



A deep contractive autoencoder for solving multiclass classification problems

Muhammad Aamir¹ · Nazri Mohd Nawawi² · Fazli Wahid³ · Hairulnizam Mahdin¹

Received: 4 January 2019 / Revised: 21 May 2019 / Accepted: 7 May 2020 / Published online: 10 June 2020
© Springer-Verlag GmbH Germany, part of Springer Nature 2020

Abstract

Contractive auto encoder (CAE) is one of the most robust variants of standard Auto Encoder (AE). The major drawback associated with the conventional CAE is its higher reconstruction error during encoding and decoding process of input features to the network. This drawback in the operational procedure of CAE leads to its incapability of going into finer details present in the input features by missing the information worth consideration. Resultantly, the features extracted by CAE lack the true representation of all the input features and the classifier fails in solving classification problems efficiently. In this work, an improved variant of CAE is proposed based on layered architecture following feed forward mechanism named as deep CAE. In the proposed architecture, the normal CAEs are arranged in layers and inside each layer, the process of encoding and decoding takes place. The features obtained from the previous CAE are given as inputs to the next CAE. Each CAE in all layers is responsible for reducing the reconstruction error thus resulting in obtaining the informative features. The feature set obtained from the last CAE is given as input to the softmax classifier for classification. The performance and efficiency of the proposed model has been tested on five MNIST variant-datasets. The results have been compared with standard SAE, DAE, RBM, SCAE, ScatNet and PCANet in terms of training error, testing error and execution time. The results revealed that the proposed model outperforms the aforementioned models.

Keywords Deep auto encoder · Contractive auto encoder · Feature reduction · Classification · MNIST variants

1 Background and context

Auto-encoders (AEs) are unsupervised neural networks that apply back propagation behaviour, setting up the high dimensional input feature set into a low dimensional output feature set and then recover the original feature set from output [1]. The reduction procedure of high dimensional data to low dimensional data is known as encoding while

the reconstruction of original data from low dimensional data is called decoding. Auto-encoder (AE) was proposed to improve the reconstruction reliability of low dimensional feature sets. AEs have the following main sub-models: sparse auto-encoders (SAEs) [2], denoising auto-encoder (DAEs) [3], laplacian regularized auto-encoder (LAE) [4, 5], coupled deep auto-encoder (CDA) [6], hessian regularized sparse auto-encoders (HSAE) [7], nonnegativity constraints auto-encoders (NCAE) [8], multimodal deep auto-encoder (MDA) [9], Bayesian auto-encoder (BAE) [10] and contractive auto-encoder (CAEs) [11].

In SAE, a sparsity factor is added to the original input nodes. There are two major ways for the representation of sparsity in AEs: firstly, penalization of the hidden layers biasness and secondly, the outputs of the hidden layers are used directly for the purpose of penalization. When solving numerical optimization problems, the weights are compensated by this penalty bias that leads to degrading the performance and efficiency of the algorithm. This is one of the important and worth attention drawbacks of the conventional AEs. For the reconstruction of noisy data from the original data, the authors

✉ Fazli Wahid
fazli.wahid@cs.uol.edu.pk

Muhammad Aamir
amir.khan1@gmail.com

Nazri Mohd Nawawi
nazri@uthm.edu.my

¹ Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn, Batu Pahat, Malaysia

² Soft Computing and Data Mining Center, Universiti Tun Hussein Onn, Batu Pahat, Malaysia

³ Faculty of Computer Science and Information Technology, The University of Lahore, Gujrat Campus, Gujrat, Pakistan

in [12] proposed DAEs. The DAEs are used for two purposes: firstly, encoding the noisy data and secondly, recovering the original input data from the reconstructed output data. When the data encoders are stacked in different layers, they form stacked DAEs. For minimizing the classification error, an extra layer is used by stacked DAEs. Stacked AEs are so named due to the phenomenon of putting the conventional AEs in a stack as highlighted by [13]. However, in this architecture, the layers are arranged in sequential order rather than parallel.

In order to enhance the performance of state-of-art AEs, many researchers have presented their additive to the AEs. e.g. In order to stop the locality preserving property for the data points of AEs, in LAE the author introduced some modification by adding the Laplacian regularization penalty to standard AE [5]. Also [7] for the purpose of reserving local structure data points in learning strategies, Hessian regularization has been added to SAE and formed HSAE which improves the robustness to noise along with sparse constraint. In [14] the author added non-negativity constraints to SAE on weight matrices to form NCAE, in order to improve the input features data reconstruction and enhancing the ability to disentangling the input hidden points geometry. While [15] presented MDA, which comprise of three stages, the first stage consists of an AE that is able to learn the internal structure of data points for 2D images. The second stage is based on two layered neural network for transforming 2D images to 3D representation and the third stage is again AE for learning the hidden data points for 3D poses. Furthermore, CDA proposed a Bigdata driven architecture which comprises two auto encoders having the ability to learn intrinsic features by extracting the hidden feature set from low resolution and high resolution image patches [6]. In [16] the author applied BayesianNet to standard autoencoder for building a multi layered BayesNet called adversarial variational Bayes auto encoder (BAE). The main purpose of BayesNet architecture in BAE is conditional probabilities adjustment for better prediction. BAE performs its operation as like BayesNet, therefore it learns the feature using belief propagations. [17] introduce multimodal video classification framework. They also performed a two-stage training architecture for learning a set of mapped latent features that capture both intra-modal and inter-modal semantics. In first stage they trained separate SCAE on three different features vectors that are audio, video and text extracted from the video. And in second stage they combine all modalities together to learns a multimodal stacked contractive auto encoders (MSCAE). In order to increase the robustness of standard AE, the researcher come up with Contractive Autoencoders (CAE). CAE are considered as the extended forms of the DAEs in which contractive penalty is added to the error function of the reconstruction. This penalty is, in turn, used for penalizing the attribute sensitivity in the input variations. The major drawback associated with CAEs penalty is its consideration for the minuscule diversity of the input data values. The authors in [18] addressed this issue

but failed to fully resolve the problem and there is still much room for further improvement.

2 Related work

In advance classification methods and models, the interconnected network of artificial neuron, called artificial neural network is of high interest among researchers [19–21]. Each neuron in the network describes a feature and the deep layers in the network present more essential features as compared to the previous layer. The exceeding number of feature results a complex network [22, 23], therefore researchers present AEs for reducing the features set from a high dimensionality to a lower one. While in this AEs based feature reduction models, there is a loss of informative features. To solve this issue, many researchers have provided their efforts and ideas in the literature and proposed some variants of AEs. These AEs variants are used in different domains of classification for example, in [13] the authors used sparse multi layered auto encoder framework for auto-detection of nuclei on a set of 537 marked histopathological breast cancer images. The input data is splitted in two subgroups: training (37 images) and testing (500 images). The whole architecture consists of 1 input layer, 2 hidden layers and 1 output layer. They stacked the sparse Autoencoder in their work and used Softmax for classification. There were 3468 nodes as input to the input layer, 400 nodes in the first hidden layer and 255 nodes in the second hidden layer. The output from second hidden layer was an input to the final Softmax layer, which is mapped to two classes, either 1 or 0. Further they perform comparative analysis of Stacked Sparse Autoencoder with Color Convolution, Convolution Neural Network and Expectation maximization based nuclei auto-detection.

The authors in [12] has adopted a weighted reconstruction loss function to the conventional DAE for noise classification in speech enhancement system. They stacked several weighted DAEs to construct the model. In their experiments, they performed 50 steps with the number of input nodes from 50 to 100. An unnoisy data comprises of 8 languages and a white noise having SNR of 6, 12 and 18 dB is selected from NTT database is selected for the purpose of model training. Their model was trained by 1hour length of data and was tested on a data of 8min length. The author in [4] applied a regularized function framework in learning feature to AE, in order to enhance the locality-preserving of input data points on the manifold during encoding and named it LAE. They used benchmark datasets MINST and CIFAR for recognition of hand-written digits and object recognition respectively. In MINST they used 50,000 images for training while 10,000 for validation and 10,000 for testing purpose. In CIFAR-10, they used 50,000 images for training object recognition model and 10,000 for testing of their model. There were 3468 input nodes as input to the model and 10 output classes.

