to identify the weights that contributed the most to the loss function (L) and discover strategies to modify them such that the loss is reduced. To do this, we employ optimization techniques such as SGD, where weights (w) are iteratively updated using a loss function (L) as follows:

$$W_{ij^t} = W_{ij^{t-1}} - \Delta W_{ij^t},\ \Delta W_{ij^t} = \eta \times \frac{\partial L}{\partial W_{ij}} \tag{2}$$

Where, the final weight in the current training epoch is denoted by W_{ij^t} ,while the weight in the preceding $(t - 1)$ training epoch is denoted$W_{ij^{t-1}}$, and η represents the learning rate (LR). The amount that the weights are updated during training is referred to as the "learning rate (LR)". Updating the LR is challenging and depends on the dataset. Generally, this parameter is found on a trial-and-error basis, and scholars sometimes set it to be constant (e.g. a value of 0.01 in case of [17]). A little change in the learning rate affects convergence and the learning speed as well as the overall performance of the network. Thus, there is a need for an optimization strategy to adapt the learning rate to make the learning process faster and more efficient. For achieving this goal, this paper used the CNNs architectural design and presents an optimization technique for object recognition problem referred to as "cyclic convolution SOMs". On the other hand, convolutional operations are a vital step in feature extraction and are among the most computationally intensive in CNNs. Several algorithms are designed to address this issue [16, 18-21]. For instance, the authors of [21] present a novel approach to self-organizing maps (SOMs) using CNNs. The proposed method, called Convolutional SOMs (CSOMs), is based on the idea of using convolutional layers to learn the topology of the SOMs. The authors stated that CSOMs can be used to learn the topology of a SOMs in a more efficient and accurate manner than traditional SOMs. The high computation times and usage of various parameters throughout the learning process are drawbacks of this technique. This paper addressed this problem and proposed a new improvement using two approaches. The type of classifier used at the end of recognition and the feature extraction technique used by SOMs make up

Received: May 24, 2023. Revised: June 3, 2023.

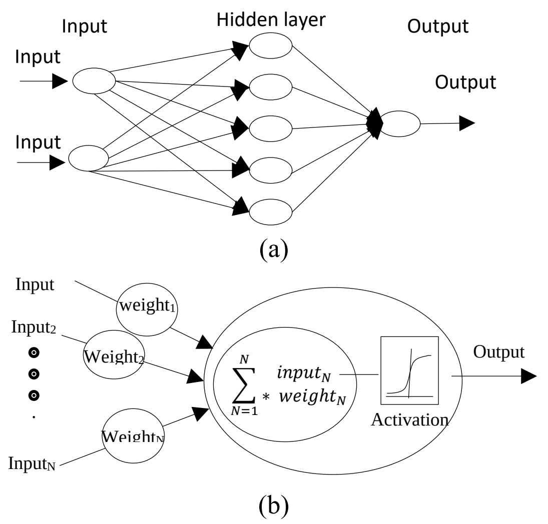


Figure. 1 (a) ANN structure, and (b) ANN components

the majority of our contribution. The concept behind feature extraction is the use of a cyclic learning rate (CLR) [22] in the learning phase of SOMs to speed up their learning process, which we refer to as a "cyclic convolution SOMs", as opposed to the convolution structures used in the CNNs architectural design. The idea is to use an optimization technique for accelerating the learning process and eliminates the need to experimentally find the best values and schedule for the learning rates by redesigning the CNN's architecture using cyclical learning rates (CLR) in the training phase of the SOMs that we previously developed [23]. Then it uses the KNN classifier rather than the conventional MLP used in CNNs. Four wide benchmark datasets illustrate the efficiency of the proposed techniques in compared with other techniques including [21].

3. Foundations and basic terminology

3.1 Neural networks

One of the most powerful technological advancements is the usage of neural networks (NNs), which are employed in a number of applications, including object recognition. Biological and artificial neural networks (ANNs) are the two different types of NNs. The fundamental building blocks of an ANNs are a collection of linked processing units referred to as neurons that collaborate to address certain issues as indicated in Fig. 1 (a)-(b) respectively.

Deep learning (DL), is the use of ANNs with several hidden layers for learning tasks. Deep belief neural networks (DBNs), long short-term memory networks (LSTMs), recurrent neural networks (RNNs), restricted Boltzmann machines (RBMs), and CNNs are the most well-known varieties of DL networks [24]. The first term in DL, “deep,” refers to the number of layers used in the data transformation. The second term, “learning,” describes the kind of learning algorithm that is being employed. Supervised learning and unsupervised learning are the two main paradigms for learning. In supervised learning, input data that has been labelled for a certain output is used in the training process. Unsupervised learning differs from supervised learning in that there are no labels in the training set. The most popular algorithm in the unsupervised learning category is used by SOMs. The object recognition application of the unsupervised learning technique using SOMs applicable to DL, in particular the CNNs architectural design is the main objective of this study.

3.2 Self-organizing maps (SOMs)

SOMs is one of the most well-known unsupervised learning neural networks [25], which is effectively applied in different research areas (e.g., [26]). Its main goal is to reduce the complexity of a high-dimensional, discrete input space to a lower-dimensional, often two- dimensional, discrete output space while maintaining the connections (or topology) within the data but not the actual distances. Such a low-dimensional representation is called a feature map, hence the word “maps” in the name. Three common processes, described as follows, are involved in the creation of SOMs: competition, cooperation and adaptation process. In a competition, the output nodes (neurons) in SOMs compete with one another to best reflect a certain input sample. The effectiveness of representation is assessed by comparing an input vector with the weight vector of each output node using a discriminant function. The winning node, or best matching unit (BMU), is the one whose connection weights are most similar to those of the input sample. The best-matching unit (BMU) on the map is chosen as the winner using a variety of different functions. SOMs use the Euclidean distance to measure the similarity between the inputs and the features maps. The authors of [23] employed cyclical learning rates (CLR) in the training phase of the SOMs instead of the conventional Euclidean distance in order to achieve great performance and a higher recognition rate. Regarding the cooperation process, a neighbourhood of collaborating nodes is spatially located by the winning node. These nodes cooperate to learn from the same input because they have similar properties. In the adaptation process, the winner's and its surrounding units' weight vectors on the map are modified to favour greater values of their

