Deep Convolutional Self-Organizing Map Network for Robust Handwritten Digit Recognition

SALEH ALY1,2, (Associate Member, IEEE), AND SULTAN ALMOTAIRI3

1Department of Information Technology, College of Computer and Information Sciences, Majmaah University, Majmaah 11952, Saudi Arabia
2Department of Electrical Engineering, Faculty of Engineering, Aswan University, Aswan 81542, Egypt
3Department of Natural and Applied Science, Community College, Majmaah University, Majmaah 11952, Saudi Arabia

Corresponding author: Saleh Aly (s.haridy@mu.edu.sa)

This work was supported by the Deanship of Scientific Research at Majmaah University under Grant RGP-2019-24.

ABSTRACT Deep Convolutional Neural Networks (DCNN) are currently the predominant technique commonly used to learn visual features from images. However, the complex structure of most recent DCNNs impose two major requirements namely, huge labeled dataset and high computational resources. In this paper, we develop a new efficient deep unsupervised network to learn invariant image representation from unlabeled visual data. The proposed Deep Convolutional Self-organizing Maps (DCSOM) network comprises a cascade of convolutional SOM layers trained sequentially to represent multiple levels of features. The 2D SOM grid is commonly used for either data visualization or feature extraction. However, this work employs high dimensional map size to create a new deep network. The N-Dimensional SOM (ND-SOM) grid is trained to extract abstract visual features using its classical competitive learning algorithm. The topological order of the features learned from ND-SOM helps to absorb local transformation and deformation variations exhibited in the visual data. The input image is divided into an overlapped local patches where each local patch is represented by the N-coordinates of the winner neuron in the ND-SOM grid. Each dimension of the ND-SOM can be considered as a non-linear principal component and hence it can be exploited to represent the input image using N-Feature Index Image (FII) bank. Multiple convolutional SOM layers can be cascaded to create a deep network structure. The output layer of the DCSOM network computes local histograms of each FII bank in the final convolutional SOM layer. A set of experiments using MNIST handwritten digit database and all its variants are conducted to evaluate the robust representation of the proposed DCSOM network. Experimental results reveal that the performance of DCSOM outperforms state-of-the-art methods for noisy digits and achieve a comparable performance with other complex deep learning architecture for other image variations.

INDEX TERMS Deep convolutional self-organizing map, DCSOM, unsupervised feature learning, robust handwritten digit recognition.

I. INTRODUCTION

Learning hierarchies of features for image representation is a major goal for many computer vision and pattern recognition applications [1]. Most existing deep Convolutional Neural Networks (CNN) architectures are developed to learn supervised features while few unsupervised learning algorithms are emerged. Classical unsupervised feature learning approaches almost focus on exploiting the availability of unlabeled training data images to understand the underlying structure of data [2]–[4]. The learning algorithm of Self-Organizing Map (SOM) [5] can be considered as one of the most distinctive algorithms which employ neighborhood function to learn topology of the high-dimensional input space.

Unsupervised deep convolutional neural networks receive little attention as compared to the supervised CNN architectures. Sparse coding unsupervised learning [6] was employed to learn orientation-sensitive filters from natural images which are similar to Gabor filters in V1 area. Information generated by these filters are combined to develop a new
model for visual cortex [7]–[9]. Several unsupervised deep learning algorithms such as: K-means [10], PCANet [11], ScatNet [12] and SOMNet [13] are proposed to build a simple unsupervised deep network architecture.

The SOM is a neural network architecture trained using unsupervised competitive learning algorithm to map high-dimensional input space into low-dimensional map [5]. SOM can preserve the topological relationships of input data by applying neighborhood function in the update step of neuron weights. Self-organizing map is related to the classical Vector Quantization (VQ) which is used extensively in data clustering and visualization. SOM is different than the classical K-means algorithm in which it uses a specific neighborhood function to update the weights of neuron centroids. The elegant way used by SOM to update neuron weights leads to the creation of a topographically ordered feature map in which similar inputs are mapped to nearby neurons. This property is exploited in this work to map every local patch of the input image into the coordinates of the best matching unit of SOM grid. Thus, the input image can be mapped into multiple feature images according to the dimension of SOM grid. SOM can be considered as an approximation of non-linear principal component analysis algorithm. Each dimension of the SOM grid can be viewed as a non-linear principal component. The centroids of the neurons can be considered as an approximate of the non-linear data manifold lies in the high dimensional input space. A single convolutional feature learning layer using self-organizing map (SOM) was developed in [14] to realize this idea. Mahalanobis and Euclidean distance metrics are utilized in the 2-Dimensional SOM grid to learn topographic feature detectors from local patches of the input training images. Index of the winner neuron is exploited to represent each local patch. Our proposed DCSOM extends the work of the single convolutional SOM layer in [14] to learn deep network using high dimensional SOM convolutional layers.

A lot of research works have been developed to exploit the advantages of self-organizing map learning in deep architectures such as: Convolutional Recursive Modified Self-Organizing Map (CR-MSOM) [15], Convolutional Self-organizing map (CSOM) [16], self-organizing maps with convolutional layers [17], Denoising Autoencoder Self-Organizing Map (DASOM) [18], Unsupervised Deep Self-Organizing Map (UDSOM) [19], [20], Deep Product Self-Organizing Maps (DP-SOM) [21], [22]. However, our proposed Deep Convolutional Self-Organizing Map (DCSOM) model is completely different from the previously mentioned deep SOM architectures. All previous architectures utilized 2-dimensional SOM grid while none of them explored the impact of increasing the dimensionality of SOM grid. The UDSOM architecture proposed in [19], [20] utilized multiple SOMs to learn region-specific features. However, our proposed model trains only one convolutional SOM layer in each stage as contrary to other methods [20], [23], [24] which also trains multiple region-specific SOMs.