In [6], the authors proposed CDA, based on an individual architecture which learns the intrinsic representations of low-resolution and high-resolution image patches for single image super-resolution simultaneously. This research used two datasets, that are Multi-PIE and CHUNK face Sketch FERET dataset for validation. They modified all the images in both dataset to 64x80 pixels before preprocessing. There were 7 comparative experiments conducted on Multi-PIE dataset for evaluating face recognition in different poses and 8 experiments on CHUNK face Sketch FERET dataset for drawing sketch with shape exaggeration.

Liu et al. [7] proposed Hessian Regularized Sparse Auto-Encoders (HSAE) for enhancing the internal geometry of auto encoder and robustness to noise using Hessian regularization and sparse constraint respectively. This research also stacked the HSAE to form deep architecture for better performance. It performed classification experiments using MNIST and CIFR-10 benchmark datasets. The number of input features are 1024 and the output layer has 10 classes. The performance evaluation parameter considered in the experimentation was the classification accuracy which was found to outperform the basic auto-encoder, Laplacian auto-encoder, sparse auto-encoder and Hessian auto-encoder when applied to the same datasets.

Chorowski et al. [8] applied a non-negative weights constraint to basic auto-encoder and form a deep architecture called NCAE. Firstly the non-negative weights constraint was applied to unsupervised training part of auto-encoder and the secondly it was applied to stage of fine tuning. They conducted the experiments based on classification using MNIST set of hand written digits, small NORB set of object images, Reuters-21578 text corpus set and ORL set of images benchmark datasets. Reconstruction error, sparseness of hidden encoding and part-based representation of features, in unsupervised learning phase were used as performance metrics in this work. The practical experiments showed that NCAE outperformed Nonnegative Sparse Autoencoder (NNSAE), Dropout Autoencoder (DpAE), Denoising Autoencoder (DAE), and Sparse Autoencoder (SAE).

In addition, the authors applied CAEs for the purpose of video semantics classification by introducing a two phases learning framework based on CAE [17]. In order to learn the discriminative feature set by stacking CAE for representing multi-modal fine tuning from single-modal pre-training, they put inter-modal and intra-model semantics under importance. To verify and validate their model, they stacked all the comparative models based on multi-modal fine and tuning and single-modal pre-training same as like their own multi-modal SCAE. In the first phase of their model, image representation was reduced from 546 to 128, text representation reduced from 1285 to 128 and audio representation was reduced from 38 to 20, using a two layered architecture. In the second stage all the adjacent connected feature

was optimized from 276 to 128 collectively while in the 3rd layer to 32 and 16 finally. The final evaluation explained that the multi-modal SCAE outperform support vector machine (SVM) and Linear Discriminant Analysis (LDA) [24] based on 10-fold cross validation.

3 The proposed model

Features reduction is one of the most important steps in solving big data problem and dealing complex data with high dimensional attributes. In fact the feature reduction is necessarily valuable method for any classification and prediction model [25, 26]. It is actually the better way for splitting the useful and effective features from ineffective and useless feature within the features representation space [27]. However, if there are irrelevant raw feature given as input may cause failure of better feature reduction and also results inefficient classification. In the proposed deep contractive auto encoders (D-CAE), three parallel CAEs are used for feature reduction. All the CAEs are arranged in layers and trained in a feedforward nature for minimizing the objective function in order to minimize the reconstruction error. The minimization of objective function and reconstruction error results in reducing the classification error. This section has been divided into three phases. The first phase explains the working mechanism of auto encoder. Secondly, the contractive auto encoder operational steps are explained and the last phase presents the overall work flow of the proposed three layered deep contractive auto encoder.

1 Conventional CAE

1: Parameters initialization

- No of hidden layers: h
- Input feature set: $[x_1, x_2, x_3, \dots, x_n]$
- Encoding activation function: EAF
- Decoding activation function: DAF
- Inputs weights: W_i
- Biasness values: b_i

2: Encoding

- Compute encoded inputs $f(w)$ by Multiplying x_n and W_i

3: Decoding

- Compute decoded outputs $f(w^0)$ by Multiplying x_n and W_i

4: Optimization

- Optimize value of equation 6
-

The internal structure of a conventional autoencoder is similar to a standard neural network with three layers. The autoencoder's whole processing takes place in two parts: encoding and decoding. The process of encoding and decoding take place in all layers, i.e layer-1, layer-2 and layer-3. In

order to clarify the structure of each layer in the proposed D-CAE, it is only highlighted in first layer. All the proceeding layers follow the same architecture as shown in Fig. 1.

3.1 Encoding

The process of mapping the input feature set to transform it to give as intermediate representation to the hidden layer is called encoding. Mathematically the process of encoding is given by Eq. 1.

$$y = f(x) = \phi(W_x + b_h) \quad (1)$$

where $f(x)$ represents the outputs of input layer which is given as inputs to the hidden layer h . ϕ represents the encoding activation function W represents weights given to each input and b represent the biasness value associated with input feature set.

3.2 Decoding

The process of mapping the output of the hidden layer back into the input feature set is called decoding. Mathematically the process of decoding is given by Eq. 2.

$$r = f(y) = \phi^o(W_y + b_r) \quad (2)$$

where $g(x)$ represents the output of hidden layer, W represent the weights given to the inputs of the hidden layer, b represents biasness of the inputs to the hidden layer.

The ϕ and ϕ^o are the encoding and decoding activation functions respectively, and are given by Eq. 3 for nonlinear representation (sigmoid function) whereas by Eq. 4 for the linear representation (hyperbolic tangent function).

$$\text{Sigmoid}(x) = 1/(1 + e^{-x}) \quad (3)$$

$$\tanh(x) = (e^x - e^{-x})/(e^x + e^{-x}) \quad (4)$$

The major aim of the reconstruction is to generate the outputs as much similar to the original inputs as possible by reducing the reconstruction error. The following parameter set is used to reconstruct the original inputs by reconstruction layer.

$$\Theta = [W, b_h, b_r] \quad (5)$$

Suppose, we have the input feature set as: $D_i = [x_1, x_2, x_3, \dots, x_n]$, then the reconstruction error is minimized by minimizing the following cost function

$$JAE(\Theta) = \sum_{x \in D_i} R(x, r) \quad (6)$$

where R is reconstruction error. Since minimization take place at the decoding phase of conventional CAE, it is applied all of the three layers in D-CAE. In case of linear

representation, it is the Euclidian distance whereas in case of nonlinear representation, it is the cross entropy loss. In order to avoid overfitting and penalizing the large weights rose from Eq. 6, the simplest form of Eq. 6 is Eq. 7:

$$JAE - wd(\theta) = \sum_{x \in D_i} R(x, r) + 1/2(\lambda||W||)2/2 \quad (7)$$

In which the relative importance of regularization is controlled by weight coefficient decay λ . Based on Eq. 6 and Eq. 7, the contractive auto encoder becomes Eq. 8.

$$JCAE(\theta) = \sum_{x \in D_i} R(x, r) + 1/2(\lambda||f(x)||)2/F \quad (8)$$

Where, $f(x)$ represents Jacobian matrix of encoder f at x . In case of D-CAE, Eq. 7 can be formulated as for n CAE.

$$JCAE(\theta) = \sum_{i=1}^v \sum_{x \in D_i} R(x, r) + 1/2(\lambda||f(x)||)2/F \quad (9)$$

The complete of structure of the conventional AE is given in Algorithm 1, while the algorithmic structure of proposed model is shown in Algorithm 2. The overall workflow and network architecture of the proposed model is shown in Fig. 1. There are three layers in the proposed model, each layer is itself a single CAE. In Fig. 1, the internal architecture of first layer is divided into encoder and decoder. Separately layer 2 and layer 3 are also CAEs, having the same architecture as layer 1 followed by the Softmax layer that is last layer for the purpose of final classification. In order to minimize the reconstruction error, the optimization has been applied to each layer of CAE based on Equation 6.