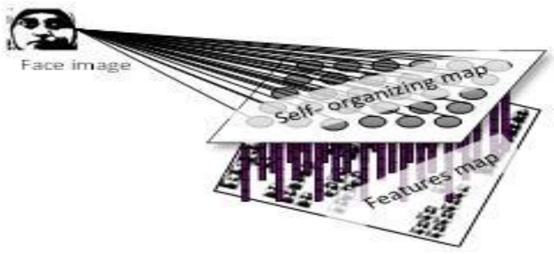


Figure. 2 Self-organizing maps work mechanism

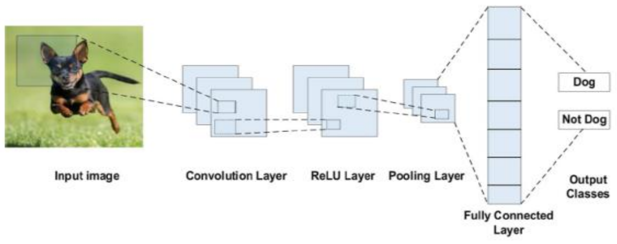


Figure. 3 An example of CNNs architecture [24]

discriminant functions. Through this learning process, the relevant nodes become more comparable to the input sample. As a result, nodes with a strong response to a certain set of input data will be more likely to respond to input data that is similar in the future. The traditional SOMs algorithm is provided as follows: consider I as an input vector of one sample. For measuring the propinquity, the Euclidian distance computed between the input and all the neurons weights vectors w_u in the map. The neuron which has more propinquity fromthe input, or what's called the winner neuron, w_c is calculated by the minimum distance from the input I and all the neurons in the SOMs map. The function which finding the winner neuron can be as follows:

$$||I - w_c|| = \min_u(||I - w_u||) \quad (3)$$

After deciding the winner, the SOMs neuron's weights are updated as follows:

$$w_u(t+1) = w_u(t) + h_{cu}(t)[I(t) - w_u(t)] \quad (4)$$

Where

$$h_{cu}(t) = \eta(t) \times \exp\left[\frac{||r_c - r_u||}{2\,\sigma^2(t)}\right] \quad (5)$$

is the neighborhood kernel of the winner "c" at time t, $\eta(t)$ expresses the learning rate, and $\sigma^2(t)$ expresses a factor used to control the neighborhood kernel. The term $||r_c\text{-}r_u||$ calculates the distance between the winner weights "c" and the other neuron weights "u". After training, the neuron is organized into a feature map of two dimensions. As a result of using the SOMs, related features from the inputs are transferred into neighboring spots in the feature map. Fig. 2 shows a simplified illustration of the SOMs work on face recognition.

3.3 Convolutional neural networks (CNNs)

CNN's success is largely due to its in-built capacity to automatically extract features from input data without operator intervention [14]. With CNN's popularity, companies like Google, Microsoft, AT&T, NEC, and Facebook have established active research groups to investigate new architectures of CNNs that use numerous convolutional layers like the AlexNet (8 layers) [27], GoogLeNet series (22 layers) [28], VGGNets (16 layers) [29], ResNet (152 layers) [30], and others for a variety of tasks, including facial recognition and image search. A general representation of CNNs architecture for image classification is generally composed of four layers: input, convolution, pooling (sub-sampling) and fully connected as demonstrated in Fig. 3 (as presented by [24]).

A convolutional layer is applied by taking an image as an input matrix of pixels and then applying learnable filters (or kernels) of a fixed size i.e., $m \times m$ to each $m \times m$ block of the input matrix. A kernel convolves with the images using a specific set of weights by multiplying its elements with the corresponding elements of the receptive field. Receptive fields are the area of the visual field where a single neuron is activated in response to a stimulus. These multiplications are all summed up and the process is repeated for every location in the input volume. A group of parameters called hyper-parameters are associated with a convolution layer: filter size, stride and zero-padding [6]. When working with a convolution neural network, the following are some essential functions:

$$O = \left(\frac{W - K + 2 * p}{S}\right) + 1, \quad P = \frac{K-1}{2} \quad (6)$$

Where O is the (height \ width) of the output layer, W is the (height \ width) of the image size, K is the (height \ width) of the filter, P is the zero padding, and S is the stride. These are also the functions we have for the convolutional layer:

$$Z_j = \sum X_i K_{ij} + B_j, \quad (7)$$

$$A_j = F(Z_j) \quad (8)$$

Where, Z_j represents the output from the convolution operation, X_i denotes the input to the convolutional layer, K_{ij} is the convolution kernel, B_j

is the additive bias, like the intercept added in a linear equation, A_j is the output feature map of the convolutional layer, and $F(.)$ is an activation function. Convolution produces “feature maps,” which are collections of several different features. As a result, a pooling layer (e.g., Max pooling) is used to reduce the dimensionality of each feature map but hold the most critical data. The system has been doing computations linearly up until the convolutional layer. The selection of an appropriate activation function, such as sigmoid, the rectified linear unit (ReLU), and variants of the ReLU, is used to introduce non-linearity in the system. For instance, the purpose of ReLU is to replace negative activations by 0. Each CNNs architecture has a fully connected layer at the end. Inside this layer, each neuron is connected to all the neurons of the previous layer, the so-called FC approach. It is used as the CNNs classifier.

4. Methodology

This section presents the CNNs optimization methodology, including the two approaches we have proposed that make use of the SOMs method.

4.1 Optimized convolution SOMs

Four basic operations listed in Table 1 are involved in the creation of typical CNNs. The first three operations can be repeated several times for getting the final feature vector, which is used in the fourth operation for getting the final recognition. A commonly used type of CNNs, which is similar to the multi-layer perceptron (MLP), consists of numerous convolution layers preceding sub-sampling (pooling) layers, while the ending layers are the fully connected (FC) layers. The fourth operation represents the FC layer which is located at the end of each CNN architecture. It is a type of feed-forward ANN. The input of the FC layer comes from the last pooling or convolutional layer. This input is in the form of a vector, which is created from the feature maps after flattening; see Fig. 4 (as presented by [24]).

Table 1. The basic operation in any CNNs

Operation	Equation
Convolution	$C^l = AC^{l-1} * W^l$
Max pooling	$AC^l_{xy} = max_{i=0,..,s} AC^{l-1}_{(x+i)(y+i)}$
Relu	$ReLU(zi) = \text{Max}(0, zi)$
FC Layer	$C^l = W^l AC^{l-1}$
Where C^l Convolution output of layer l, AC^l_{xy} The output of the max pooling layer, W is a learnable parameter, (x,y) refers to the pixel location number, and I refer to the number of the output from every kernel.	