The proposed Deep Convolutional Self-organizing Map (DCSOM) network is a new deep learning architecture which uses multiple convolutional SOM layers. It has a similar structure with common convolutional neural networks. However, each convolutional SOM layer is trained separately after finishing the training of previous layers. The first step of the proposed method is to divide each input image into a set of overlapped local patches. These patches are then normalized using Z-score normalization by subtracting the patch mean and dividing by standard deviation. The normalized local patches of all training images are used to train N-dimensional SOM grid which considered as the first convolutional SOM layer. The employment of the neighborhood function in SOM competitive learning algorithm helps to organize local features of the training image patches in a topologically ordered feature maps. Then, the local patches of the input images are mapped into the coordinates of the winner neuron of its corresponding ND-SOM layer. N-multiple feature index images (FII) bank is generated for the input image in which each feature image corresponds to one coordinate of the SOM grid. Every feature index image considered as a mapping of the input image along each dimension of the ND-SOM layer. The second convolutional SOM layer is trained in similar way without applying Z-score normalization on the local patches derived from each feature index image of the previous layer. The number of FII banks in the second layer increases as we go deeper in the network. The previous process is repeated to learn more convolutional SOM layers and create more deeper network. Invariant feature representation can be simply obtained by applying block histogram computation for each feature index image bank resulted from the final convolutional SOM layer. The main contributions of this work can be summarized as follows:

- Developing a new deep architecture using multiple stages of convolutional SOM layers.
- Exploiting the topological order property of SOM feature map to efficiently represent local patches of the input image using SOM neuron coordinates.
- Employing high dimensional SOM maps rather than 2D to learn deep feature representation.
- Evaluation of the proposed DCSOM on various challenging handwritten digit variations.
- Achieving state-of-the-art performance on noisy handwritten digits.

The paper is organized as follows, section II describes the most recent related works to our proposed method. In Section III, we explain in details the proposed robust image representation method using DCSOM network. In Section IV, we show the results and experimental setup. The paper concludes with Section V.

II. RELATED WORKS

Unsupervised feature learning considered as an elegant solution in case of the lack/difficulty of labeling training data. Coates and Ng [10] successfully employed the simple
K-means clustering algorithm to build a simple unsupervised feature representation. They first normalized and whitened all local patches of the training images to learn different types of low-frequency edge-like features through K-means centroids. Image is then represented by finding the similarity between each local patch and all the K-centroids which generate very large number of response images. Feature representation is performed by spatially polling local features of the K response images. Despite the simplicity of this method, it is difficult to build an efficient deep network from this method due to the lack of topographical order of K-means centroids and its increasing number of response images.

The topological order of neurons in SOM grid make it unique and more appropriate for high-dimensional data visualization and feature representation. Lawrence et al. [25] combined self-organizing map (SOM) and convolutional neural network to solve face recognition problem. In their work, SOM is utilized to quantize local facial image patches into the topological space of SOM which reduce their dimensionality and increase robustness to the minor variations. The convolutional neural network is trained using mapped input face image to learn larger features. Another work conducted by Tan et al. [26] employed SOM for occlusion invariant face recognition. Similarly, they exploited the topological property of SOM to map local patches into the coordinates of winner neuron. A modified soft-KNN ensemble classifier was developed to robustly match two face images represented by the the SOM topological space. A hierarchical model based on multiple SOMs was developed in [27] for visual feature extraction using variable map dimension. In [28]–[30], SOM is combined with Gabor filters to build a hierarchical feature representation model for invariant face recognition. Zhao et al. [31] developed a stacked multilayer self-organizing map for background modeling. Sakkari et al. [19] used deep SOMs for automated feature extraction and classification from big data streaming. Nawaratne et al. [32] proposed a hierarchical two-stream growing self-organizing maps for human activity recognition. Convolutional SOM layer is employed in [33] to represent hand shapes in sign language recognition application.

Many variations of convolutional neural networks have been proposed to exploit various image transformation techniques using unsupervised learning. Wavelet scattering network (ScatNet) [12] utilized a prefixed wavelet operators as a convolutional filters without any need for learning. However, PCANet [11] employed Principle Component Analysis (PCA) algorithm to learn different filter bank levels and create a simple deep learning architecture by cascading this operation. Instead of learning PCA filters, DCTNet [34] was developed to employ learning-free filters using Discrete Cosine Transform (DCT). On the same research direction, Hankins et al. proposed SOMNet [13] to learn a set of non-orthogonal filters using self-organizing map (SOM). In their work, SOM is utilized to approximate the non-linear manifold of the input space through the discretized representation of neurons. The previous methods utilized a simple trick of binarization and block histogram computation to represent the final feature vector. However, when these network become more deeper, its computational resource complexity dramatically increases.