4 Experimental results

In this section, we conducted the experiments of the proposed D-CAE and other comparative models. All of the considered models for this experiments are evaluated on 5 benchmark variant datasets of MNIST. 5000 images from each of MNIST variant-dataset are randomly selected to validate the proposed D-CAE model. The variation datasets of MNIST are small subset (basic), random rotation digits (rot), random noise background digits (bg-rand), random background digits (bg-img) and rotation & image background digits (bg-img-rot) [28]. 6 random images from each MNIST variant dataset are shown in Figs. 2, 3, 4, 5 and 6 respectively. The experiments were performed on all of the above mentioned datasets with two phases in order to validate the D-CAE using different ratios of training data and testing

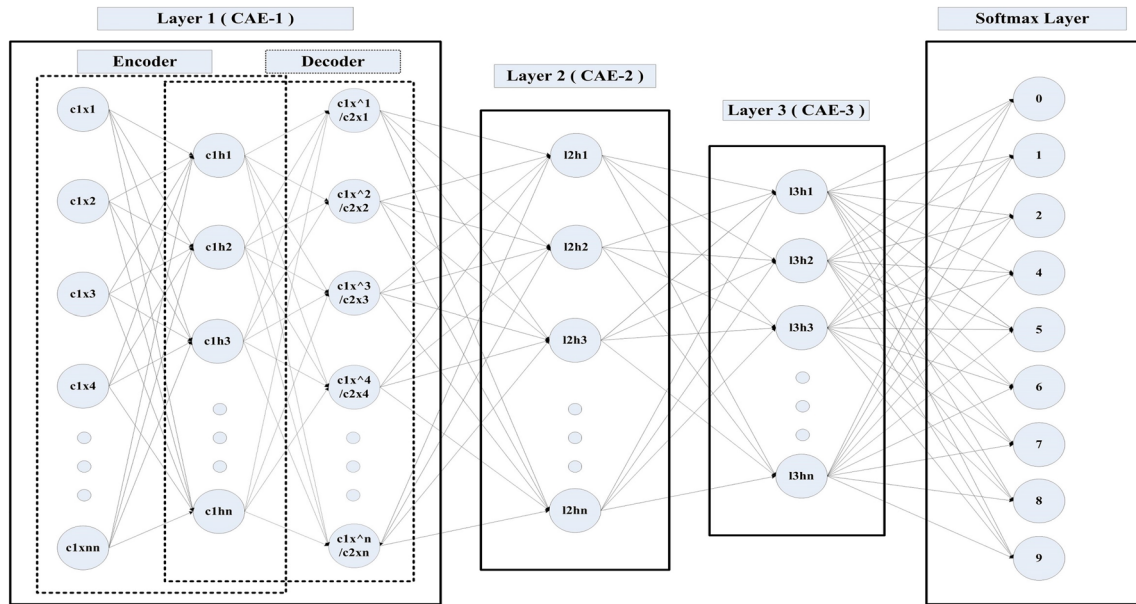


Fig. 1 Architecture of the proposed model



Fig. 2 MNIST small subset (basic)



Fig. 6 MNIST rotation and image background digits (bg-img-rot)



Fig. 3 MNIST random rotation digits (rot)

data. In first the phase of experiments, the data has been splitted in 70% training and 30% testing. While in the second phase of experiments, 50% of the data is selected for training and 50% for testing purpose. The output of experiments is mapped in the form of confusion matrices (CM) and receiver operating characteristic (ROC) curve graphs



Fig. 4 MNIST random noise background digits (bg-rand)



Fig. 5 MNIST random background digits (bg-img)

2 Proposed D-CAE

1: Parameters initialization

- No of hidden layers: h
- Input feature set: $[x_1, x_2, x_3, \dots, x_n]$
- Encoding activation function: EAF
- Decoding activation function: DAF
- Inputs weights: W_i
- Biasness values: b_i

2: while Repeat do

3: Encoding

- Compute encoded inputs $f(w)$ by Multiplying x_n and W_i
- Compute biased inputs $f(b)$ by adding b_i to encoded inputs
- Compute $f(x)$ using equation (1) by applying $f(w)$ and $f(b)$

4: Decoding

- Compute decoded outputs $f(w^o)$ by Multiplying x_n and W_i
- Compute biased output $f(b^o)$ by adding b_i to decoded outputs
- Compute $f(x^o)$ using equation (1) by applying $f(w^o)$ and $f(b^o)$

5: Optimization

- Optimize value of equation 6

6: All layers trained

7: end while

4.1 Training procedure

This section describes the training method and heuristics of the D-CAE model with details. The D-CAE consists of three layers namely input layer, hidden layer and the reconstruction layer. The input layer takes the whole input feature set. The hidden layer performs the internal processing of the auto encoder whereas the reconstruction layer maps the target output to the fed inputs. The features set obtained from the reconstruction layer are then given as inputs to the classifier (mainly softmax layer) for classification. The reconstruction of inputs from the original inputs works in unsupervised manner whereas the classification works in supervised fashion. We train our model using a feedforward mechanism. There are a few parameters that we adjust for every variant of MNIST dataset. In image processing, it is costly and memory consuming if the algorithm operates over the whole image pixels at once, due to that reason, it is more convenient to split image into patches. A single patch is rectangular or square piece of an image. For example, a 10×10 patch contains 100 pixels. The second technique applied in image processing is filtering. It refers to any enhancement or modification in image for the purpose of emphasizing some features or to remove others. These filters replace each pixel's intensity value by a weighted mean of the neighboring pixels. Third parameter to the proposed approach is aperture.

generally, the aperture size refers to the intensity resolution and sensitivity of image. With the decrement of aperture size, the resolution intensity improves and the sensitivity of image decreases which ultimately reduces the image processing cost. These parameters, the patch size, the number of layers and filters, were determined by using 10-fold cross validation. The values of parameters based on dataset variant are shown in Table 1. During our experiments, we kept the patch shape to square and for learning filters, we randomly select 1000 patches from each layer. Learning based on filters is a crucial step of preprocessing and can be expressed by Eq. 10.

Table 1 D-CAE Parameters for all MNIST variant-datasets

MNIST variants	Size of patch	Number of filters	Selected aperture
Basic	$8 \times 8, 7 \times 7, 6 \times 6$	8, 7, 7	1.4
Rot	$7 \times 7, 6 \times 6, 5 \times 5$	7, 6, 5	1.4
Bg-rand	$9 \times 9, 8 \times 8, 7 \times 7$	9, 9, 9	1.0
Bg-img	$5 \times 5, 4 \times 4, 3 \times 3$	10, 9, 8	0.7
Bg-img-rot	$10 \times 10, 9 \times 9, 8 \times 8$	9, 8, 8	0.8

