Information Sciences Letters

Volume 12
Issue 5 *May 2023*

Article 23

2023

Cyclic Self-Organizing Map for Object Recognition

S. Anter
Mathematics & Computer Science Department, Faculty of Science, Ain Shams University, Abbasia 11566, Cairo, Egypt, h_karam@sci.edu.ac.eg

H. K. Hussein
Mathematics & Computer Science Department, Faculty of Science, Ain Shams University, Abbasia 11566, Cairo, Egypt, h_karam@sci.edu.ac.eg

M. Hashim Ali
Mathematics & Computer Science Department, Faculty of Science, Ain Shams University, Abbasia 11566, Cairo, Egypt, h_karam@sci.edu.ac.eg

F. F. Mohamed
Mathematics & Computer Science Department, Faculty of Science, Ain Shams University, Abbasia 11566, Cairo, Egypt, h_karam@sci.edu.ac.eg

Inf. Sci. Lett. **12**, No. 5, 1775-1788 (2023)

Information Sciences Letters
An International Journal

http://dx.doi.org/10.18576/isl/120523

Cyclic Self-Organizing Map for Object Recognition

S. Anter, H. K. Hussein, M. Hashim Ali and F. F. Mohamed*

Mathematics & Computer Science Department, Faculty of Science, Ain Shams University, Abbasia 11566, Cairo, Egypt

Received: 22 Feb. 2022, Revised: 22 Nov. 2022, Accepted: 6 Dec. 2022.
Published online: 1 May 2023.

Abstract: Object recognition is an important machine learning (ML) application. To have a robust ML application, we need three major steps: (1) preprocessing (i.e. preparing the data for the ML algorithms); (2) using appropriate segmentation and feature extraction algorithms to abstract the core features data and (3) applying feature classification or feature recognition algorithms. The quality of the ML algorithm depends on a good representation of the data. Data representation requires the extraction of features with an appropriate learning rate. Learning rate influences how the algorithm will learn about the data or how the data will be processed and treated. Generally, this parameter is found on a trial-and-error basis and scholars sometimes set it to be constant. This paper presents a new optimization technique for object recognition problems called Cyclic-SOM by accelerating the learning process of the self-organizing map (SOM) using a non-constant learning rate. SOM uses the Euclidean distance to measure the similarity between the inputs and the features maps. Our algorithm considers image correlation using mean absolute difference instead of traditional Euclidean distance. It uses cyclical learning rates to get high performance with a better recognition rate. Cyclic-SOM possesses the following merits: (1) it accelerates the learning process and eliminates the need to experimentally find the best values and schedule for the learning rates; (2) it offers one form of improvement in both results and training; (3) it requires no manual tuning of the learning rate and appears robust to noisy gradient information, different model architecture choices, various data modalities and selection of hyper-parameters and (4) it shows promising results compared to other methods on different datasets. Three wide benchmark databases illustrate the efficiency of the proposed technique: AHD Base for Arabic digits, MNIST for English digits, and CMU-PIE for faces.

Keywords: Machine learning, Self-organizing maps (SOMs), Neural networks, Feature extraction, Cyclic learning rate.

1 Introduction

Object recognition is an important aspect of visual perception. It is the basis of many machine learning applications, such as biometrics, surveillance, motion detection, object detection, tumor detection, industrial inspection, content-based image retrieval, robotics, medical analysis, document recognition, and human-computer interaction (HCI), intelligent vehicle systems, etc. It can be defined as a labeling problem based on models of known objects. Humans perform object recognition effortlessly and instantaneously, but how do we train computers to do the same thing? This is the question that the researchers are attempting to answer.

Formally, given an image containing one or more objects of interest and a set of labels corresponding to a set of models known to the system, the system should assign correct labels to regions, or a set of areas in the image. Research in object recognition has been dominated by approaches that separate processing into distinct stages of feature extraction, recognition, and matching [1]. In the first stage, discrete primitives, or "features" are detected. In the second stage, stored models are matched against those features. A feature is some attribute of the object that is considered important in describing and recognizing the object from other objects. Shape, size, and color are some commonly used features. By referring to the method of extraction of features, we were referring to the methodology of dimensional reduction [2]. This method was used to delete irrelevant data features to increase recognition [3]. Many algorithms developed for these processes such as the self-organizing map (SOM) [4] the multi-layer perceptrons (MLPs) [5], the support vector machine (SVM) [6], and the principal component's analysis (PCA) [7]. In the feature recognition process, the identification or the classification of the objects appears. The accuracy of this process depends on the fidelity of extracted features. Common algorithms for performing classification include K-nearest neighbor (K-NN), Naïve Bayes, discriminant analysis, logistic regression, and neural networks. For instance, the K-nearest neighbor (K-NN) technique [8] classifies the inputs according to the number of elements surrounding them. It was utilized in this paper to assess the accuracy of features extracted using either the normal SOM or the proposed Cyclic-SOM.

*Corresponding author e-mail: h_karam@sci.edu.ac.eg

Cyclic-SOM is a new optimization proposed in this paper technique for object recognition problems. It accelerates the learning process of the self-organizing maps (SOMs) using a non-constant learning rate. The extraction of features at an appropriate acceptable learning rate is an essential step in any object recognition technique. Learning Rate is an important hyper-parameter that should be given carefully during the training of the model. It influences how the algorithm will learn about the data or how the data will be processed and treated. When choosing a learning rate that is too small, can lead to a long training process, whereas a value that is too large can lead to a sub-optimal set of weights in an unstable training process. Generally, this parameter is found on a trial-and-error basis and researchers sometimes set it to be constant. The key elements of Cyclic-SOM are measuring the propinquity between the input and the feature map using mean absolute difference instead of traditional Euclidean distance and training with cyclical learning rates (CLR). The used CLR was first suggested by [9] and later updated by [10]. Instead of setting the learning rate to fixed values, the proposed technique lets the learning rate cyclically vary within reasonable boundary values. Cyclic-SOM achieves near-optimal recognition accuracy without tuning and often in many fewer iterations. Comparisons with existing techniques are included, and the advantages of our proposed technique over existing ones are examined using three wide benchmark datasets: AHDBase for Arabic digits, MNIST for English digits, and CMU-PIE for faces.