A two-layers SOM called Deep Self-Organizing Map (DSOM) was recently proposed by Liu et al. [23]. DSOM consists of alternative self-organizing and mapping layers. The self-organizing layer contains multiple number of local SOMs, where each SOM focuses on a local area of the input image. Index value of the winning neuron from each SOM in the first self-organizing layer is organized in the mapping layer to produce another 2D map. The new generated 2D map is used as input to the second self-organizing layer. DSOM model utilized a supervised learning version of SOM to solve MNIST handwritten digit classification problem. DSOM architecture has two limitations, first it utilized a supervised learning algorithm and thus affects the natural topographic order of SOM neurons. Second, this model learns a specific SOM for each local region of the input image which increase the redundancy and complexity of the network. DSOM was then extended in [24], [35] to rely on the original unsupervised SOM learning algorithm and to learn features at different resolution in the first convolutional layer. One major drawback of this model is its wider rather than deeper feature learning strategy. Our proposed Deep Convolutional Self-Organizing Map (DCSOM) differs from DSOM model in the sense that it utilizes the index of winner neurons to represent local patches rather than using winner distance values. In addition, it learns a single SOM map in each layer which make its computational cost less than that of DSOM.

In recent works, Wang et al. [22] integrated the Convolutional Neural Networks (CNN) and Self-Organizing Map (SOM) into a unified deep architecture. They employed supervised quantization objective to minimize the differences between two similar image pairs and maximize the differences between dissimilar image pairs on the map. This leads to jointly extract deep features and quantize the features into suitable neurons in self-organizing map. However, they trained several self-organizing maps for each input sub-space and used SOM for data visualization only. In another work, Sakkari and Zaied [20] proposed an architecture which alternate self-organizing, RELU rectification function, and abstraction layer. In their work, the self-organizing layer is composed of a multiple region-based SOMs with each map focuses on modeling a local sub-region of the input image. The most winning neurons of each SOM are then organized in a second sampling layer to generate a new 2D map and to extract the details of each set.

### III. DEEP CONVOLUTIONAL SELF-ORGANIZING MAP NETWORK

The proposed Deep Convolutional Self-Organizing Map (DCSOM) feature extraction method shown in Fig. 1 contains: input layer, local contrast normalization layer, multiple convolutional SOM Layers and output layer. Each ND-SOM...
convolutional layer accepts connection from its previous layer. In the first convolutional SOM layer, a set of overlapped local image patches of size $w \times w$ are collected from the input image $I$. Every local patch is normalized using Z-score normalization which subtract each patch from its mean and divide it by standard deviation. The normalized patches are then mapped into a pre-trained N-dimensional SOM feature map to find the coordinates of best winning neuron. The Euclidean distance between every local image patch and all neurons of the map is computed to find the position of the winner neuron. The topographic order of the SOM feature maps helps to map similar patches into nearby feature indexes. It is assumed that small deformation and distortions of local patches can be absorbed by the quantization process or the competitive learning algorithm of SOM. As contrary to the non-topological learning algorithm such as K-means, the order of codebook indexes has no meaning.

The process of creating convolutional SOM layer is repeated for the successive stages until the output layer is reached.

The next SOM convolution layer of the DCSOM learns larger features than the previous one. Due to the topological order of the features learned from previous SOM layer, similar patches are mapped into nearby neurons. Therefore, local patches extracted from the feature index images can be treated in similar way as local intensity patches come from the input image. Finally, feature representation is achieved by dividing each final feature index image bank into a set of overlapping blocks and calculating the local histogram of each block. A more detailed explanation of the network structure of DCSOM will be considered in the following subsections.

A. LOCAL CONTRAST NORMALIZATION LAYER

Before running the SOM learning algorithm on the collected local image patches, it is useful to standardize the brightness and contrast of the patches (known as Z-score). Z-score normalization is a standard preprocessing for many neural network-based methods. That is, for each input local patch of size $w \times w$ we subtract patch mean and then divide it by the standard deviation. The normalization layer is applied only for the input image while no need to apply it for the next SOM convolutional layer. Let us denote the unnormalized local patches as $\hat{x}_i$, the following equation is applied to normalize the patches:

$$x_i = \frac{\hat{x}_i - \text{mean}(\hat{x}_i)}{\sqrt{\text{var}(\hat{x}_i)} + \epsilon}$$

where $x_i$ are the normalized patches, $\text{mean}$, $\text{var}$ are the mean and variance of the unnormalized local patches, and $\epsilon$ is a very small number which used to avoid dividing by zero.

B. CONVOLUTIONAL SELF-ORGANIZING MAP LAYER

SOM [5] is considered as one of the best competitive learning algorithm which can be used to approximate the input space
with a set of fixed neurons. Also, the generated SOM can be viewed as a non-linear discretized approximation of principal component analysis algorithm, in which each map dimension considered as a nonlinear principal component. Most existing clustering algorithms did not consider the relationship between clusters while SOM utilizes a specific neighborhood function to organize its neuron centroids. This process is very helpful to learn a set of topologically ordered filters for the convolution layer of DCSOM. Traditionally, 2D maps are always trained and used for high-dimensional data visualization. In this work, we explore the effectiveness of using high dimensional SOM which help to capture the intrinsic dimension of the feature space.