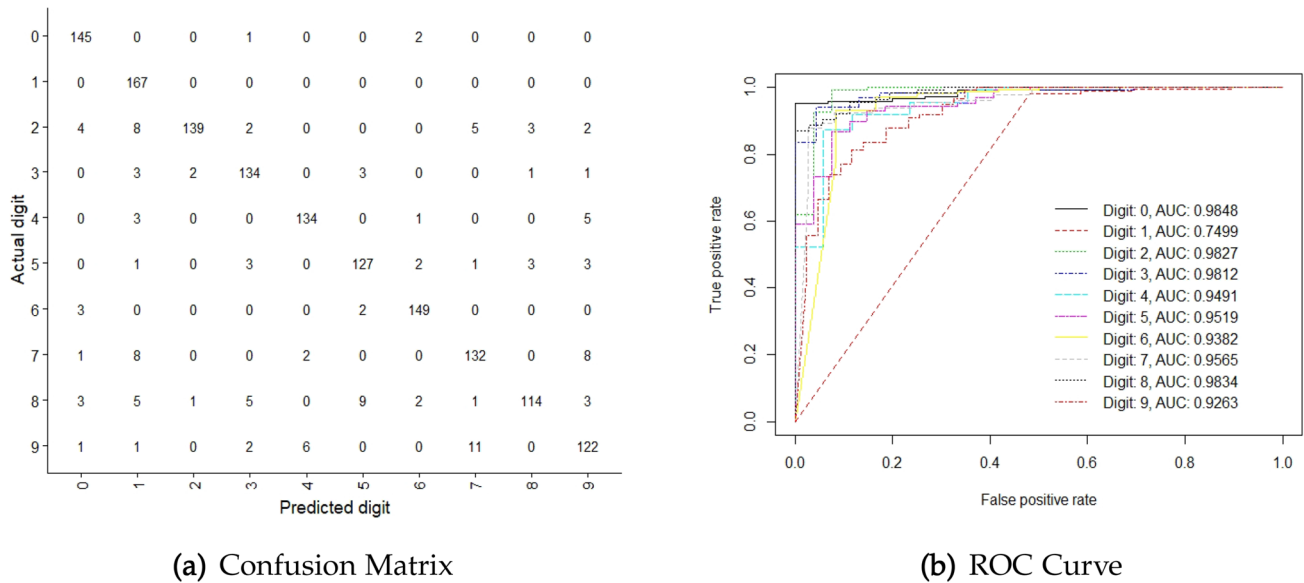


Fig. 7 CM & ROC for MNIST small subset (basic) with 70:30 training and testing data

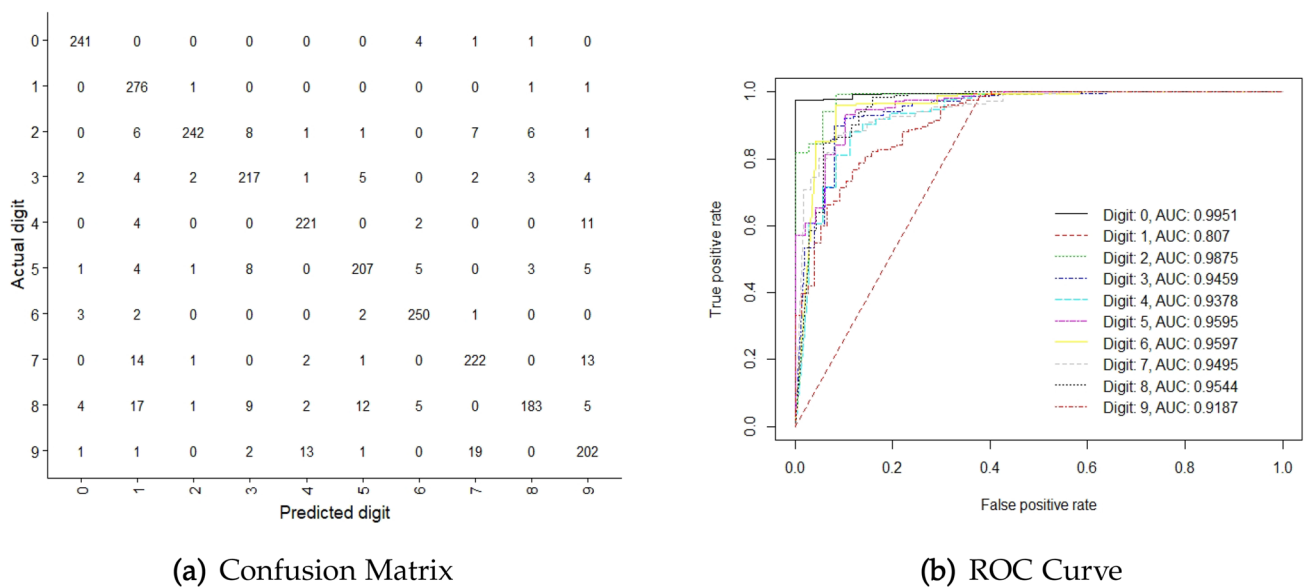


Fig. 8 CM & ROC for MNIST small subset (basic) with 50:50 training and testing data

$$\begin{cases} Z_i = U_i \sum_i VT_i \\ U_i = [u_1, u_2, u_3, \dots, u_p] \\ W_i = [u_1, u_1, u_1, \dots, u_k] \end{cases} \quad (10)$$

In Eq. 10, $Z_i \in R^{2 \times 1000}$ denotes collection of patch matrix having 1000 patches as a vector for i th layer where $i = 1, 2, 3$. $V_i \in R^{1000 \times 1000}$ and $U_i \in R^{p^2 \times p^2}$ denotes unitary matrices. $W_i \in R^{p^2 \times k}$ while $k < p^2$ denotes a matrix of filters collection where k is the number of filters. All the filtered features are penalized with variance 1 and mean 0. The aperture for each

of the extracted matrix is adjusted empirically at interval of [0,3] having a value that is determined by cross-validation. Parameters for each MNIST variant-dataset are shown in Table 1.

In these experiments, the core D-CAE model leverages a softmax classifier to find out the overall classification behaviour of D-CAE based image classification. Figures 7a, b and 8a, b shows the output confusion matrix and roc curve for 70:30 and 50:50 training-testing ratios respectively. In confusion matrices, it is clear that the proposed model shows better results for 70:30 as compared to 50:50

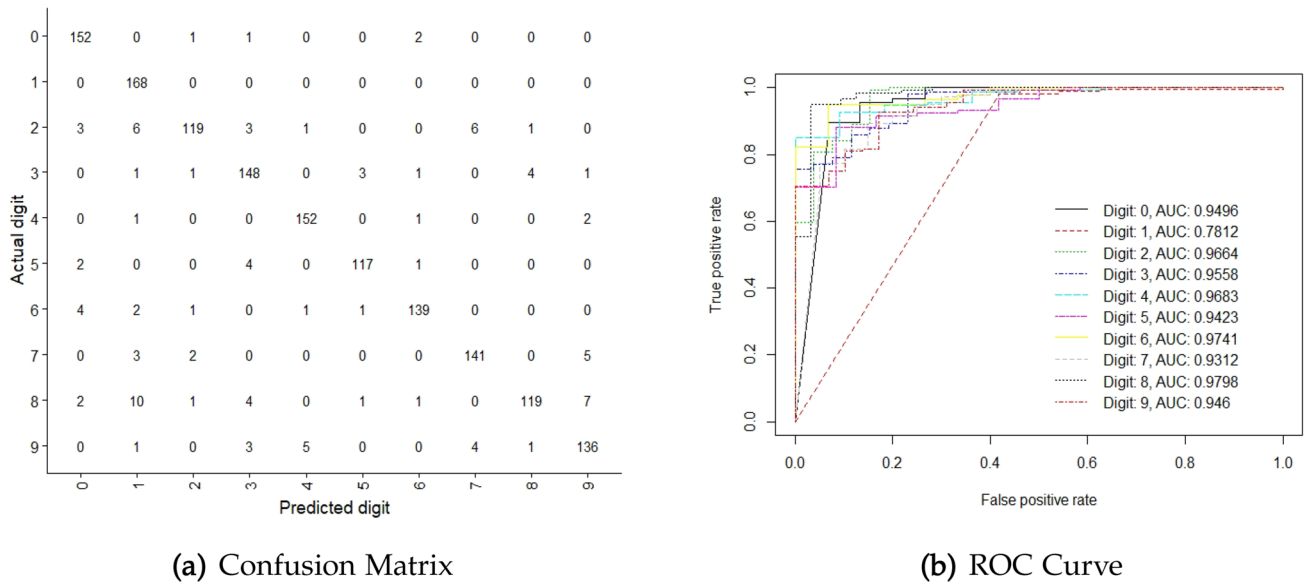


Fig. 9 CM & ROC for MNIST random rotation digits (rot) with 70:30 training and testing data