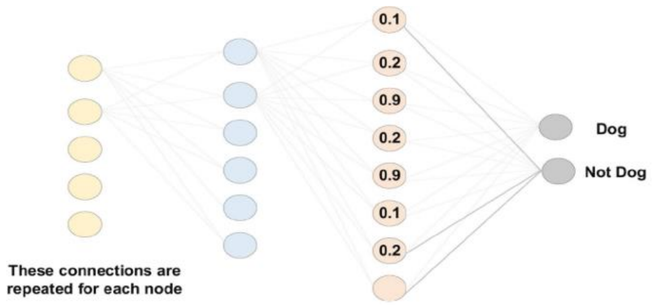


Figure. 4 An illustration of the FC layer [24]

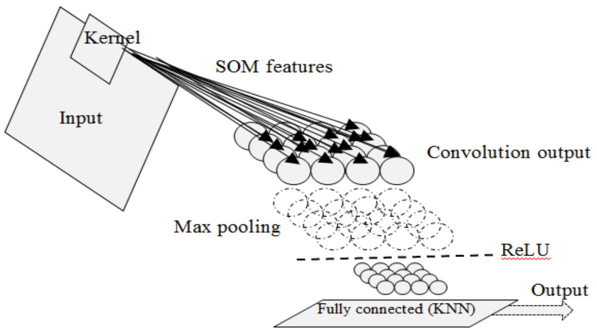


Figure. 5 A simple representation for the structure of the optimized convolution SOMs

In the optimized convolution SOMs approach proposed in this paper, and based on the regular SOMs' algorithm previously discussed, we used several small SOM with size 3×3 referred to as “SOM kernels” for getting the final feature vector. When we apply one kernel of them, we calculate the distance between the kernel neuron and the input slices pixel during the convolution process for finding the best matching unit or neuron (BMU). A small feature map with a size of 3×3 was produced after this operation. Then the average of this feature map is passed into the convolution output. The output image from the convolution layer for every small SOM or kernel is passed to the max-pooling layer and the activation function called Relu. For updating the weight; we use a regular updating function for the SOMs kernels with a fixed learning rate as in Eq. 4. The previous operations were repeated after adding max-pooling and the Relu layer four times before reaching the fully connected layer. In the final fully connected layer, we employed the K-nearest neighbour (KNN) for classification [31] instead of the basic method of the conventional neural network in the typical CNNs. An illustrated example for the optimized convolution SOMs' structure, with one convolution layer, is depicted in Fig. 5.

4.2 Cyclic convolution SOMs

Learning rate is a critical hyper-parameter that should be carefully considered during the training

Table 2. The cyclic learning equations

$\eta_t = \eta_{min} + (\eta_{max} - \eta_{Min}) * \max(0, 1-x)$

Where x is defined as

$x = \left\| \frac{iteration}{step\ size} - 2\ cycle + 1 \right\|$

The term "cycle" can be calculated as follows:

$cycle = floor(1 + \frac{iteration}{2\ step\ size})$

Where the decreasing learning rate was

$\eta_{i+1} = \eta_i - (\frac{\eta_{max}}{n})$

Where i is the previous iteration number

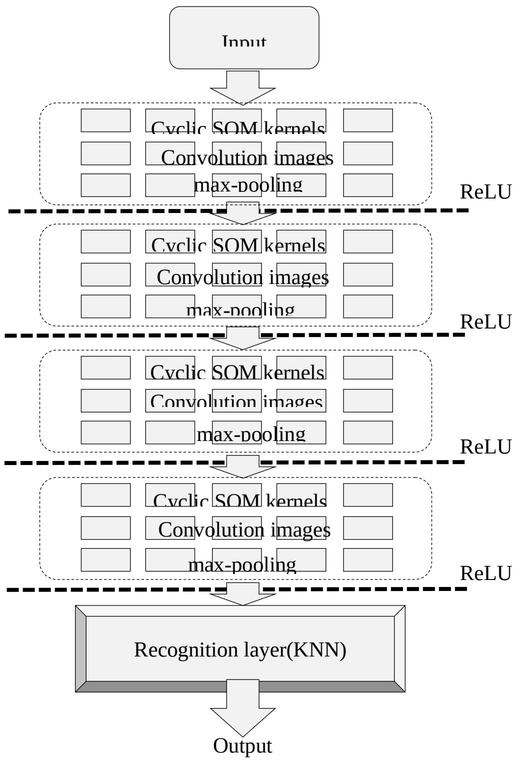


Figure. 6 An illustration of cyclic convolution SOMs method

Table 3. The cyclic convolution SOMs algorithm steps

1. Determine the number of kernels (small SOMs) for every Convolution layer and use one by one.
2. Determine the feature map for the convolved image based on the Cyclic SOMs.
3. In the convolution output, insert the average of the kernels' Feature map for every pixel.
4. Perform the max-pooling operation.
5. Perform the Relu operation.
6. The steps from 2 to 6 are repeated four-times.
7. All the previous operation steps from 2 to 7 are repeated several times to get the best accuracy.
8. Use the cyclic learning algorithm to update the learning rate for updating kernel weights.
9. Concatenate the last features and pass them for the final recognition using the KNN.
10. Determine the recognition rate.

Figure. 7 A sample image of a person under five illuminations in two poses

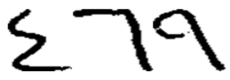


Figure. 8 Sample Images for AHDBASE databases

process. The key elements of the cyclic convolution SOMs approach are based on the type of learning algorithm proposed in our previous work [23]. The idea is to find the best matching neuron between the input and the feature map during the convolution process using mean absolute difference instead of traditional Euclidean distance and using cyclical learning rates (CLR) in the training phase of the SOMs. Instead of setting the learning rate to fixed values, the kernel weights cyclically update within reasonable boundary values, as illustrated in Table 2.

As demonstrated in Fig. 6, the proposed cyclic convolution SOMs method can perform in four main convolution layers and every layer contains some operation. It is based on using small kernels also of the cyclic SOMs with size 3×3 moved on the input in a convolution way. A pseudo-code for the cyclic convolution SOMs is illustrated in Table 3.

5. Datasets and pre-processing

This section describes the different datasets and training details for the experimental results reported in this paper. Four wide benchmark datasets were used to compare the performance of our proposed technique to existing methods: AHDBase for Arabic digits, MNIST for English digits, CMU-PIE for faces, and CIFAR-10 for objects.

5.1 CMU-PIE for faces

CMU-PIE face database [32] is an available database for studying illumination, pose, and expression problems in face recognition. There are 68 individuals in three different facial expressions, under 43 different lighting, and for 13 poses. Here, the experiment on face images was in two poses and five illuminations. Fig. 7 shows samples from this database.

5.2 AHDBASE for Arabic digits

AHDBase is an Arabic character digits' database [33]. It is composed of digits' images in BMP format from 700 subjects who wrote 70,000 digits; 10,000 for testing and 60,000 for training. Each subject wrote

Received: May 24, 2023. Revised: June 3, 2023.