C. BATCH LEARNING ALGORITHM

Batch learning algorithm has been developed to produce faster and more efficient results as compared to the sequential algorithms. Assume that we have training data \( X^{M \times D} = \{x_i | x_i \in R^D, i = 1, ..., M \} \), where \( D \) denotes the input dimension and \( M \) is the total number of samples. Also consider \( K \) and \( \alpha \) parameters are the number of neurons in the SOM map and the learning rate, respectively. \( \sigma \) represents radius of the neighborhood function. For each neuron, \( m_j \) is an associated mean vector where \( j = 1, ..., K \).

1) NETWORK INITIALIZATION

Generally, random vectors can be assigned as initial values for the \( K \) neurons of the SOM. However, to speed up the ordering and convergence of SOM training, linear initialization using regular N-dimensional hyperplane spanned by the N largest principal components of training data is recommended. For ND-SOM, the N principal components are computed using PCA algorithm while keeping only the eigenvectors with N largest eigenvalues. Practically, the selection of good initial values of neuron centroids speed up the convergence of the algorithms.

2) NEURON COMPETITION

Finding the best matching unit for each input sample is a key step in SOM learning algorithm. Every neuron in the SOM layer is associated with a mean vector in addition to its location in the map. The winner neuron \( c \) can be located by minimizing the Euclidean distance between input image patch \( x(t) \) and centers of all neurons in the SOM grid.

\[
c = \arg \min_i \| (x(t) - m_i(t)) \|^2 \quad (2)
\]

3) NETWORK UPDATE

For each iteration, all samples from the training data set \( X \) are used to update the map at once. Winning neuron \( c \) is computed for each sample using Eq.(2). The means vectors of all neurons \( (i = 1, ..., K) \) are updated as follows:

\[
m_i(t + 1) = \frac{\sum_j h_{ci}(x(t))}{\sum_j h_{ci}} \quad (3)
\]

where \( h_{ci} \) is the neighborhood function of the \( i^{th} \) neuron given the winning neuron \( c \). The winning neuron \( c \) and all its spatial neighbors in the map grid are modified as depicted in Eq. (3). The amount of changes for each neuron depends on the selected neighborhood function \( h_{ci} \). The neighborhood function has an important role in self-organization. Radial basis function with radius \( \sigma \) is a common choice for the neighborhood function.

\[
h_{ci} = \alpha(t)e^{(-|c-i|^2/\sigma^2(t))} \quad (4)
\]

where \( \alpha(t) \) is the a monotonically linear decreasing function of \( t \) which represent learning rate, \( c \) and \( i \) represent the coordinate of the \( c \)-winner neuron and \( i \)-neuron in the SOM map grid. Their dimensionality depends on the SOM map dimension (i.e for ND maps this value represent N coordinate values). The neighborhood value gradually decrease according to the neighborhood radius function \( \sigma(t) \), its value is calculated for each iteration \( t \) using the following equation.

\[
\sigma(t) = \sigma_i + \frac{t}{T} (\sigma_f - \sigma_i) \quad (5)
\]

where \( \sigma_i, \sigma_f \) are the initial and final radius parameter, respectively. \( T \) denotes the number of iterations. The neighborhood radius function \( \sigma(t) \) is a linear decreasing function of \( t \). The initial radius \( \sigma_i \) value is almost take large value at the beginning of training and gradually decreased to small value \( \sigma_f \) at the end of training. The topological order of the neurons is developed at the beginning of training, and converge to nearly optimal values at the end. However, in the image classification problems the objective is to keep features as discriminative as possible. To this end, the initial radius value should take a relatively small value while the final radius value may approach zero.

D. SOM MAPPING

We compute a representation \( F \) for every given \( w \times w \) image patch of the input image using learned convolutional SOM layer. Thus, we can define feature index image output representation of the input image by mapping all patches using function \( f(x) \). Given an input \( I \) of size \( n \times n \) pixels (and possibly d-channels), we calculate a set of \( (n-w+1) \times (n-w+1) \) feature index images with \( N \) channels. The representation \( F_{ij} \) for the \( ij \) patch of the input image take \( N \) coordinate values of the winner neuron. This hard quantization of the local patch strongly helps to reduce the effect of noise in the local patches as compared with other methods based on minimizing patch reconstruction error like PCANet.

We utilize the convolutional ND-SOM layer to map every input local image patch into the N coordinates of the winner neuron. The N coordinate values of the winner neuron is computed by changing the linear index of the winner neuron \( c \) into its corresponding SOM grid coordinates. The input image is converted into \( N \) feature index images (FII). Thus, each generated feature index image is related to a specific dimension of the learned ND-SOM. Local patches extracted from every \( N \) feature index images (FII) are considered as

VOLUME 8, 2020

107039
input to the next training session of the second convolutional SOM layer.

E. LEARNING SECOND CONVOLUTIONAL SOM LAYER
Learning subsequent SOM convolutional layers of DCSOM network can be accomplished by feeding the output of previous feature index images into the SOM training process. Simply, we apply the exact same learning algorithm to all patches collected from all FII to learn high-level features. Due to the topological order of the features learned from SOM, similarities between local patches can be handled in similar way as the input image (without any need for an intermediate local contrast normalization layers). By mapping each FII into the second ND-SOM, N bank feature index images are generated in which each bank contains N feature images. This method is valuable for building multiple layers of features for deeper network.