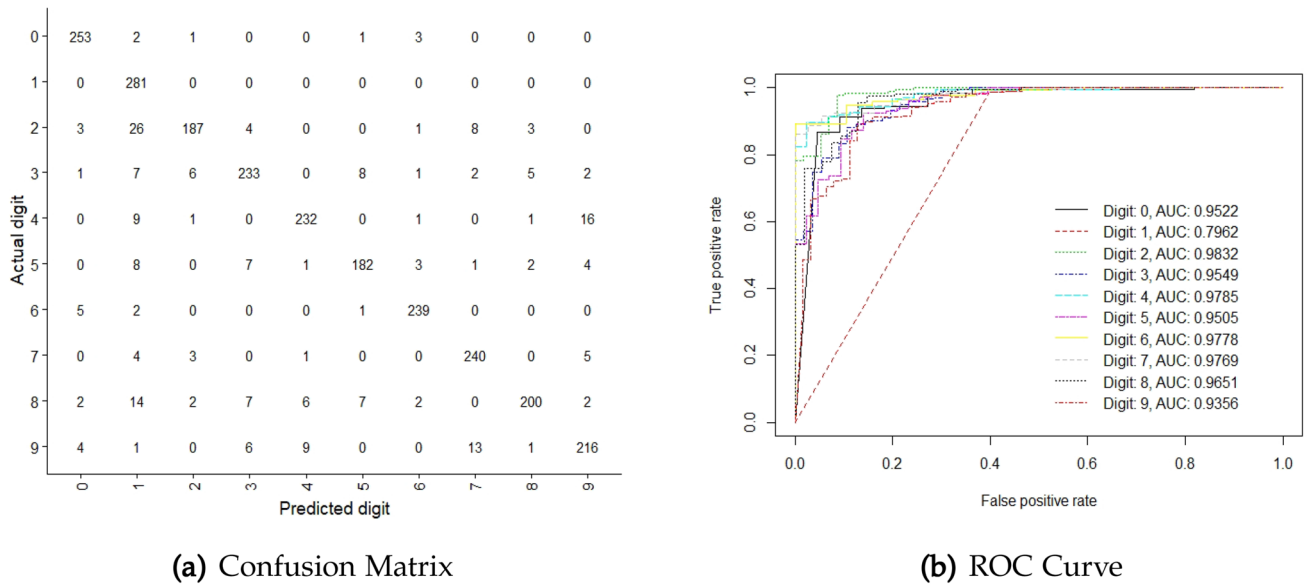


Fig. 10 CM & ROC for MNIST random rotation digits (rot) with 50:50 training and testing data

training-testing ratio on MNIST basic dataset. The highest accuracy is observed for class 1 followed by class 6 and class 0 because their features were more distinct as compared to other classes. On the other hand, the lowest accuracy can be seen for class 8 followed by class 9 and class 5, as these

classes were misclassified to one another because of high similarity in features.

From Figs. 9a and 10a we can observe also that our proposed model give better results for 70:30 training-testing ratio on MNIST random rotation digits (rot) dataset. the

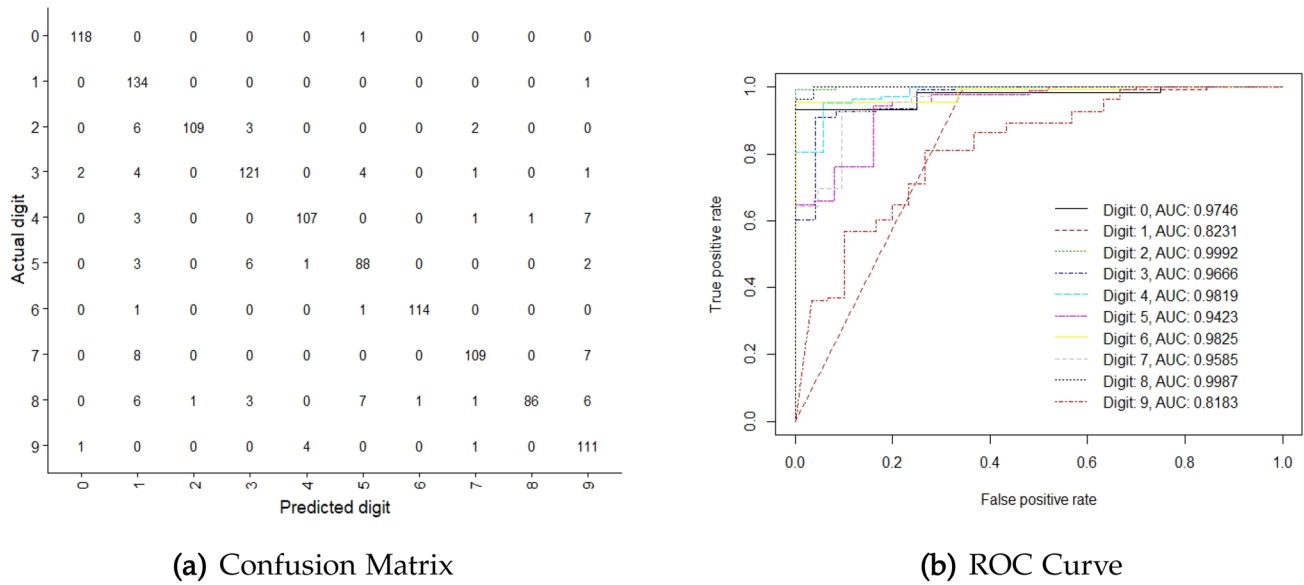


Fig. 11 CM & ROC for MNIST random noise background digits (bg-rand) with 70:30 training and testing data

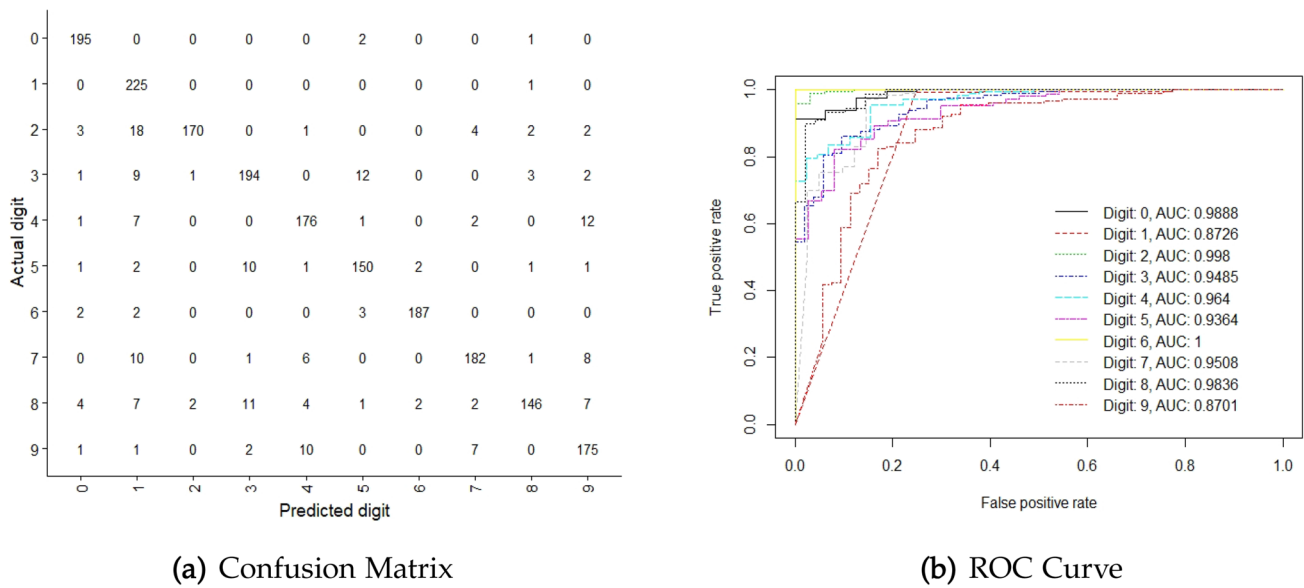


Fig. 12 CM & ROC for MNIST random noise background digits (bg-rand) with 50:50 training and testing data

highest accuracy is gained by class 1 and the lowest accuracy remains by the class 5. Also the ROC curve in Figs.9b and 10b resembles the same results for this MNIST variant dataset.