This paper is organized as follows. The problem statement, motivation, and objectives of this research are discussed in section 2. Section 3 summarizes the background, the basic terminology, and notions that will be used throughout this paper. Section 4 introduces our proposed Cyclic-SOM methodology. Analysis of our proposed algorithm compared with existing methods is discussed in Section 5. Evaluation and practical performance results are discussed in Section 6. Section 7 concludes this paper with some direction for future work.

2 Problem Statement, Motivation, and Objectives

Training and testing are two key processes of any object recognition technique. During the training process, each class is given its description. In the testing process, the test images are classified into different classes for which the system is supposed to be trained. Assignment of images to a particular class is performed based on training features. Increasing training time for the model doesn't necessarily improve the performance of the model and can cause other issues such as overfitting and exploding gradient [11].

Overfitting occurs when the model fits well enough to the training dataset but the model is not capable of generalizing new examples which are not part of the training dataset. The exploding gradient is identified when the training model cannot learn from the training data after a certain number of epochs and results in overflow. Before training, the model can be configured with different parameters (e.g. weights, biases, number of layers, etc.) and hyper-parameters (e.g. processing units (neurons), filter size, activation function, learning rate, etc.). Hyper-parameters are constants, which need their values to be predefined before the models could be constructed. According to [12], these hyper-parameters affect the overall performance of the network. Learning rate is a critical hyper-parameter that should be carefully considered during the training process. Generally, this parameter is found on a trial-and-error basis and scholars sometimes set it to be constant. Self-organizing maps (SOMs) are one of the common neural network models, which solve a lot of object detection issues. A little change in the hyper-parameter values can affect the overall performance of SOMs. That is why careful selection of hyper-parameters is a major design issue that needs to be addressed through some suitable optimization strategy. This paper addresses this issue using a new optimization strategy for object recognition problems called Cyclic-SOM.

Cyclic-SOM improved the (SOMs) as a neural network technique in feature extraction and recognition by considering image correlation using mean absolute difference instead of traditional Euclidean distance. It uses cyclical learning rates [10] and measures the appropriateness between the input and the feature map in a reasonable time with a maximum recognition rate. Our approach shows promising potential with low computational cost compared to other methods such as regular SOMs, (MLPs), (SVM) and (PCA)) using three different benchmark datasets: AHDBase for Arabic digits, MNIST for English digits, and CMU-PIE for faces.

3 Foundations and Basic Terminology

This section briefly provides an overview of the neural networks, regular SOMs, the training process, and the most common hyperparameters (e.g. learning rate) in the training process that SOMs go through.

3.1. Neural Networks

Neural Networks (NNs) is a computational model that is inspired by the way biological neural networks in the human brain process information. The "network" part of the (NNs) comes from the fact that they consist of interconnected neurons, where each neuron activating (or firing) in the network cascades down to other neurons which are connected to it. Neurons are divided into layers (input, hidden, and output layers) and each layer performs a certain change in the input parameters. Each neuron has activation, which is a number between 0 and 1. In the input layer, this activation changes based on the input data, and in the case of the later neurons in the network, it is based on what is input from the neurons in the layers before them. This input is influenced by the connections between the neurons, which are called weights, which hold a rational number. Weights are used to connect each neuron in one layer to every neuron in the next layer. Weight determines the strength of the connection of the neurons. There are two types of (NNs): biological and artificial neural networks. Artificial neural networks (ANNs) are algorithms for learning. ANNs consist of the following components: inputs, which are multiplied by weight coefficients, and then are computed by some mathematical function (like summation function) which determines the activation of the neuron (activation function), and at the end, there is an output. These ANNs are called networks because they are composed of different functions, which gather knowledge by detecting the relationships and patterns in data using past experiences known as training examples. The learned patterns in the data are modified by an appropriate activation function (AF) and presented as the output of the neuron. Activation functions [13] are functions used in (NNs) to compute the weighted sum of input and biases, which is used to decide if a neuron can be fired or not (refers to Fig. 1). The bias is a measure of how easy it is to get the perceptron to fire. Weight determines the strength of the connection of the neurons. It decides how fast the activation function will trigger whereas bias is used to delay the triggering of the AF. AF manipulates the presented data through some gradient processing usually gradient descent [14] and afterward produces an output for the NNs, that contains the parameters in the data. A commonly used (AF) is the sigmoid function. It "squishes" the activation between 0 and 1; with the output of the neuron being 0 if the input is very small and 1 if the activation is very large, other values being in between. The purpose of the (AF) is to introduce non-linearity into the output of a neuron. The non-linear output after the application of the AF is given by:

$$y = \alpha(w_1 x_1 + w_2 x_2 + \cdots + w_n x_n + b) \tag{1}$$

Where α is the AF, x=input vectors, w=weights, and b=biases (a constant which helps the model in a way that it can fit best for the given data).

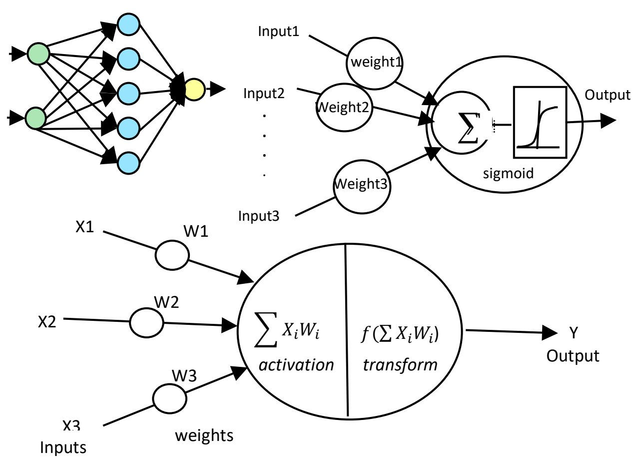


Fig. 1: ANN and its components

3.2 Training Process

The general machine-learning algorithms aim to update a set of parameters "x" to optimize an objective function f(x). The objective function is the most general term for any function that you optimize during training. The goal of any optimization problem when training and tweaking neural networks is to find the best weights/biases which will return the lowest cost. The cost function (also referred to as loss or error) is used to estimate how badly models are performing. It is meant to measure how well each iteration of the neural network is performing, with a higher cost meaning a worse performance of the network.