F. OUTPUT LAYER
Multiple convolutional SOM layers can be stacked together to further learn more complex features in the higher level. Feature representation can be simply achieved by applying overlapped block histogram computation in the output of the final convolutional SOM layer. Local spatial histograms are computed from each feature index image bank and then concatenated to generate the final feature vector. Employing local spatial histograms for representation increases the invariance of DCSOM network similar to PCANet. Given the final feature vectors and labels of all training images, we take the square root of each final feature vector and apply linear Support Vector Machine (SVM) classifier for image classification.

IV. EXPERIMENTAL RESULTS
This section focuses on the evaluation of various invariance aspects of DCSOM network using MNIST handwritten digit databases [36] and MNIST variations [37]. Most real handwritten digit recognition applications suffers from many geometric variations which are caused by mis-alignment, scale and rotation. Moreover, the existence of random noise may cause a sever problem in the recognition accuracy. In all experiments, we use two convolutional SOM feature extraction layers to examine the effectiveness of the DCSOM and to analyze the effect of changing its hyper-parameters. For classification, linear support vector machine classifier [38] based on LibSVM library implementation [39] is employed while features are extracted from either the first or second convolutional layers. We denoted DCSOM-1 and DCSOM-2 for features extracted from the first and second convolutional SOM layers, respectively. The default parameter values of linear SVM are used in all experiments as we are interested only on studing the impact of the new DCSOM feature extraction module.

A. EXPERIMENTS ON MNIST DATABASE
MNIST dataset considered as one of the most widely used benchmarks for deep convolutional neural network architectures. The MNIST dataset consists of gray-scale handwritten digits images from zero to nine of $28 \times 28$ size. The MNIST dataset contains 60000 samples commonly used as a training set and 10000 for testing. Fig. 2 shows sample images from MNIST dataset. In the following set of experiments, we examine the performance of proposed DCSOM feature extraction method by changing the most important hyper-parameters such as: dimension of SOM layer, patch size, block size, number of neurons and the initial neighborhood radius. It is worth noting that we utilized a subset from the MNIST dataset (6000 training samples and 1000 testing samples) to find the optimal values of DCOM parameters in the experiments from 1 to 5, while the whole dataset is
FIGURE 5. Feature extraction using two 2D-SOM convolutional layers of proposed DCSOM.

utilized in experiment 6 to compare the performance of our proposed DCSOM model with other state-of-the-art methods.

Experiment 1: The effect of varying the dimensionality of SOM grid is first examined in this experiment. A total of 256 neurons are organized into six different map dimensions (i.e. 1D, 2D, 3D, 4D, 5D and 6D) to generate six different map grids of size $(256 \times 1, 16 \times 16, 6 \times 6 \times 6, 4 \times 4 \times 4 \times 4, 3 \times 3 \times 3 \times 3 \times 3, \text{and } 2 \times 2 \times 2 \times 2 \times 2 \times 2)$, respectively. Example of various SOM map dimensions is shown in Fig. 3. The results of the experiment is shown in Fig. 4. It reveals that the recognition accuracy will be improved as the map dimension increase for both (DCSOM-1) and (DCSOM-2) features. For 1-dimensional map, the features extracted from the second DCSOM-2 layer give larger error rate than the first DCSOM-1 features, this is due to that 1D-SOM can not capture well the feature space of the input patches and hence badly affect the feature representation in the second layer. For higher dimension maps, CDSOM-2 features always give better performance than DCSOM-1 which confirm that selecting appropriate map dimension highly affect the recognition accuracy of the deep structure. For clarification, Fig. 5 shows an example of feature index images for the first and second SOM layers of digit five using 2-dimensional SOM.

Experiment 2: Secondly, we investigate the effect of changing the number of neurons along each map dimension of the best performing 4D map resulted from previous experiment. The number of neurons is changed from 2 to 7 in each dimension of the 4D map grid. Fig 6 shows the recognition accuracy with changing the number of neurons, the error rate is decreased as the number of neurons increase till 4 neurons. Since, SOM quantize the features space using discrete set of nodes, the feature space can be accurately quantized using 4 nodes in each dimension of the 4D SOM grid. The error rate increased when the number of neurons become greater than 4 which is indicator that SOM overfit data.

Experiment 3: This experiment examines the effect of changing the patch size of both convolutional layers simultaneously. The size of local patches is changed from $3 \times 3$ till $15 \times 15$ pixels. Fig 7 shows the error rate when changing the local patch sizes of both DCSOM-1 and DCSOM-2 layers simultaneously. The error rate is increased when the patch size increase. The optimal patch size is obtained at $5 \times 5$ pixels which indicates that small patch sizes is much better than larger patch sizes. These results depict that as the patch size increased we loss most of the discriminative features between classes.

Experiment 4: This experiment investigates the effect of changing the block size of the local histograms in the output layer. The local histogram block sizes is changed from $4 \times 4$ till $14 \times 14$ pixels. Fig 8 shows the error rate as block sizes change. The error rate is increased when the block size increase. The optimal block size is obtained at $6 \times 6$ pixels which indicates that using small block size is much better than larger ones. As we increased the block size, we loss the spatial
information and hence the feature representation become less discriminative.