Figures 11a, b and 12a, b shows that the accuracy of D-CAE for random noise background digits (bg-rand) dataset is also approximately above than 90% expect Class 1 and class 9. Similarly Figs.13a, b and 14a, b presents the experimental results for MNIST random background digits (bg-img) on 70:30 and 50:50 training-testing data ratio.

the results are better for all classes except class 1, which is below 90%.

Figure 15a, b are also showing better result but there is some variation in accuracy. The second phase of experiments is based on splitting that dataset in 50% of training and 50% of testing data. Moreover, The results for MNIST rotation and image background digits (bg-img-rot) are shown in Figs. 7a, b and 16a, b. It is shown that the experimental results follow a minor decrement in the accuracy starting from MNIST small basic dataset to most complex MNIST rotation and image

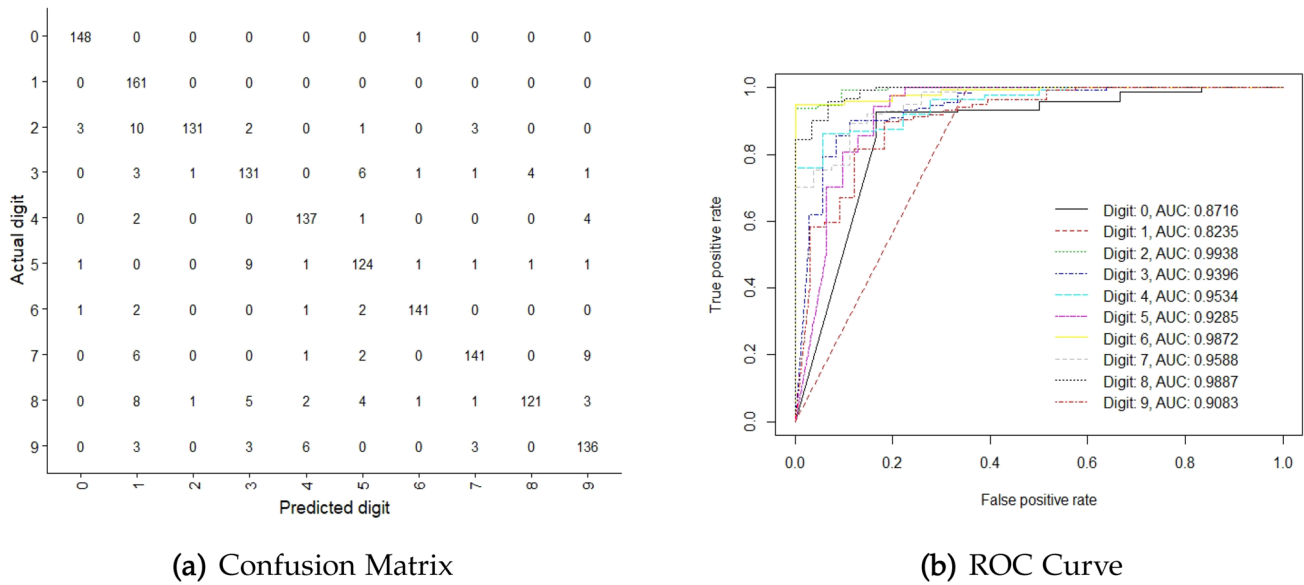


Fig. 13 CM & ROC for MNIST random background digits (bg-img) with 70:30 training and testing data

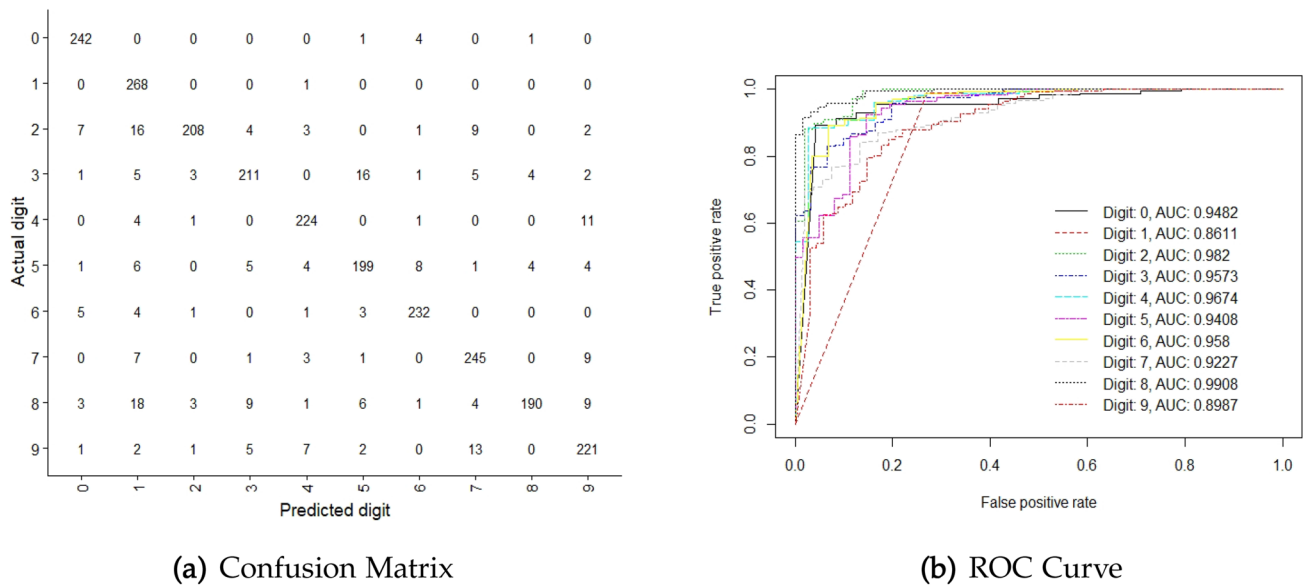


Fig. 14 CM & ROC for MNIST random background digits (bg-img) with 50:50 training and testing data

background digits dataset. Then, the gradual increment in the dataset complexity is related to a decreased accuracy.