Figure. 9 Sample images of MNIST dataset

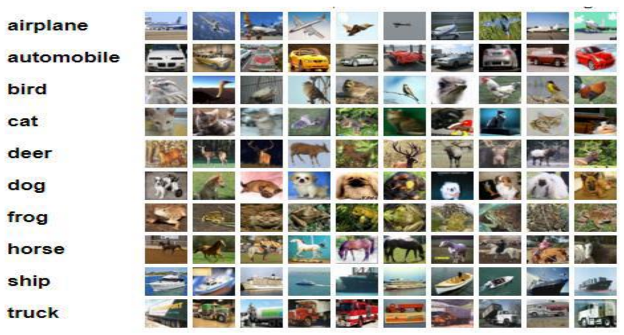


Figure. 10 An example of ten photos from each class in the CIFAR-10 database that were chosen at random

Table 4. Each dataset's pre-processing in this study

	CMU-PIE [32]	AHDBASE [33]	MNIST [34]	CIFAR-10 [35,36]
Pre-processing	✓Resize all images into 38X38 ✓Convert all images into grey			
	✓Use the Mini-max function to contrast stretching [37].		✓ Create inverse image processing [38] ✓ For each input image, perform histogram equalization (see [37])	

Table 5. Comparison between the recognition rate given in [21] and our proposed approaches, namely, cyclic convolution SOM and optimized convolution SOM method recognition rate

Method	***Recognition Rate***
CSOM [21]	87.3%
Cyclic convolution SOM (Section 4)	98.2%
Optimized Convolution SOM (Section 4)	95.4 %

twenty times each digit (from 0 to 9). Fig. 8 shows samples of the AHDBASE images.

5.3 MNIST database

The MNIST database [34] known as the modified national institute of standards and technology database, is an expansive database of transcribed digits that are generally utilized for training different networks and frameworks. The MNIST data set consists of 60,000 training and 10,000 test examples, each representing 28×28 pixel handwritten digit images. Fig. 9 shows the sample images in the MNIST dataset. All the digits in this database have been centred on fixed-size and size-normalized.

5.4 CIFAR-10 database

The well-known computer-vision dataset CIFAR-10 is employed for object recognition [35, 36]. It has 60,000 32x32 colour images that are divided into 10 item classes, each with 6000 images. Both training and test images total 50,000 and 10,000 respectively. A sample of ten randomly selected images from each class in the CIFAR-10 database is shown in Fig. 10.

5.5 Pre-processing

The datasets are pre-processed before being used in the experiment. Table 4 shows the pre-processing created with every dataset.

6. Comparisons, analysis and results descriptions

This section compares, analysis and describe the effectiveness and accuracy of our proposed algorithms, which were previously discussed in section 4. The most important comparison hold in this paper was with the CSOM which proposed in [21] and used only the MNIST dataset in their evaluation. Actually, our proposed approaches can be seen as an enhancement for this approach. In [21] the author used an image size of 256 × 256, the kernel size was 8X8, and the convolution step was equal to 4. In contrast, we used the same input images but with a size of 38×38, a kernel size of 3×3, and a convolution step equal to 1. This is because the size of the input image affects the processing time and the recognition rate. More details can be seen in the larger image than in the smaller one. Yet processing the smaller image goes more quickly than processing the bigger one. Table 5 show, for instance, how our proposed approaches previously mentioned in section 4 and those provided in [21] compare in terms of recognition rates.

The comparison between the proposed approaches and the other techniques expect the CSOM hold on the four datasets previously described. The experimental setup of the techniques used for the four databases is demonstrated in Tables 6 and 7. Table 6 presents the settings of the used deep learning techniques, while the other table shows the experimental settings of the used traditional techniques. Furthermore, the results description for each of the databases used in terms of the recognition rate is given in Table 8. The following subsections will analyze and describe in detail the experimental results for every database used.

The experimental findings for every database employed are examined and thoroughly described in the next subsections.

Received: May 24, 2023. Revised: June 3, 2023.

Table 6. Experimental setup of the used deep learning techniques

	CMU-PIE	AHDBASE	MNIST	CIFAR-10
Network parameter configuration	- The number of training epoch:100. - Training in sequential form. - The number of kernels in the proposed techniques expresses the SOM or the cyclic SOM.			
	- The experiment was on the images of 67 subjects in the train and 66 subjects in the test by using 10 images samples in five illuminations and two poses for every subject in the dataset	- The experiment was on the images of 50 subjects in the train and 55 subjects in the test by using 10 images for every digit from 0 to 9	- The experiment was on the full dataset 58% for the training phase and 42% for the testing phase	- The experiment was on the full dataset 58% fror the training phase and 42% for the testing phase
Optimized Cyclic convolution SOM	- The Kernal size is 3X3. - There are four convolution layers with different numbers of kernels. - Number of kernels for every layer (35,30,25,20) in order.	- The Kernal size is 3X3. - There are four convolution layers with different numbers of kernels. - Number of kernels for every layer (30,25,25.20) in order.	- The Kernal size is 3X3. - There are four convolution layers with different numbers of kernels. - Number of kernels for every layer (30,55,25.20) in order.	- The Kernal size is 3X3. - There are four convolution layers with different numbers of kernels. - Number of kernels for every layer (40,35,25.15) in order.
Optimized Convolution SOM				
CNN	- The Kernal size is 3X3. - There are three convolution layers with different numbers of kernels. - Number of kernels for every layer (55,30,20) in order.	- The Kernal size is 3X3. - There are three convolution layers with different numbers of kernels. - Number of kernels for every layer (50,55,.25) in order.	- The Kernal size is 3X3. - There are three convolution layers with different numbers of kernels. - Number of kernels for every layer (50,55,25) in order.	- The Kernal size is 3X3. - There are three convolution layers with different numbers of kernels. - Number of kernels for every layer (50,75,20) in order.
LSTMs	- Create four hidden layers with this number of states in order (60,50,30,20).	- Create four hidden layers with this number of states in order (50,50,20,10).	- Create four hidden layers with this number of states in order (50,50,20,10).	- Create four hidden layers with this number of states in order (60,50,20,15).

6.1 CMU-PIE database

By applying the proposed cyclic convolution SOM to the faces database its recognition rate reached 98.51% while the results of using the optimized convolution SOM was 97.01%. This provided that the change in the way of finding the best matching neuron and adding the cyclic learning rate algorithm improved the general recognition rate of the proposed method. To achieve this accuracy using the faces database from the proposed cyclic convolution SOM, we employed a maximum learning rate of 0.95 and a minimum learning rate of 0.57 with length steps of 4. On another face, when applying regular CNN to this dataset it gives a 97.01 % recognition rate only. The LSTMs recognition rate is 91.887%. When applying just the cyclic SOM we reached an accuracy equal to 96.42%. And when applying the SOM, PCA, SVM, and MLP the recognition rate was as followed 89.02%, 59.7%, 60.10, and 83.54%. Fig. 11 (a) shows the performance of the techniques in the face recognition.