A common way of calculating cost is by the use of the square of differences, where for each output of the network the output is substituted from the expected output, squared (which also avoids negative values), and these values are then summed together. This is typically expressed as a difference or distance between the predicted value and the actual value. The most important part of a neural network is the training process. To allow the network to develop the right weights in its feature detectors, the network must go through a period of training in which a manifold of the image and its correct label (classification) is provided. The network learns from the label and adjusts the weights it gives to certain features. The process of learning from errors and adjustment of weights is called “back-propagation”.

The goal of back-propagation is to optimize the weights so that the neural network can learn how to correctly map arbitrary inputs to outputs. Let us assume that we have a training set consisting of images with their corresponding correct labels. Back-propagation can be broken down into 4 parts, forward pass, loss function, backward pass, and weight adjustment. During the forward pass, an image from the training set is passed into the model as an input. The Loss function calculates the error between the output of the model and the correct label which we can obtain from the training set. A loss function is a part of a cost function which is a type of objective function. Optimization of an objective function using a back-propagation algorithm is similar to the response-based learning of the human brain. The most common loss function (L) is mean square error (MSE), which is half times (actual - predicted) squared calculated as follows:

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

Now, what we want to do is perform a backward pass through the network, which is determining which weights contributed most to the loss and finding ways to adjust them so that the loss decreases. To do this, we use optimization algorithms like gradient descent [14] where weights (w) are iteratively adjusted with a loss function L as follows:

$$w^t = w^{t-1} - \eta_t \, {\partial L}/{\partial w} \tag{3}$$

Where, η_t is the learning rate whose choice is crucial. Gradient descent [14] enables a model to learn the gradient or direction that the model should take to reduce errors (differences between actual and predicted values).

3.3. Learning Rate.

Different learning algorithms involve different sets of hyper-parameters, and it is useful to get a sense of the kinds of choices that practitioners have to make in choosing their values. Hyper-parameters are some parameters that appear in the training objective function, but not in the network itself, and need their values to be predefined before the models could be constructed. Choosing hyper-parameter values is formally equivalent to the question of model selection, i.e., given a family or set of learning algorithms, how to pick the most appropriate one inside the set? We define a hyper-parameter for a learning algorithm "LA" as a variable to be set before the actual application of "LA" to the data, one that is not directly selected by the learning algorithm itself. Learning Rate (LR) is one important hyper-parameter that should be given carefully when training a neural network and one should always make sure that it has been tuned. It changes its value during the training process according to a specific rule for enhancing the recognition rate and plays an important role in the speed of learning. Furthermore, it influences how much the weights are adjusted during back propagation, and if it is incorrectly set, it can cause the network to converge at suboptimal local minima. It is well known that too small a learning rate will make a training algorithm converge slowly, while too large a learning rate will make the training algorithm diverge [10]. Generally, the learning rate is found on a trial-and-error basis, although research is being done on finding a more optimal solution [10]. According to Eq. (3), the value of η affects convergence and the learning speed as well as the overall performance of SOMs. Thus, there is a need for an optimization strategy to adapt the learning rate to make the learning process faster and more efficient. This paper achieved this goal using a new optimization strategy called Cyclic-SOM for object recognition problems by accelerating the learning process of the SOMs using a new training process that SOMs go through. Cyclic-SOM is an effective technique that aims to provide optimal learning for the neural network. We used cyclic learning rate (CLR) [9,10] which is a step-based method as it depends on the previous steps in the training. Its merit is to have triangle schedule decay as shown in Fig. 2

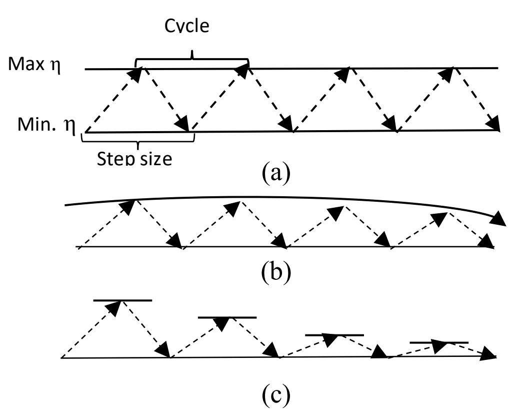


Fig. 2: The cyclic schedule prhaption (a) Triangle schedule,(b) Triangle schedule with exponential decay, and (c) schedule with fixed decay

A constant learning rate, a reduced learning rate, and a random learning rate strategy are all also available [15]. Except for the constant learning rate, these strategies can use special functions to change the value. Table 1 show the common functions which can be used in the learning rate.

Table 1: Common learning rate function

METHOD	FUNCTION
Time-based method: the learning rate in this method depends on the learning rate of the previous iteration.	$\eta_{n+1} = \frac{\eta_n}{1+d_n}$
Step-based method: in this method learning rate is defined according to some pre-defined steps.	$\eta_n = \eta_0 d^{floor\frac{1+n}{r}}$
Exponential method: in this method learning rate is calculated similar to step-based but instead of steps, a decreasing exponential function is used.	$\eta_n = \eta_0\, e^{d_n}$

Where η is the learning rate, d is the learning decay, r is the drop rate and n is the iteration number.