**Experiment 5:** In this experiment, the effect of changing the initial neighborhood radius on the performance of the proposed DCSOM network is considered. The final ($\sigma_f$) value is fixed to zero in order to prevent updating the neighbor neurons and keep the most discriminant features at the final training iteration. The initial neighborhood radius ($\sigma_i$) is changed from 0 to 1. The experiment is conducted using 4D maps and fixed number of neurons ($4 \times 4 \times 4 \times 4$). Fig. 9 shows that the accuracy improves when the radius getting smaller for both DCSOM-1 and DCSOM-2 features. The choice of neighborhood radius also affects the topological structure of the neurons. Therefore, selecting an appropriate value of ($\sigma$) will help to preserve the topographic order of the neurons without affecting the discrimination between features. Example of ($8 \times 8$) 2D-SOM feature map learned from MNIST dataset is shown in Fig. 10. The map illustrate that SOM learning algorithm can successfully organizes neuron centers according to their similarities in the horizontal and vertical direction.

**Experiment 6:** We compare the accuracy of the proposed DCSOM-1 and DCSOM-2 with other state-of-the art methods. In PCANet [11], the number of PCA filters in the first and second stage are set to 8 and 8, respectively. The number of scales and orientations of wavelet filters in ScatNet were set to 3 and 8, respectively. In this experiment, the best values of the parameters obtained from the previous experiments are utilized. Table 1 summarizes the parameter values used in this experiment. The error rates of various state-of-the art methods are shown in Table 2. The results shown in this table are divided into three parts, the first part shows results of CNN-based methods, second part reports results of SOM-based methods while the third part shows results of our model. It is noted that the performance of DCSOM-2 using 4D SOM grid is comparable with other state-of-the-art methods while
beats the results of all SOM-based methods. As it is depicted in Table 2, the error rate of the proposed method is lower than all other methods except ConvNet [40] and ScatNet [12] models. One advantage of our method over ConvNet is its relying on the unsupervised feature extraction as compared with the stochastic gradient descent (SGD) used in ConvNet. In addition, using unsupervised feature learning module does not require large dataset for training. As contrary to ScatNet which depends on pre-fixed filter bank, DCSOM can directly learn features from data which enable it to generalize well and solve other object recognition tasks.

**B. EXPERIMENTS USING MNIST VARIATIONS DATASETS**

Multiple factors of variations on the basic MNIST dataset are introduced in [37] which leads to the generation four challenging datasets. Each of the dataset has 10000 and 50000 training and testing samples, respectively. Fig. 11 shows examples of the existing variations in each dataset which depict the difficulty of the classification task. The description of variations in each dataset can be summarized as follows:

- **MNIST-basic**: a smaller subset of the original MNIST dataset.
- **MNIST-rot**: digits are rotated using randomly selected rotation angle from 0 to $2\pi$.
- **MNIST-rand**: the digits are contaminated with a random background uniformly selected from 0 to 255.
- **MNIST-img**: A randomly selected patch from 20 gray images with high pixel variations was used as a background for digit image.
MNIST-rot: A combination of MNIST-rot and MNIST-img was used to perturb each digit image.

The parameters shown in Table 1 are used in the training of DCSOM network for various MNIST variations datasets classification. To tackle the severe variations exhibited in this datasets, we also compare the performance of two DCSOM models, one which is based on 3D-SOM grid and the other based on 4D-SOM grid. In addition, features of the first convolutional layer DCSOM-1 and the second convolutional layers DCSOM-2 are compared. Tacking features from either of the two layers and using two different SOM grid size (3D and 4D) creates four different models for comparison. Table 3 shows a comparison of the error rates with different methods on MNIST variations datasets. It is clear that, using 3D-SOM grid for DCSOM-2 features and 4D-SOM grid for DCSOM-1 features outperform state-of-the-art results in MNIST-rotation and MNIST-img datasets, respectively. These results prove that features of DCSOM are more robust to noise than other methods when we carefully choose the appropriate dimension of SOM grid. The hard quantization of the SOM mapping highly improve the representation of the noisy patches as compared to other CNN architecture which use either supervised or unsupervised learning. The obtained performance for the other MNIST variations datasets are also comparable with other state-of-the-art methods.

### V. CONCLUSIONS

In this paper, we propose a new unsupervised deep convolutional network based on self-organizing maps (SOM). The network utilizes multiple cascade of convolutional SOM layers to extract hierarchical features from training images. This work exploits high dimensional SOM maps which has more than two-dimension grid to extract and represent more rich features than the traditional 2D-SOM grid. N-dimensional SOM can be considered as a non-linear alternative for principal component analysis (PCA) with N components. N-dimensional SOM convolutional layers are trained sequentially using batch learning algorithm of SOM. Increasing the dimensionality of SOM maps helps to improve the performance of the network. The proposed deep architecture is unique and different from other deep networks which are based on SOM. The quantization process of the input image patches which depends on the winner neuron coordinates increases the robustness of the representation. This representation is different than that of PCANet and deep auto-encoder architectures that relies on minimizing the reconstruction error. The proposed DCSOM network achieves comparable results with other state-of-the-art methods while it achieves best performance on handwritten digits datasets contaminated with random and image background noise. Thus, DCSOM network can efficiently eliminate the effect of noise in the input image through its competitive learning algorithm and feature index image bank representation. However, other severe geometric variability like rotation can not be efficiently tackled with this architecture.

### ACKNOWLEDGMENT

The authors extend their appreciation to the Deanship of Scientific Research at Majmaah University for funding this work.