Fig. 12 (b) shows the learning curve performance of which shows the loss curve for the training data and the val_loss of the validation data. Also, it shows the iteration which we have in it as the best model for face recognition. From this figure, we can see that we have the best model for this data set in only the first 20 iterations and the validation loss was less than 0.2 for having a 98.51% recognition rate.

6.2 AHDBASE database

In the case of the Arabic digits database, the recognition rate of cyclic convolution SOM reached 97.7% while the optimized convolution SOM recognition rate was 96.57%. To achieve this

Received: May 24, 2023. Revised: June 3, 2023.

Table 7. Experimental setup of the used traditional techniques

	CMU-PIE	AHDBASE	MNIST	CIFAR-10
Network parameter configuration	- The number of training epoch:100. - Training İn sequential form. The experiment was on the images of 67 subjects in the train and 66 subjects in the test by using 10 images samples in five illuminations and two poses for every subject in the dataset	The experiment was on the images of 50 subjects in the train and 55 subjects in the test by using 10 images for every digit from 0 to 9	The experiment was on the images of 50 subjects in the train and 55 subjects in the test by using 10 images for every digit from 0 to 9	The experiment was on the images of the ten classes of the objects by using 600 images form every class in train and just 100 images form every class in test
SOM Cyclic SOM	- Create a map with a size of 20X30 neuron - The number of training images is 670 - The number of testing images is 660	- Create a map with a size of 20 X18 neuron - The number of training images is 5000 - The number of testing images is 5500	- Create a map with a size of 20X20 neuron - The number of training images is 5000 - The number of testing images is 5500	- Create a map with a size of 30X35 neuron - The number of training images is 6000 - The number of testing images is 1000
PCA SVM	- The number of training images is 670 - The number of testing images is 660	- The number of training images is 5000 - The number of testing images is 5500	- The number of training images is 5000 - The number of testing images is 5500	- The number of training images is 6000 - The number of testing images is 1000
MLP	- Create a network with 3 hidden layers - The total number of hidden neurons was 67	- Create a network with 3 hidden layers - The total number of hidden neurons was 20	- Create a network with 3 hidden layers - The total number of hidden neurons was 20	- Create a network with 3 hidden layers - The total number of hidden neurons was 84

Table 8. The results description in terms of the recognition rate across four databases

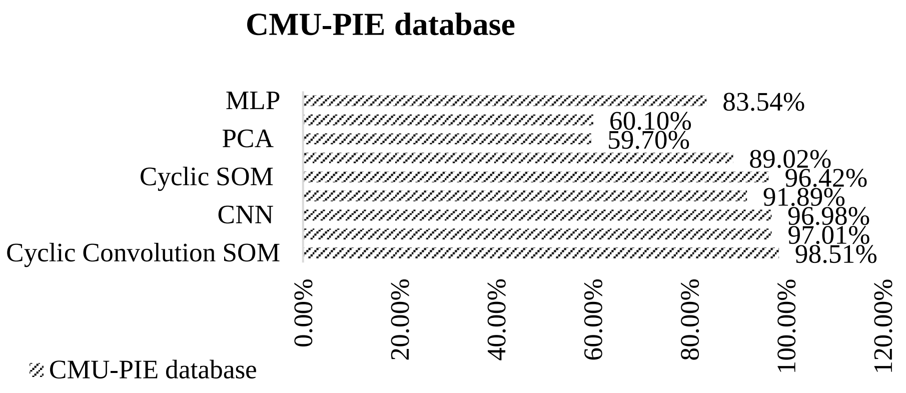
	CMU-PIE	AHDBASE	MNIST	CIFAR-10
Cyclic Convolution SOM	98.51%	97.7%	98.2%	93.8%
Convolution SOM	97.01%	96.57%	95.4%	89.23%
CNN	96.98%	95.4%	97.8%	92.2%
LSTMs	91.89%	92.4 %	96.25%	90.24%
Cyclic SOM	96.42%	80.9%	90.35%	85.77%
SOM	89.02%	74.8 %	80.4%	73.87%
PCA	59.7%	60%	76.78%	56.01%
SVM	60.1%	80%	85.6%	65.34%
MLP	83.54 %	75%	82 %	80.14%

accuracy from the proposed cyclic convolution SOM, we employed a maximum learning rate of 1 and a minimum learning rate of 0.9 with length steps of 2. On another hand, when applying regular CNN to this dataset it gives a 95.04% recognition rate. The LSTMs recognition rate is 92.4%. When applying just the cyclic SOM we reached an accuracy equal to 80.9%. And when applying the SOM, PCA, SVM, and MLP the recognition rate was as followed 74.80%, 60%, 80, and 75%. Fig.12 (a) shows the performance of the in Arabic digits recognation.

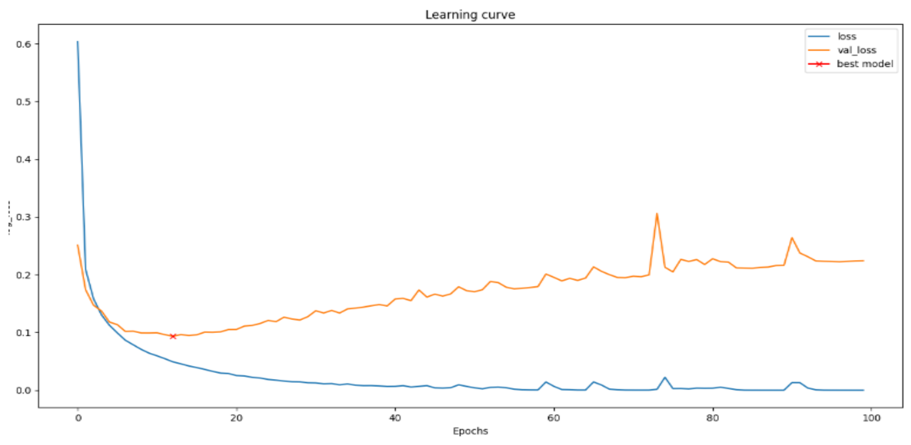
Fig. 12 (b) displays the learning curve performance of the cyclic convolution SOM and displays the loss of the validation data together with the loss curve for the training data. Additionally, it demonstrates the iteration that contains the most accurate model for recognising Arabic numerals. This figure shows that we were able to create the best model for this data set in just the first 15 iterations, and the validation loss was less than 0.2 for a recognition rate of 97.7 %.