3.4. Self-organizing Maps (SOMs)

One of the most prevalent neural network models is self-organizing maps (SOMs). It's an unsupervised learning neural network [4] that can handle a wide range of object detection problems. Compared with supervised learning, unsupervised learning does not include any labels in the training set. Success is usually determined by whether the network can reduce or increase an associated cost function. For example, the authors in [16] use the SOMs for bridge crack recognition. The 1D and 2D SOMs for skeleton tracking of body and hand are discussed in [17]. In [18], the SOMs are used for recognizing objects in an unsupervised manner. Several researchers have successfully applied various Artificial Neural Network (ANN) methods to seafloor problem classification [19,6]. In remote sensing, they used the fuzzy and SOMs techniques to decompose mixed pixels of hyperspectral imagery [20] and used the fuzzy to segment acoustic images [21]. Two main functions represent the core of SOMs: the function of finding the propinquity between the input and all the neurons (sometimes called the function of finding the winner neuron), and the function of updating the weights. The change in one or both of them can provide a big change in the SOMs work. To clearly understand the SOM algorithm, 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 wu in the map. The neuron which has more propinquity from the 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 SOM map. The function which finding the winner neuron can be as follows:

$$\| I - w_c \| = \min_u (\| I - w_u \|) \tag{4}$$

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

$$w_u(t+1) = w_u(t) + h_{cu}(t)[I(t) - w_u(t)] \tag{5}$$

Where

$h_{cu}(t) = \eta(t).\exp\left[\frac{\|r_c - r_u\|}{2\sigma^2(t)}\right]$ is the neighborhood kernel of the winner c at time t, η((t) expresses the learning, 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 SOM, related features from the inputs are transferred into neighboring spots in the feature map. Fig. 3 shows a simplified illustration of the SOM work on face recognition.

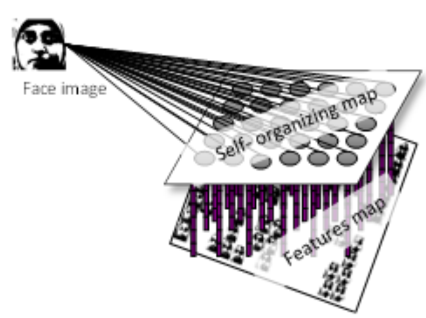


Fig. 3: Self-organizing maps work mechanism

4 Methodology

Our methodology is based on a modification of the two main core functions of the regular SOMs which are the function of finding the winner neuron and the function of updating the weights. This section presents our proposed optimization strategy called Cyclic-SOM for object recognition problems. The proposed technique accelerates the learning process of the regular SOMs using a cyclic learning rate in the learning phase. The key elements of cyclic-SOM are measuring the propinquity between the input and the feature map using mean absolute difference instead of traditional Euclidean distance and training with cyclical learning rates (CLR).

4.1. Mean Absolute Difference (MAD)

MAD is a well-known measuring method. It can compute the variation of value from a distribution [22,23]. To clearly understand its functionality, let us consider a vector X= {x1, . . ., xn } acts values in the distribution. So, the MAD can be computed as follows:

$$MAD = \frac{1}{n}\sum_{i=1}^{n} | x_i - \bar{x} | \tag{6}$$

Where n is the number of the values in the distribution, i is an incremental value used to determine the value placed in the distribution, and $\bar{x}$ is the distribution mean and compute as follows:

$$\bar{x} = n^{-1}\sum_{i=1}^{n} x_i \tag{7}$$

The MAD computes the variation of independent value and distribution which is called calculating the dissimilarity. It was utilized to calculate the propinquity between the inputs and the cyclic-SOM feature map in this paper.

4.2. Measuring the Propinquity

Measuring propinquity between the inputs and the feature maps is a crucial step in the SOMs algorithm. To present the advantage of using the mean absolute difference accurately in the proposed system, the proposed cyclic-SOM used in this paper used the mean absolute difference instead of the traditional Euclidean distance. Consider DE is the distance of Euclidean and DM is the mean absolute difference in the N-dimensional space between the input vector x and the neuron in the SOMs feature map y. The equation that defines the distance of Euclidean here is generalized as follows:

$$D_e(x, y) = (x, y)^T A^{-1}(x - y) \tag{8}$$

Where A is NxN positive symmetric semi-definite matrix.

Fig. 4 clarifies how using the distance of Euclidean and mean absolute difference is sensitive to object changes. In Fig. 4, two face pictures represent a person under two different lighting conditions, while face picture 4(c) represents another person. From these pictures, we can expect that the difference from Fig.4 (a) to Fig.4 (b) should present smaller than that of Fig.4 (b) and Fig.4 (c). By using the Euclidean distance, the computing results yield DE (a, b) = 158 and DE (b, c) = 134. While computing the mean Absolut difference yields DM (a, b) = 78 and DM (b, c) =83. Traditional Euclidean distance gives a large difference between one-person pictures. It gives a counter-intuition result. Nevertheless, using the mean absolute difference gives a small difference in one-person pictures. The distance of Euclidean only determines the pixel value differences and neglects its correlation. As regards the mean absolute difference, it determines the variation of pixel values and interests the correlation of the pixels. This means a small effect on the pictures may result in a large Euclidean distance, but it does not affect the mean absolute difference.

(a) (b) (c)

Fig.4: ((a), (b)) are pictures of a subject face under two different illumination conditions, while (c) is another person's face picture

4.3. Cyclic-SOM

As mentioned above, the Cyclic-SOM uses the statistical mean absolute difference and the CLR method. Here, we attempted to demonstrate how we included this into the proposed technique depending on finding the winner neuron in the SOMs. From Eq. (4) the winner is computed by measuring the least distance between the input and all the neurons weights in the feature map. This means the SOM tries to get the greatest propinquity between the input and the neuron's weights. As mentioned, the common equation used in SOM is the distance of Euclidean. This made the common equation for finding the winner as follows:

$$\| I_i - w_c \| = \min_j \left(\sqrt{\sum_{i=0}^{n} (I_i - w_{ij})^2} \right) \tag{9}$$

Where x is the input vector, w_c is the winner, n is the input size, and j=0, j< no. of neurons. The proposed technique converts this equation into the following:

$$\| I_i - w_c \| = \min_j \left(n^{-1} \sum_{i=0}^{n} (| I_i - w_{ij} | - M) \right) \tag{10}$$

Where M is the distance means and given by:

$$M = n^{-1} \sum_{i=0}^{n} (|I_i - w_{ij}|)$$

This conversion gives the presented algorithm more firmness as the exhibition in the results here. Also, exchanging the distance of Euclidean with the mean absolute difference and using the cyclic learning rate instead the decreasing learning rate help to upgrade the processing time. The cyclic learning rate functions can be illustrated as follow:

$$\eta_t = \eta_{min+(}\eta_{max-}\eta_{\min}) * \max(0, 1 - x) \tag{11}$$

Where x is defined as

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

The term "cycle" can be calculated as follow

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

Where the decreasing learning rate was

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

Where i is the previous iteration number.