5 Comparative analysis

In this section, it is evaluated that the capability of feature learning under the following conditions: (a) minimal time complexity, (b) better accuracy. To verify and validate our D-CAE model, we performed several experiments. All the experiments were carried out on Intel core i7 cpu with 8GB

of RAM having windows 10 operating system. The compiler and language used for developing and testing these algorithms is python3.6. For rapid development of the D-CAE model, we used keras [29]. Keras is python library for deeplearning based on a fast numeric computational base-library with high performance called Tensorflow [30]. Tensorflow allowed the easy implementation based on both CPU and GPU support. The benchmark variation datasets of MNIST are used for evaluation. MNIST is a handwritten digit images dataset contain 70,000 images from 0 to 9. Each image in this dataset has a size of 28 × 28 pixels. We

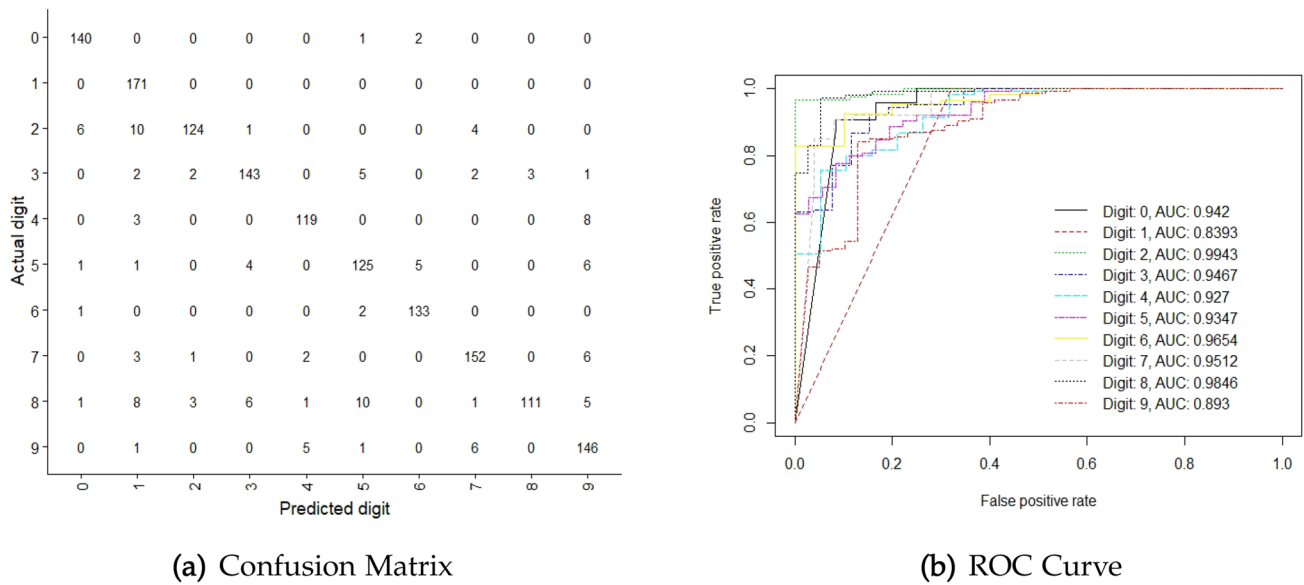


Fig. 15 CM & ROC for MNIST rotation and image background digits (bg-img-rot) with 70:30 training and testing data

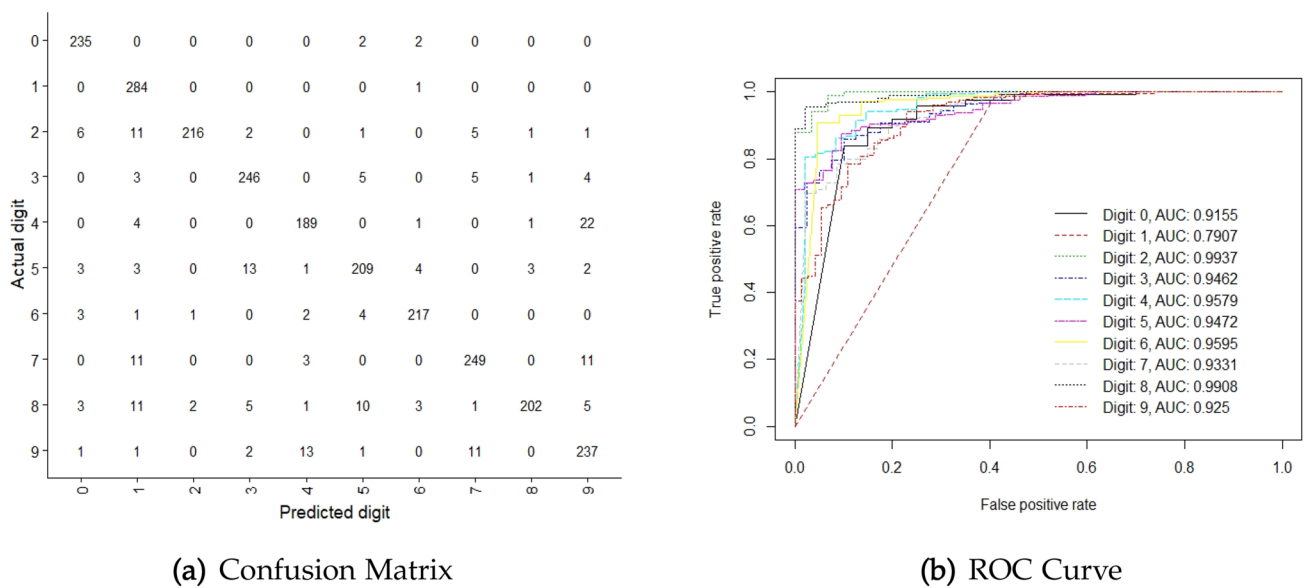


Fig. 16 CM & ROC for MNIST rotation and image background digits (bg-img-rot) with 50:50 training and testing data

Table 2 Running time comparison of different models on MNIST variant-datasets

MNIST variants	Sub-Basic		Sub-Bg-rand		Sub-Bg-img-rot	
	50/50	70/30	50/50	70/30	50/50	70/30
Training/testing	50/50	70/30	50/50	70/30	50/50	70/30
ScatNet	32m 32s	38m 51s	44m 25s	55m 44s	1h 6m 7s	1h 01m 31s
PCANet	24m 36s	28m 49s	38m 53s	43m 42s	47m 6s	56m 15s
RBM	1h 06m 45s	1h 32m 32s	1h 47m 5s	2h 15m 25s	1h 57m 20s	2h 42m 45s
DAE	57m 30s	1h 07m 09s	1h 32m 15s	2h 2m 02s	1h 48m 42s	2h 18m 05s
SAE	2h 0m 43s	2h 22m 57s	2h 6m 48s	2h 34m 35s	2h 18m 08s	2h 53m 32s
SCAE	1h 56m 43s	2h 13m 32s	2h 2m 26s	2h 22m 56s	2h 26m 54s	2h 48m 18s
D-CAE	31m 22s	36m 57s	34m 27s	39m 56s	41m 57s	47m 12s

Table 3 Training accuracy of different models on MNIST variant-datasets

MNIST variants	Sub-basic		Sub-rot		Sub-Bg-rand		Sub-Bg-img		Sub-Bg-img-rot	
Training/testing	50/50	70/30	50/50	70/30	50/50	70/30	50/50	70/30	50/50	70/30
PCANet	92.30	93.84	88.24	89.20	82.50	84.25	80.43	86.50	66.26	72.60
ScatNet	92.98	92.44	85.22	87.62	76.90	79.85	81.10	81.98	59.84	63.36
RBM	93.04	93.40	84.20	86.52	86.15	88.38	83.04	85.24	60.32	71.52
DAE	90.96	92.45	88.12	90.44	79.05	82.63	77.56	77.98	56.30	64.61
SAE	91.05	93.81	86.96	88.48	88.80	90.24	81.03	83.30	60.70	63.31
SCAE	91.11	94.90	87.05	87.25	88.21	88.98	74.25	77.34	65.85	68.28
D-CAE	92.24	93.73	87.23	89.34	90.89	92.67	83.96	87.12	72.60	76.09

split our experiments in two phases based on training and testing data ratio for each dataset. In first phase of experiments we split the data into 70% training and 30% testing dataset, that is 49,000 training and 21,000 testing. While in the second phase 35,000 images are considered as training set and 35,000 for testing. We also used 5000 images as validation for each of the MNIST variation dataset as discussed in Sect. 4.