6.3 MNIST database

The English digits database's recognition rate increased to 98.2 % when the proposed cyclic convolution SOM was applied to it, as opposed to 95.4 % when the optimized convolution SOM was

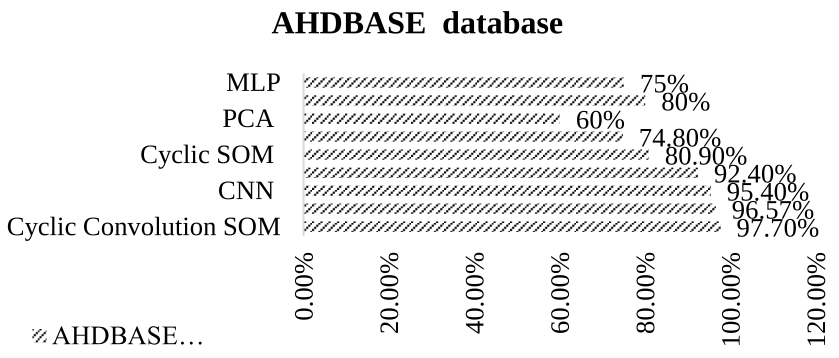


(a)



(b)

Figure 11: (a) Faces recognition rate and (b) the learning curve of cyclic convolution SOM in face recognition



(a)

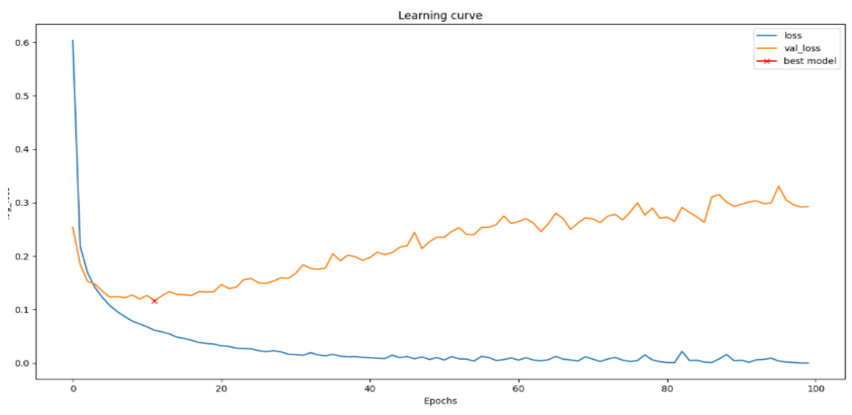
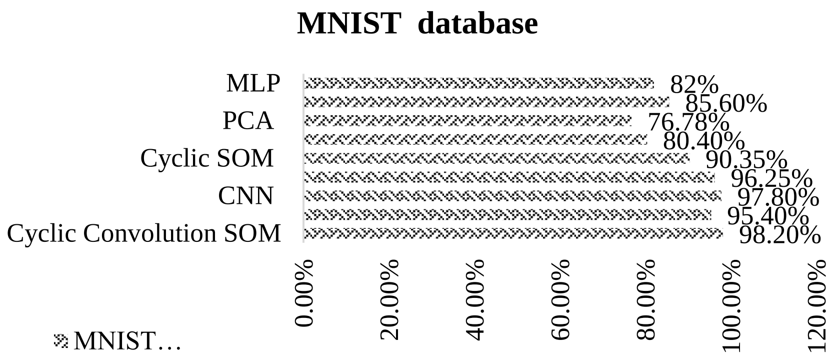
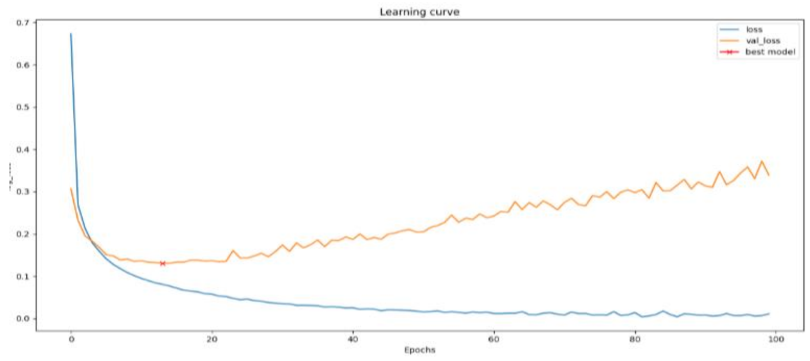


Figure. 12: (a) Arabic digits' recognition rate and (b) the learning curve of cyclic convolution SOM in Arabic digits' recognition

Received: May 24, 2023. Revised: June 3, 2023.

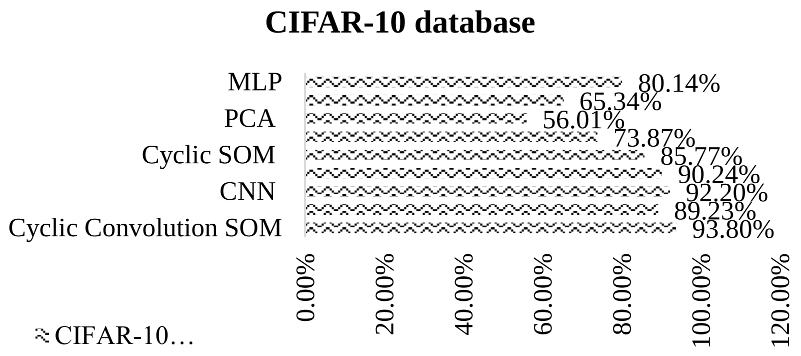


(a)

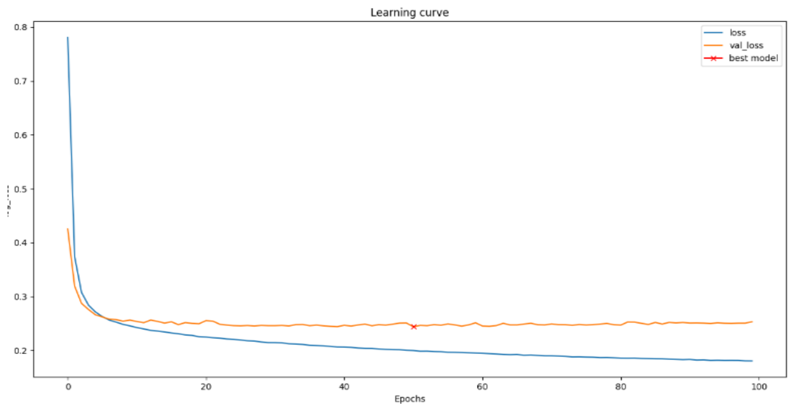


(b)

Figure. 13: (a) English digits recognition rate and (b) The learning curve of cyclic convolution SOM in English digits' recognition



(a)



(b)

Figure. 14: (a) Object recognition rate and (b) the learning curve of cyclic convolution SOM in object recognition

used. Using the English digit database from the suggested cyclic convolution SOM, we used a maximum learning rate of 0.95 and a minimum learning rate of 0.57 to obtain this accuracy. When using regular CNN on a different face in this dataset, it only achieves a 97.8% recognition rate. The LSTMs recognition rate is 96.25%. When using only the cyclic SOM, we were able to get an accuracy of 90.35 %. The recognition rate was 80.4 %, 76.78 %, 85.60 %, and 82 % when the SOM, PCA, SVM, and MLP were applied. The effectiveness of the techniques in English digits recognations appear in Fig. 13 (a).