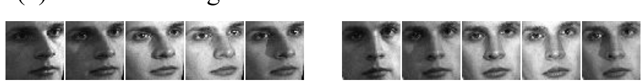
It is worth mentioning that some researchers have tried to improve the self-organizing map by using Mahalanobis distance [24] and other distance and learning rates [25]. They have good results, but the performance of their technique can be slow. Because of the specifications required for the features and because of the nature of the method.

5 Databases

This section describes the different datasets and training details for the experimental results reported in this paper. For object recognition problems, we trained and compared the performance of our proposed technique to existing methods using three wide benchmark databases: CMU-PIE for faces, AHDBase for Arabic digits, and MNIST for English digits.

5.1 CMU-PIE face database

CMU-PIE face database [26] 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. Figure 5 shows a sample for the used images under the illumination conditions in two poses. Figure 5(a) shows the images under five illuminations in pose rotated at 45 degrees. Figure 5(b) shows the images under five different illumination conditions in a frontal pose.



(a) (b)

Fig. 5: The face image for a person under 5 illuminations in two poses.

There are some enhancements for the face images prepared for improving the recognition accuracy in the experiment in this paper. It includes the normalization of the captured images by making contrast stretching [27]. Figure 6 shows the effect of the contrast stretching on the face image given in figure 5.

Fig. 6: The effect of the contrast stretching on the face image for a person in figure 5 under 5 illuminations in two poses

5.2 AHDBase

AHDBase is an Arabic character digits database [28]. 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 twenty times each digit (from 0 to 9). This paper used the inverse process with histogram equalization stretching [27] for making a pre-processing of the used images. Fig.7 (a) and 7(b) show samples of the AHDBase images before and after the pre-processing respectively.

(a) (b)

Fig. 7: Sample of the AHDBase images before and after the pre-processing.

5.3 MNIST Database

The MNIST database [29,30] 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. 8 shows the sample images in the MNIST dataset. All the digits in this database have been centered on fixed-size and size-normalized. In this paper, the same pre-process of the Arabic digits is performed on this database. Figure 9 (a) and (b) show samples of the MNIST images before and after the pre-processing, respectively.

Fig. 8: Sample Images of MNIST dataset

(a) (b)

Fig. 9: Sample of the MINST images before and after the pre-processing.

6 Experimental Results

In this section, we compare the performance of our proposed technique (Cyclic-SOM) and other existing methods using the three benchmark databases previously discussed.

6.1 Experimental Setup

The experiments in this paper used a variety of setups to collect all the studies that looked at object recognition in response to changes in feature details. It also illustrates the firmness of the suggested methodology. By referring to the proposed cyclic-SOM and the regular SOM, the experiments here were on three databases. A three-feature map was built separately for the regular SOM. In addition, three separate feature maps for the Cyclic-SOM system were built with the same characters as the used SOM map. Also, three separate feature maps for the SOM+MAD were created to check the effect of changing the way of finding the propinquity between the inputs and the feature map before using the cyclic learning rate. The feature maps in this paper were for face recognition, Arabic digit recognition, and recognition of the English digits. In the case of the faces database, a features map is made up of 30x20 neurons or ten image samples for each person used in five illuminations and two poses during the training phase. The experiments included

67 subjects for the training phase. A total of 670 images were used to create this map. The testing included 660 images from 66 different subjects. The features map of the Arabic digits database has 20x20 neurons. The train uses 5000 images as input, whereas the test uses 5500 images. The experiment employed 10 images of 50 subjects in the train and 55 subjects in the test for each digit from 0 to 9 in the written Arabic digits. For the recognition of the English digits, we also used 5000 input written English digits images of 50 subjects on the train and 5500 written English digits images of 55 subjects in the test. 10 images were used in the experiment for every digit from 0 to 9, written by the same subject. The created map was 20x18 neurons. The same face images, Arabic digits images, and English digits images were used with PCA, MLP, and SVM to evaluate their performance on the presented data. A small comparison was then performed between them and the presented technique. It's worth mentioning that the MLP, which is used for the faces database, contains 3 hidden layers. Besides, the total number of hidden neurons was 67 neurons. Also, in the Arabic and the English databases, 3 hidden layers were used but the total numbers of neurons, in this case, were 20 neurons.

6.2 Results Description

The following results are based on the feature data that was used in the experiment. The object recognition rate was divided into fifth phases, one to demonstrate the performance of the regular SOM which used the Euclidean distance and decreasing learning rate, and the other to demonstrate the performance of the proposed cyclic-SOM, followed by experiments on the PCA, MLP, and SVM. It’s worth mentioning that the results of this paper were computed after using the K-nearest neighbor (KNN) [8] classifier with three nearest neighbors. Only the MLP does not use it here.

6.2.1 Results for CMU-PIE Face Database

In the case of the CMU-PIE face database, the used faces were in five illuminations and two poses. It is worth mentioning that the proposed technique performs well due to the variety of feature data and the difficulty of changing the face pose or illumination in comparison with the other techniques in this paper. Fig.10 shows the firmness of the proposed cyclic-SOM by comparing its performance with the SOM after changing only the technique of finding the winner neuron by the MAD and regular SOM, MLP, PCA, and SVM.