### REFERENCES

1. Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” Nature, vol. 521, no. 7553, p. 436, 2015.
2. K. Fukushima, “Neocognitron for handwritten digit recognition,” Neurocomputing, vol. 51, pp. 161–180, Apr. 2003.
3. G. E. Hinton, S. Osindero, and Y. W. Teh, “A fast learning algorithm for deep belief nets,” Neural Comput., vol. 18, no. 7, pp. 1527–1554, Jul. 2006.
4. Y. Bengio, A. Courville, and P. Vincent, “Representation learning: A review and new perspectives,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 35, no. 8, pp. 1798–1828, Aug. 2013.
5. T. Kohonen, “Essentials of the self-organizing map,” Neuro Netw., vol. 37, pp. 52–65, Jan. 2013.
6. B. A. Olshausen and D. J. Field, “Sparse coding with an overcomplete basis set: A strategy employed by v1?” Vis. Res., vol. 37, no. 23, pp. 3311–3325, Dec. 1997.
7. M. Riesenhuber and T. Poggio, “Hierarchical models of object recognition in cortex,” Nature Neurosci., vol. 2, no. 11, pp. 1019–1025, Nov. 1999.
8. T. Serre, L. Wolf, and T. Poggio, “Object recognition with features inspired by visual cortex,” in Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2005, pp. 994–1000.
9. J. Mutch and D. G. Lowe, “Object class recognition and localization using sparse features with limited receptive fields,” Int. J. Comput. Vis., vol. 80, no. 1, pp. 45–57, Oct. 2008.
10. A. Coates and A. Y. Ng, “Learning feature representations with K-means,” in Neural Networks: Tricks of the Trade. Berlin, Germany: Springer, 2012, pp. 561–580.
11. T.-H. Chan, K. Jia, S. Gao, J. Lu, Z. Zeng, and Y. Ma, “PCANet: A simple deep learning baseline for image classification?” IEEE Trans. Image Process., vol. 24, no. 12, pp. 5017–5032, Dec. 2015.
12. J. Bruna and S. Mallat, “Invariant scattering convolution networks,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 35, no. 8, pp. 1872–1886, Aug. 2013.
13. R. Hankins, Y. Peng, and H. Yin, “SOMNet: Unsupervised feature learning networks for image classification,” in Proc. Int. Joint Conf. Neural Netw. (IJCNN), Jul. 2018, pp. 1–8.
14. S. Aly, “Learning invariant local image descriptor using convolutional Mahalanobis self-organising mapping,” Neurocomputing, vol. 142, pp. 239–247, Oct. 2014.
15. E. Mohebi and A. Bagirov, “A convolutional recursive modified self organizing map for handwritten digits recognition,” Neural Netw., vol. 60, pp. 104–118, Dec. 2014.
16. H. Dozono, G. Niina, and S. Araki, “Convolutional self organizing map,” in Proc. Int. Conf. Comput. Sci. Comput. Intell. (CSCI), Dec. 2016, pp. 767–771.
17. L. Elend and O. Kramer, “Self-organizing maps with convolutional layers,” in Advances in Self-Organizing Maps, Learning Vector Quantization, Clustering and Data Visualization, A. Vellido, K. Gibson, C. Angulo, and J. D. Martín Guerrero, Eds. Cham, Switzerland: Springer, 2020, pp. 23–32.
[18] C. Ferles, Y. Papanikolaou, and K. J. Naidoo, “Denoising autoencoder self-organizing map (DASOM),” Neural Netw., vol. 105, pp. 112–131, Sep. 2018.
[19] M. Sakkari, R. Ejbali, and M. Zaied, “Deep SOMs for automated feature extraction and classification from big data streaming,” in Proc. 9th Int. Conf. Mach. Vis. (ICMV), Mar. 2017, Art. no. 103412L.
[20] M. Sakkari and M. Zaied, “A convolutional deep self-organizing map feature extraction for machine learning,” Multimedia Tools Appl., early access, Mar. 6, 2020, doi: 10.1007/s11426-020-08822-9.
[21] M. Wang, W. Zhou, Q. Tian, J. Pu, and H. Li, “Deep supervised quantization by self-organizing map,” in Proc. ACM Multimedia Conf. (MM), 2017, pp. 1707–1715.
[22] M. Wang, W. Zhou, Q. Tian, and H. Li, “Deep scalable supervised quantization by self-organizing map,” ACM Trans. Multimedia Comput., Commun., Appl., vol. 15, no. 3, pp. 1–18, Sep. 2019.
[23] N. Liu, J. Wang, and Y. Gong, “Deep self-organizing map for visual classification,” in Proc. Int. Joint Conf. Neural Netw. (IJCNN), Jul. 2015, pp. 1–6.
[24] C. S. Wickramasinghe, K. Amarasinghe, and M. Manic, “Deep self-organizing maps for unsupervised image classification,” IEEE Trans. Ind. Informat., vol. 15, no. 11, pp. 5837–5845, Nov. 2019.
[25] S. Lawrence, C. L. Giles, A. Chung Tsoi, and A. D. Back, “Face recognition: A convolutional neural-network approach,” IEEE Trans. Neural Netw., vol. 8, no. 1, pp. 98–113, 1997.
[26] X. Tan, S. Chen, Z.-H. Zhou, and F. Zhang, “Recognizing partially occluded, expression variant faces from single training image per person with SOM and soft-KNN ensemble,” IEEE Trans. Neural Netw., vol. 16, no. 4, pp. 875–886, Jul. 2005.
[27] S. Aly, N. Tsuruta, R.-I. Taniguchi, and A. Shimada, “Visual feature extraction using variable map-dimension hypercolumn model,” in Proc. IEEE Int. Joint Conf. Neural Netw., IEEE World Congr. Comput. Intell., Jun. 2008, pp. 845–851.
[28] S. Aly, A. Shimada, N. Tsuruta, and R.-I. Taniguchi, “Robust face recognition using multiple self-organized Gabor features and local similarity matching,” in Proc. 20th Int. Conf. Pattern Recognit., Aug. 2010, pp. 2909–2912.
[29] S. Aly, N. Tsuruta, and R.-I. Taniguchi, “Self-organized Gabor features for pose invariant face recognition,” in Neural Information Processing, Berlin, Germany: Springer, 2009, pp. 733–742.
[30] S. K. Aly and R.-I. Taniguchi, “On face recognition using hierarchical self-organized Gabor features,” in Proc. MVA, 2009, pp. 475–478.
[31] Z. Zhao, X. Zhang, and Y. Fang, “Stacked multilayer self-organizing map for background modeling,” IEEE Trans. Image Process., vol. 24, no. 9, pp. 2841–2850, Sep. 2015.
[32] R. Nawaratne, D. Alahakoon, D. De Silva, H. Kumara, and X. Yu, “Hierarchical two-stream growing self-organizing maps with transience for human activity recognition,” IEEE Trans. Ind. Informat., early access, Dec. 4, 2019, doi: 10.1109/TII.2019.2954754.
[33] S. Aly and W. Aly, “DeepArsSLR: A novel signer-independent deep learning framework for isolated arabic sign language gestures recognition,” IEEE Access, vol. 8, pp. 83199–83212, 2020.
[34] C. J. Ng and A. B. J. Teoh, “DCTNet: A simple learning-free approach for face recognition,” in Proc. Asia–Pacific Signal Inf. Process. Assoc. Annul. Conf. (APSIPA), Dec. 2015, pp. 761–768.
[35] C. S. Wickramasinghe, K. Amarasinghe, and M. Manic, “Parallalizable deep self-organizing maps for image classification,” in Proc. IEEE Symp. Ser. Comput. Intell. (SSCI), Nov. 2017, pp. 1–7.
[36] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” Proc. IEEE, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
[37] H. Larochelle, Y. Bengio, J. Louradour, and P. Lamblin, “Exploring strategies for training deep neural networks,” J. Mach. Learn. Res., vol. 10, no. 1, pp. 1–40, Jan. 2009.