Fig. 13 (b) shows the learning curve performance of the cyclic convolution SOMs which shows the loss curve for the training data and the val_loss of the validation data. Also, it shows the iteration which we have in it as the best model for English digit recognition. From this figure, we can see that we have the best model for this data set in the first 20 iterations and the validation loss was less than 0.2 for having a 98.2% recognition rate.

6.4 CIFAR-10 database

In the case of the object recognition database, the recognition rate of cyclic convolution SOM reached 93.8% while the optimized convolution SOM recognition rate was 89.32%. To achieve this accuracy from the proposed cyclic convolution SOM, we employed a maximum learning rate of 1 and a minimum learning rate of 0.5 with length steps of 4. On another hand, when applying regular CNN to this dataset it gives a 92.2% recognition rate. When applying just the cyclic SOM we reached an accuracy equal to 85.77%. The LSTMs recognition rate is 90.24%. And when applying the SOM, PCA, SVM, and MLP the recognition rate was as followed 73.87%, 56.01%,65 .34%, and 80.14%. Fig. 14 (a) shows the performance of the techniques after applying them to the object recognition dataset.

Fig. 14 (b) shows the learning curve performance of the cyclic convolution SOMs which shows the loss curve for the training data and the val_loss of the validation data. Also, it shows the iteration which we have in it as the best model for object recognition. From this figure, we can see that we have the best model for this data set in the first 40 iterations and the validation loss was less than 0.3 for having a 93.8% recognition rate.

7. Conclusion and future work

CNNs have millions of parameters, and both the convolutional layer and the fully connected layer are computationally expensive and memory-intensive, which limits their practical implementation. As discussed in [16, 24, 39], this paper addressed issues such as hyper-parameter optimization, feature-map exploitation, fast convolution, and learning issues. Following are some of the challenges encountered when training deep CNNs models that are relevant to these issues and we have addressed:

1) Hyper-parameter tuning, particularly learning rate (LR) is a tedious and intuition driven task, which cannot be defined via explicit formulation.
2) Training a convolutional neural network can be computationally expensive and time consuming.
3) Since deep CNNs rely on supervised learning techniques, enough labelled data must be available for proper learning. Humans, however, can learn from a few examples and make generalizations.

The main contribution of our work is the type of classifier located at the end of the CNNs architecture (i.e. FC layer) and the feature extraction method used by SOM. The concept is to employ SOMs as a key component for feature extraction in a convolutional layer with a cyclic learning rate instead of conducting experiments to find the appropriate learning rate values and schedule. For the first two items, a novel optimization approach called “cyclic convolution SOM” is developed using a cyclic learning rate (CLR) in the learning phase of SOMs to speed up their learning process as opposed to the convolution structures used in the CNNs architectural design. Moreover, classification using the KNN classifier in place of CNN's conventional MLP is performed. Regarding the last issue, unsupervised learning under various datasets using SOM was presented as it is the best approach for the recognition issue. Four wide benchmark datasets illustrate the efficiency of the proposed techniques compared with various recognition algorithms. The use of several novel concepts in CNN's architectural design has transformed the focus of study, particularly in healthcare, such as in relation to the COVID-19 pandemic [12]. Our future work will focus on this direction using DL algorithms, particularly CNNs.

Conflicts of interest

The authors declare no competing interests.

Author contributions

“Conceptualization, all the authors; methodology, all the authors; software, Samar; Validation, Hussein, and Samar; formal analysis, Hussein and Samar; investigation, Hussein and Samar; writing—original draft preparation, Samar; writing—review and

editing, Hussein and Samar; visualization, Hussein; data collection and implementation, Hussein and Samar; supervision, Hussein; project administration, Hussein, Fayed, and Mohamed"

References

[1] S. Guyon, M. Gunn, Nikravesh, and L. A. Zadeh, *Feature Extraction: Foundations and Applications*, Vol. 207, Springer, Cham, Switzerland, 2008, doi: 10.1007/978-3-540-35488-8, 2008.

[2] O. Krakovska, G. Christie, A., Sixsmith, M. Ester, and S. Moreno, "Performance comparison of linear and non-linear feature selection methods for the analysis of large survey datasets", *Fragkos KC*, Vol. 14, No. 3, 2019.

[3] A. Najafi, Joudaki, and E. Fatemizadeh, "Nonlinear dimensionality reduction via path-based isometric mapping", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 38, No. 7, pp. 1452-1464, 2015.

[4] M. Piernik and T. Morzy, "A study on using data clustering for feature extraction to improve the quality of classification", *Knowledge and Information Systems*, Vol. 63, No. 7, pp. 1771-1805, 2021,

[5] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, "A survey of convolutional neural networks: analysis, applications, and prospects", *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 33, No. 12, pp. 6999-7019, 2021.

[6] H. Rahul, H. and R. L. Jyothi, "Convolutional Neural Networks: A Comprehensive Survey", *International Journal of Applied Engineering Research, ISSN 0973-4562*, Vol. 14, No. 3, pp. 780-789, 2019.

[7] H. Mohamed, A. Hamza, and H. Hefny, "An Efficient Intrusion Detection Approach Using Ensemble Deep Learning models for IoT", *International Journal of Intelligent Engineering and Systems*, Vol. 16, No. 1, 2023, doi: 10.22266/ijies2023.0228.31.

[8] G. Bompem, N. Chiluka, and D. Pandluri, "Effective Twitter Sentiment Analysis Using Deep Belief Network with Enhanced Dragonfly Optimization Algorithm", *International Journal of Intelligent Engineering and Systems*, Vol. 16, No. 2, 2023, doi: 10.22266/ijies2023.0430.06.