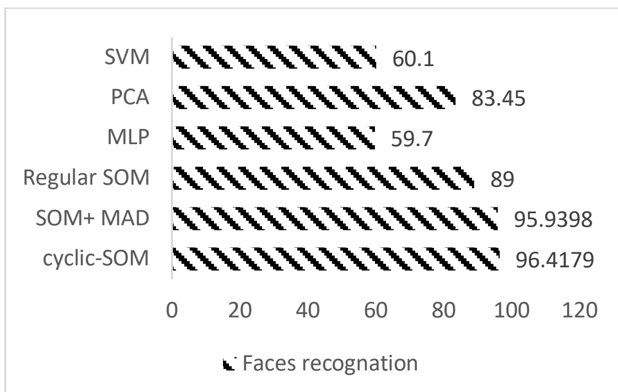


Fig. 10: Comparison between the proposed Cyclic-SOM performance with the SOM+MAD, regular SOM, MLP, PCA, and SVM in the faces recognition

The Cyclic-SOM has a high recognition rate for facial features, as shown in Fig.10. Its accuracy was 96.4179 % for Face recognition through pose and illumination. Before adjusting the learning rate to cyclic learning, we experimented on the SOM by changing only the way of finding the propinquity between the input and the feature map, or what's known as “the way of finding the winner neuron”. Its accuracy reached 95.9398 %. SOM+MAD used a learning rate of 0.95, which decreased over time in a fixed pattern until it reached 0 at the end of the training period, as in the regular SOM. This decrease equals 0.95 by the number of training periods. When we used the cyclic learning with the SOM which used MAD, we used a maximum learning rate equal to 0.95 and a minimum learning rate equal to 0.57 with length steps equal to 4 to reach this accuracy with the faces' database. Adding cyclic learning after the MAD improves the accuracy and speed of the proposed technique, as shown in Fig. 11. Adjustments in the learning rate and propinquity calculation enable the method to achieve full accuracy after 113 training periods. However, in the case of other techniques, a longer training duration of 400 times is needed, as opposed to the regular SOM, which has only 89 % accuracy in this state of the experiments. The identification rate was 59.7% when using the PCA, 83.54 % when using the MLP, and 60.1 % when using the SVM. Based on the previous results, we can conclude that the proposed technique responds to the complexities of feature data in faces under various illumination and poses with robust performance when used for object recognition.

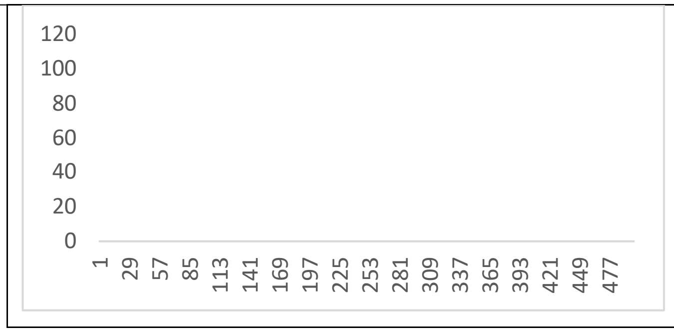


Fig. 11: the cyclic-SOM performance for face recognition

6.2.2 Results for AHDBase

After checking the cyclic-SOM technique on face recognition, we checked its firmness in the recognition of the Arabic digit using AHDBase. The written digits from different subjects come in a wide range of styles. What's made it a significant obstacle for any device to address. Digit recognition is one of the most common problems in the computer vision community. It has a wide variety of applications. Cyclic-SOM shows promising results in the Arabic recognition in comparison with common techniques like SVM, regular SOM, and also MLPs as shown in Fig.12. It's worth mentioning the cyclic SOM compared its performance with SOM after changing only the way of finding the winner neuron using the MAD.

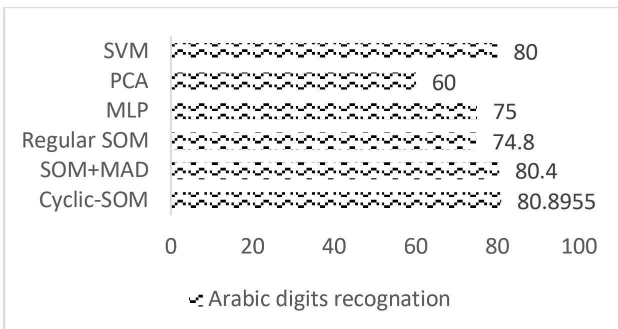


Fig. 12: Comparison between the proposed Cyclic-SOM performance with the SOM+MAD, regular SOM, MLP, PCA, and SVM in the recognition of the Arabic digit

Fig.12 shows the Arabic digit recognition rate of the Cyclic-SOM. Its accuracy equals 80.8955%. However, the experiment using the SOM+MAD gives an accuracy of 80.4% by comparing 74.8% using the regular SOM, 60% using PCA, 75% using MLP, and 80% using the SVM. In the Cyclic–SOM for the Arabic recognition, the maximum learning rate started was 1 and the minimum learning rate was 0.9 with a step size equal to 2. The learning rate in the regular SOM and SOM+MAD started at 1 and decreased to 0 in the final training period. As shown from the results, the proposed Cyclic-SOM technique presents better performance in the recognition of the Arabic digit. Fig.13 shows that the cyclic reached its firmness accuracy from the training period number101, however, the regular SOM needs more than 300 training periods for its accuracy.

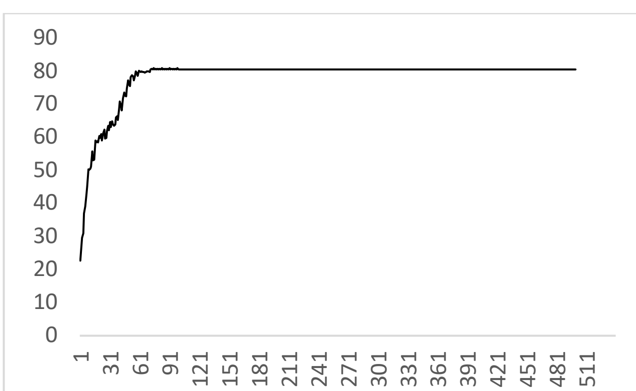


Fig. 13: The cyclic-SOM performance for Arabic digit recognition

6.2.3 Results for MNIST Database

English digit recognition is critical in the field of computer vision, as previously stated. As a result, we attempted to assess the proposed system's stability using both Arabic and English digits. Fig. 14 shows the response of the proposed cyclic-SM by comparing its performance with the SOM+MAD, regular SOM, MLP, PCA, and SVM in the recognition of the English digit.