[38] V. Vapnik, The Nature of Statistical Learning Theory. New York, NY, USA: Springer, 1999.
[39] C.-C. Chang and C.-J. Lin, “LIBSVM: A library for support vector machines,” ACM Trans. Intell. Syst. Technol., vol. 2, no. 3, pp. 1–27, 2011.
[40] K. Jarrett, K. Kavukcuoglu, M. A. Ranzato, and Y. LeCun, “What is the best multi-stage architecture for object recognition?” in Proc. IEEE 12th Int. Conf. Comput. Vis., Sep. 2009, pp. 2146–2153.
[41] K. Yu, T. Zhang, and Y. Gong, “Nonlinear learning using local coordinate coding,” in Proc. Adv. Neural Inf. Process. Syst., vol. 22, 2009, pp. 2223–2231.
[42] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, “Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations,” in Proc. 26th Annu. Int. Conf. Mach. Learn. (ICML), 2009, pp. 609–616.
[43] S. Rifai, P. Vincent, X. Muller, X. Glorot, and Y. Bengio, “Contractive auto-encoders: Explicit invariance during feature extraction,” in Proc. 28th Int. Conf. Int. Conf. Mach. Learn. Madison, WI, USA: Omnipress, 2011, pp. 833–840.
[44] G. Qian and L. Zhang, “A simple feedforward convolutional conceptor neural network for classification,” Appl. Soft Comput., vol. 70, pp. 1034–1041, Sep. 2018.

S. Aly, S. Almotairi: DCSOM for Robust Handwritten Digit Recognition

SALEH ALY (Associate Member, IEEE) received the B.Sc. and M.Sc. degrees in electrical and computer engineering from Assiut University, Assiut, Egypt, in 1997 and 2004, respectively, and the Ph.D. degree from the Department of Intelligent Systems, Graduate School of Information Science and Electrical Engineering, Kyushu University, Japan, in 2010. He is currently an Associate Professor with the Department of Information Technology, College of Computer and Information Sciences, Majmaah University, and on a leave from the Electrical Engineering Department, Faculty of Engineering, Aswan University, Egypt. In 2012 and 2014, he was a Visiting Researcher with the Department of Media Science, Nagoya University, and the Department of Computer Science, Tsukuba University, Japan, respectively. His research interests include deep learning, neural networks, pattern recognition, image processing, machine learning, and computer vision.

SULTAN ALMOTAIRI received the B.Sc., M.Sc., and Ph.D. degrees in computer science from the Florida Institute of Technology, Melbourne, USA, in 2010, 2012, and 2014, respectively. He is currently an Associate Professor with the Department of Natural and Applied Sciences, Community College, Majmaah University. He has acted as the Dean of the Community College, Majmaah University, since June 2015. In 2016, he was elected as the Chairman of the Municipality Council of Al-Majmaah, Saudi Arabia. His research interests include neural networks, deep learning, pattern recognition, machine learning, image processing, and computer vision.