[9] P. Govindan and A. Kumarappan, "A Reversible Convolutional Neural Network Model for Sign Language Recognition", *International Journal of Intelligent Engineering and Systems*, Vol. 15, No. 2, 2022, doi: 10.22266/ijies2022.0430.16.

[10] B. H. Prasetio, E. R. Widasari, and F. A. Bachtiar, "A Study of Machine Learning Based Stressed Speech Recognition System", *International Journal of Intelligent Engineering and Systems*, Vol. 15, No. 4, 2022, doi: 10.22266/ijies2022.0831.04.

[11] M. Heidari, S. Mirniaharikandehei, A. Z. Khuzani, G. Danala, Y. Qiu, and B. Zheng, "Improving the performance of CNN to predict the likelihood of COVID-19 using chest X-ray images with pre-processing algorithms", *International Journal of Medical Informatics*, Vol. 144, 2020.

[12] C. Shorten, T. M. Khoshgoftaar, and B. Furht, "Deep learning applications for COVID-19", *Journal of Big Data*, Vol. 8. No. 1, pp. 1-54, 2021.

[13] C. V. Angkoso, H. P. A. Tjahyaningtijas, I. K. E. Purnama, and M. H. Purnomo, "Multiplane Convolutional Neural Network (Mp-CNN) for Alzheimer's Disease Classification", *International Journal of Intelligent Engineering and Systems*, Vol. 15, No. 1, 2022, doi: 10.22266/ijies2022.0228.30.

[14] A. Dhillon and G. K. Verma, "Convolutional neural network: a review of models, methodologies and applications to object detection", *Progress in Artificial Intelligence*, Vol. 9. No. 2, pp. 85-112, 2020.

[15] S. Joshi, D. K. Verma, G. Saxena, and A. Paraye, "Issues in training a convolutional neural network model for image classification", In: *Proc. of Advances in Computing and Data Sciences: Third International Conf., ICACDS 2019*, Springer Singapore, Ghaziabad, India, April 12–13, pp. 282-293, 2019.

[16] G. Habib and S. Qureshi, "Optimization and acceleration of convolutional neural networks", A survey", *Journal of King Saud University, Computer and Information Sciences*, Vol. 34, No. 7, pp. 4244-4268, 2022.

[17] T. Ahonen, A. Hadid, and M. Pietikainen, "Face description with local binary patterns: Application to face recognition", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol 28, No. 12, pp. 2037-2041, 2006.

[18] C. Cheng and K. K. Parhi, "Fast 2D convolution algorithms for convolutional neural networks", *IEEE Transactions on Circuits and Systems I*, Vol. 67, No. 5, pp. 1678-1691, 2020.

[19] Y. Zhao, D. Wang, and L. Wang, "Convolution accelerator designs using fast algorithms", *Algorithms*, Vol. 12, No. 5, No. 112, 2019.

[20] P. Samudre, P. Shende, and V. Jaiswal, "Optimizing performance of convolutional

neural network using computing technique", In: *Proc. of 5th International Conf. for Convergence in Technology (I2CT)*, Bombay, India, pp. 1-4, 2019.

[21] H. Dozono, G. Niina, and S. Araki, "Convolutional self-organizing map", In: *Proc. of International Conf. on Computational Science and Computational Intelligence (CSCI), IEEE*, pp. 767-771, 2016.

[22] L. N. Smith, "Cyclical learning rates for training neural networks", In: *Proc. of 2017 IEEE Winter Conf. on Applications of Computer Vision (WACV)*, Santa Rosa, CA, USA, pp. 464-472, 2017.

[23] S. Antar, H. K. Hussein, M. Hashim, and F. Ghaleb, "Cyclic Self-organizing Map for Object Recognition", *Information Sciences Letters*, Vol. 12, No. 5, 2023.

[24] L. Alzubaidi, J. Zhang, A. J. Humaidi, et al., "Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions", *Journal of big Data*, Vol. 8, pp. 1-74. 2021.

[25] T. Kohonen, "Self-organizing maps", *Information Sciences*, Berlin: Springer, Vol. 30, 2001.

[26] M. I. Khelil, M. Ladjal, Y. Brik, and M. A. Ouali, "Self-Organizing Maps-Based Features Selection with Deep LSTM and SVM Classification Approaches for Advanced Water Quality Monitoring", *International Journal of Intelligent Engineering and Systems*, Vol. 15, No. 3, 2022, doi: 10.22266/ijies2022.0630.09.

[27] K. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks", *Communications of the ACM*, Vol. 60. No. 6, pp. 84-90, 2017.

[28] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, et al., "Going deeper with convolutions", In: *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 1-9, 2015.

[29] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition", In: *Proc. of the 3rd International Conf. on Learning Representations (ICLR2015)*, 2015,

[30] H. K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition", In: *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 770-778, 2016.

[31] D. Varghese, "Comparative study on classic machine learning algorithms", Available online: https://towardsdatascience.com/comparative-study-on-classic-machine-learning-algorithms-24f9ff6ab222, 2018.

[32] T. Sim, S. Baker, and M. Bsat, "The CMU Pose, Illumination, and Expression (PIE) database", In: *Proc. of Fifth IEEE International Conf. on Automatic Face Gesture Recognition*, Washington, DC, USA, pp. 53-58, 2002.

[33] E. A. E. Sherif and S. Abdelazeem, "A Two-Stage System for Arabic Handwritten Digit Recognition Tested on a New Large Database", *Artificial Intelligence and Pattern Recognition*, pp. 237-242, 2007.

[34] L. Deng, "The MNIST Database of Handwritten Digit Images for Machine Learning Research [Best of the Web]", *IEEE Signal Processing Magazine*, Vol. 29, No. 6, pp. 141-142, 2012.

[35] URL: https://www.cs.toronto.edu/~kriz/cifar.html

[36] P. Shakti, P. Aishwarya, and P. Ambika "Object detection using CIFAR-10 dataset simplene", *International Research Journal of Engineering and Technology*, Vol. 8, No. 6, 2021.

[37] M. Lucas, Iliadis, R. Molina, and A. K. Katsaggelos, "Using deep neural networks for inverse problems in imaging: beyond analytical methods", *IEEE Signal Processing Magazine*, Vol. 35. No.1, pp. 20-36, 2018.

[38] P. Swaroop and N. Sharma, "An overview of various template matching methodologies in image processing", *International Journal of Computer Applications*, Vol. 153, No. 10, pp. 8-14, 2016.

[39] A. Khan, U. Sohail, Zahoora, and A. S. Qureshi, "A survey of the recent architectures of deep convolutional neural networks", *Artificial Intelligence Review*, Vol. 53, pp. 5455–5516, 2020.