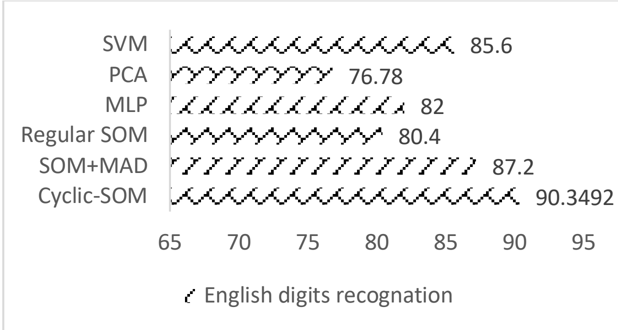


Fig. 14: Comparison between the Cyclic-SOM performance with the SOM+MAD, regular SOM, MLP, PCA, and SVM in the recognition of the English digit

The Cyclic-SOM had a recognition rate of 90.3492 %. This is done with a maximum learning rate of 0.95 and a minimum learning rate of 0.57. In this case, the cyclic-SOM has a step size of 150. Figure 15 shows the accurate results for the cyclic SOM from the 229-training period. By comparing 80.4 % using the regular SOM, 76.78 % using PCA, 82 % using MLP, and 85.6% using the SVM, the results of changing the way of measuring the propinquity from the input and the feature map with MAD in the SOM reached 87.2 %. In both the SOM+MAD and regular SOM, the learning rate began at 1 and gradually decreased to 0 as the training progressed. All of these tests show that the proposed method is reliable.

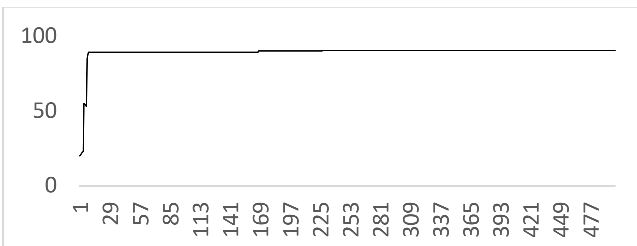


Fig. 15: The cyclic-SOM performance for English digits recognition

As demonstrated in the experimental results, our proposed technique shows promising results with a high recognition rate compared to other methods on different datasets. Fig. 16 illustrates the efficiency of the proposed technique compared to other methods using three different benchmark databases (AHDBase for Arabic digits, MNIST for English digits, and CMU-PIE for faces).

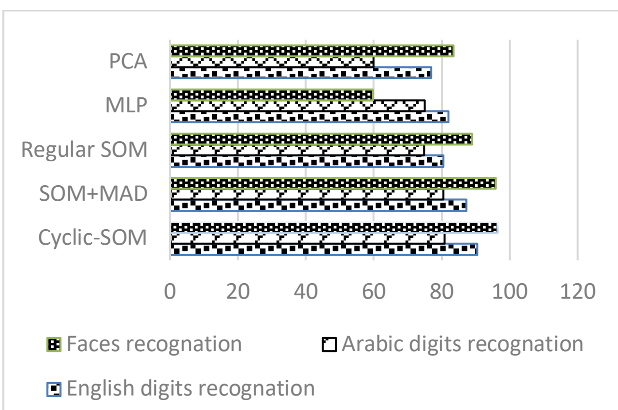


Fig. 16: Comparison between the proposed Cyclic-SOM performance with the regular SOM, MLP, PCA, and SVM in the objects recognition in this paper

6.3. Computational Cost

When training a neural network, one is usually interested in obtaining a network with optimal generalization performance. Typically, the generalization error is estimated by a validation error, i.e., the average error on a validation set, a fixed set of examples not from the training set. Computing the training cost and the validation error represent a significant computational effort because it requires additional passes over the training and validation data. A validation error is not the only measure to consider in selecting hyper-parameters. Often, one has to consider computational cost, either of training or prediction. Big Oh is an analysis algorithm that describes a function's behavior.

It also describes the implementation time of the algorithm analytically. Using the mean absolute difference, the big oh in the technique presented was O(DM) instead of O(DM^2) while using the Euclidean distance to find the best matching neuron. After using the cyclic learning rate, the time decreases to O(DM/N) once more. Where D represents the size of the map, M is the input size and N is the step size. This gives speed to the implementation of the algorithm.

7. Conclusion and Future Work

This paper presented a new optimization strategy called Cyclic-SOM for object recognition problems by improving the learning way of the self-organizing map (SOMs). The basic idea is based on a modification in the two main core functions of the regular SOMs which are the function of finding the winner neuron and the function of updating the weights. The proposed technique accelerates the learning process of the regular SOMs using a cyclic learning rate in the learning phase. The main core of cyclic-SOM is measuring the propinquity between the input and the feature map using mean absolute difference instead of traditional Euclidean distance and using cyclical learning rates (CLR) in the training phase of the SOM. Exchanging the distance of Euclidean with the mean absolute difference and using the CLR showed promising results compared to other methods (regular SOM, PCA, SVM, and MLP) on different datasets with a better recognition rate. The proposed Cyclic-SOM algorithm possesses the following merits: (1) it accelerates the learning process and eliminates the need to experimentally find the best values and schedule for the learning rates; (2) it offers one form of improvement in results or training improvement; (3) it requires no manual tuning of a learning rate and appears robust to noisy gradient information, different model architecture choices, various data modalities and selection of hyper-parameters; and (4) it is fast enough for an interactive environment. Consequently, it may achieve human-level performance and can be applied in a variety of applications. Our future work will focus on extending our proposed algorithm based on deep learning techniques.

Conflict of interest

The authors declare that there is no conflict regarding the publication of this paper.

References

[1] R. Jain, R. Kasturi, & B. G. Schunck. *Machine vision,* New York: McGraw-Hill, 5, 309-364, (1995).

[2] J. De. Vries . *Object recognition: a shape-based approach using artificial neural networks*, Department of Computer Science, University of Utrecht, (2006).

[3] Z. Liu, Z. Zhao, and M. Larson. *Who's afraid of adversarial queries? The impact of image modifications on content-based image retrieval*, In proc. of the international Conference on Multimedia Retrieval.(ICMR), 306-314, (2019).

[4] T. Kohonen . *Self-organizing maps,* Information Sciences, Berlin, Springer, 30, (2001).

[5] J. Tang, C. Deng, & G. B. Huang. *Extreme learning machine for multilayer perceptron*, IEEE transactions on neural networks and learning systems, 27(4), 809-821, (2015).

[6] M.Yelameli, B. Thornton, T. Takahashi, T. Weerkoon, Y. Takemura, & K. Ishii. *Support vector machine-based classification of seafloor rock types measured underwater using laser-induced breakdown spectroscopy*, In proc. of OCEANS 2016-Shanghai, IEEE , 1-4, (2016).

[7] S. Wold, K. Esbensen, & P. Geladi. *Principal component analysis*. Chemometrics and intelligent laboratory systems, 2(1-3), 37-52, (1987).

[8] Z. Xing, J. Pei, & E. Keogh, *A brief survey on sequence classification*. ACM Sigkdd Explorations Newsletter, 12(1), 40-48. (2010).

[9] L. N. Smith. *No more pesky learning rate guessing games*, CoRR, (2015).

[10] L. N. Smith. *Cyclical learning rates for training neural networks*, In Proc. of the IEEE winter conference on applications of computer vision (WACV) , 464-472, (2017).

[11] S. Joshi, D. K. Verma, G. Saxena, & A. Paraye. *Issues in training a convolutional neural network model for image classification*, In Proc. Of 3th International Conference of Advances in Computing and Data Sciences(ICACDS), Ghaziabad, India, 282-293, (2019).

[12] J. Bergstra, D. Yamins , & D. Cox. *Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures*. In Proc. of the International conference on machine learning, 115-123, PMLR. (2013).

[13] C. Nwankpa, W. Ijomah, A. Gachagan, & S. Marshall. Activation functions: Comparison of trends in practice and research for deep learning , ArXiv. /abs/1811.03378, (2018)

[14] Y. Bengio. *Practical recommendations for gradient-based training of deep architectures*, In Neural networks: Tricks of the trade , Springer, Berlin, Heidelberg , 437-478, (2012).

[15] L. Blier, P. Wolinski, , & Y. Ollivier. *Learning with random learning rates*, In Proc. of Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, Cham, 449-464, (2019)

[16] J. H.Chen, M. C. Su, R. Cao, S. C. Hsu, & J. C. Lu. *A self-organizing map optimization-based image recognition and processing model for bridge crack inspection*, Automation in Construction, 73, 58-66. (2017).

[17] F. Coleca, A. State, S. Klement, E. Barth, & T. Martinetz. *Self-organizing maps for hand and full body tracking*. Neurocomputing, 147, 174-184. (2015).

[18] T. Kinnunen, J. K. Kamarainen, L. Lensu, & H. Kälviäinen. *Unsupervised object discovery via self-organization*. Pattern Recognition Letters, 33(16), 2102-2112. (2012)

[19] B.Chakraborty, A.Menezes, S. Dandapath, W. A. Fernandes, S. M. Karisiddaiah, K. Haris, & G. S. Gokul. *Application of hybrid techniques (self-organizing map and fuzzy algorithm) using backscatter data for segmentation and fine-scale roughness characterization of seepage-related seafloor along the western continental margin of India,* IEEE Journal of Oceanic Engineering, 40(1), 3-14, (2014).

[20] L.Liu, B.Wang, & L. Zhang, *An approach based on a self-organizing map and fuzzy membership for decomposition of mixed pixels in hyperspectral imagery*, Pattern recognition letters, 31(11), 1388-1395, (2010) .

[21] A. N. Chabane, N. Islam, & B. Zerr. *Incremental clustering of sonar images using self-organizing maps combined with fuzzy adaptive resonance theory*, Ocean Engineering, 142, 133-144, (2017).

[22] P. Swaroop, & N. Sharma. *An overview of various template matching methodologies in image processing*, International Journal of Computer Applications, 153(10), 8-14. (2016).

[23] R. R. Yager, & N. Alajlan, *A note on mean absolute deviation*. Information Sciences, 279, 632-641 (2014).

[24] H. Sarmadi, A. Entezami, B. Saeedi Razavi, , & K. V. Yuen, *Ensemble learning-based structural health monitoring by Mahalanobis distance metrics*, Structural Control and Health Monitoring, 28(2), e2663, (2021).

[25] P. Stefanovič, & O. Kurasova. *Outlier detection in self-organizing maps and their quality estimation,* Neural Network World, 28(2), 105-117, (2018).

[26] T. Sim, S. Baker, & M. Bsat,. *The CMU pose, illumination, and expression (PIE) database*, In Proc. of the 5th IEEE International Conference on Automatic Face Gesture Recognition. IEEE, pp. 53-58, (2002).

[27] Z.Wang , & A. C. Bovik, *A universal image quality index*, IEEE signal processing letters, 9(3), 81-84. (2002).

[28] E. A.El-Sherif, , & S. Abdelazeem. *A Two-Stage System for Arabic Handwritten Digit Recognition Tested on a New Large Database*, In Artificial intelligence and pattern recognition, 237-242, (2007).

[29] L. Deng . *The ministry database of handwritten digit images for machine learning research [best of the web]*. IEEE Signal Processing Magazine, 29(6), 141-142, (2012)

[30] Y. LeCun, C. Cortes, C.J. Burges. *The MNIST Database of handwritten digits*, (2018), http://yann.lecun.com/exdb/mnist/