5.1 Running time comparison

Besides the different computational behaviour of different algorithms, All the experiment were carried out on same hardware and software architecture as mentioned in Sect. 5. Still some variance is obvious concluded from results presented in Table 2. This summarizes the runtime for training of different competitive models from literature based on MNIST variant-datasets. Table 2 shows the clear observation of the time taken by iterative based methods training i.e. RBM [31], DAE [32], SAE [33] and SCAE [17], is few times greater than that of non-iterative based methods i.e. PCANet [34, 35] and ScatNet [36]. The different parameter settings affects the runtime complexity of the aforementioned models. Nevertheless, the proposed D-CAE outperform non-iterative methods and iterative based models with the same parameter tuning.

5.2 Digital recognition on MNIST variation-datasets

The results in Table 3 show the evaluation of our proposed D-CAE model. We did not use the backpropagation in our training, which makes our training speed faster than the other models based on non-iterative mechanisms. Moreover, the training time of our model is mostly used in image representation learning but we used low level feature representation that reduces this time complexity as well as we can further reduce it by increasing the capacity of memory. Because memory consumption is directly proportional to representation learning, so by extending the memories will enhance the representation learning speed. In addition we can apply some principal component analysis function in a parallel behaviour with after memory extension to boost the training phase. This property of our model is also a key to move towards big data applications. In the comparative analysis of our model, some state-of-art methods are used. Some of them are non-iterative in nature e.g. PCANet [37, 38] and ScatNet [39] while some are iterative e.g. DAE [32], RBM [31], SAE [13] and SCAE [17]. Table 3 provide a conclusion of results that the proposed D-CAE model is moving towards better performance as the dataset is getting more complex and increase in size, as it can be seen in the last and second last datasets, our ranking is high where the MNIST variant-datasets are more complicated. The significance of our model is not clearly proved by the basic MNIST, because the latter is standardized. On the other hand, the rest of

Table 4 Parameter setting

MNIST variants	Sub-basic		Sub-rot		Sub-Bg-rand		Sub-Bg-img		Sub-Bg-img-rot	
Trainingtesting	50/50	70/30	50/50	70/30	50/50	70/30	50/50	70/30	50/50	70/30
PCANet+Softmax	89.43	90.67	77.90	79.25	77.55	79.14	71.45	74.84	59.95	63.38
ScatNet+Softmax	87.34	89.21	69.94	75.80	68.21	74.45	69.32	71.58	54.10	61.76
RBM+Softmax	88.86	90.12	73.64	76.54	78.58	80.32	73.20	76.18	63.10	68.36
DAE+Softmax	85.30	89.80	71.46	76.5	71.02	74.10	69.90	72.56	61.86	63.08
SAE+Softmax	85.51	89.35	70.64	70.07	75.36	79.11	71.58	77.80	69.52	71.90
D-CAE+Softmax	88.25	90.23	81.96	83.92	86.98	85.21	80.20	81.23	71.53	74.96

Table 5 Parameter setting

MNIST variants	Sub-basic		Sub-rot		Sub-Bg-rand		Sub-Bg-img		Sub-Bg-img-rot	
	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
ScatNet+Softmax	0.86	0.83	0.78	0.77	0.74	0.74	0.72	0.72	0.64	0.62
PCANet+Softmax	0.89	0.85	0.82	0.82	0.77	0.76	0.75	0.75	0.66	0.63
RBM+Softmax	0.89	0.87	0.79	0.77	0.78	0.76	0.77	0.76	0.68	0.66
DAE+Softmax	0.88	0.86	0.76	0.72	0.73	0.73	0.71	0.69	0.68	0.67
SAE+Softmax	0.86	0.85	0.77	0.77	0.77	0.75	0.77	0.74	0.67	0.66
SCAE+Softmax	0.87	0.87	0.84	0.81	0.76	0.75	0.74	0.74	0.69	0.69
D-CAE+Softmax	0.87	0.86	0.85	0.85	0.82	0.81	0.80	0.80	0.73	0.71

MNIST variant-datasets are more complex due to rotations, background randomizations and the presence of background images. Hence, the experimental results in Table 3 explains the significance of our model in complex data environment.

5.3 Classification based comparison

In addition to the analysis of our proposed model based on learning error rate, some experiment are conducted for the analysis of testing error rate by using different models and softmax classifier in the last layer. We train and test all of the experimented models using the same architecture for pre-classification steps. Table 4 list all the models with their classification error rates. All of the results classification models discussed in Table 4 is the mean value for 5 times repeated experiment. The final results conclude that D-CAE with Softmax classifier outperforms ScatNet, PCANet, RBM, DAE and SAE with softmax classifier.

In order to keep the inner-class variance of extracted features from being 0, The ScatNet model's results are generated regardless the threshold value. In some cases the variance of few class feature become 0, we applied normal fit using normal distribution for each of that class. Table 4 shows some interesting results for PCANet, the most usable model having non-iterative behaviour [40] did not show any effective results here. However, it outperforms well-known RBM based classifier usually does not perform well if the data size is increasing in training [41]. The rest of the aforementioned models performed almost similar to their variants performance based on the same accuracy rate. By considering the multi-dimensionality extracted features for some of the non-iterative methods is decisive. But classifier based on models like SAE, performs well in high-dimensional feature space.

Besides the other classifiers in our experiments, Softmax classifier performed well. The Softmax regression classifier is based on an iterative algorithm, for this reason it is provided an iterative solution implementing the Limited-memory Broyden-FletcherGoldfarb-Shanno (L-BFGS) [42] algorithm. We adjust the iteration count to 500 and γ is 10-4,

which is weight decay term coefficient. These parameters settings are provided by many researchers in literature [43, 44]. Due to feature dimensionality reduction property with simple architecture and easy way of implementation, the proposed model with Softmax classifier outperformed the state-of-art models as listed in Table 4.

In addition with the training and testing accuracies in Tables 3 and 4, and 5 provide the more explanatory performance evaluation metric called precision and recall. Precision refers to the ratio of positive observations that are correctly classified to the overall positive classified instances. Recall is the ratio of correctly classified positive instances to the all actual class instances. Generally both precision and recall are meant for the purpose of binary classification therefore, we calculate these measures for each class separately and then take the average of all classes in order to get the final precision and recall values. Table 5 shows the precision and recall measures comparison of the proposed model with other state-of-art models.

6 Conclusion

This paper proposed a fast and simple architecture based D-CAE for feature reduction and abstract representation learning in image classification. The research work presented in this manuscript is concluded with three main contributions: it used the CAE for learning high-level features without using backpropagation scheme. Softmax classifier is used as the last layer of trained D-CAE for classification and lastly, we performed comparative experiments in order to prove the significance of our model on MNIST variant-datasets as it is described in Sect. 5. The experiments show that our model has produced effective results compared to state-of-art classification algorithm on relatively complex datasets. The simplicity of proposed D-CAE architecture provides a valid base for the implementation and experimentation. Even if, as it is shown in Sect. 4, when the data is getting more complex; D-CAE outperforms the comparative models in feature learning, classification as well as in time

complexity. As a conclusion, D-CAE performed better on complex MNIST variants datasets. Therefore, we will focus our future researches on the evaluation of more complex datasets, such as: Toronto Face Detection (TFD) and Canadian Institute For Advanced Research (CIFAR).

Acknowledgements The authors would like to thank Universiti Tun Hussein Onn Malaysia (UTHM) and Ministry of Higher Education (MOHE) Malaysia for supporting this Research.

References

- Aamir M, Mohd Nawi N, Mahdin HB, Naseem R, Zulqarnain M (2020) Auto-encoder variants for solving handwritten digits classification problem. *Int J Fuzzy Logic Intell Syst* 20(1):8–16
- Hosseini-Asl E, Zurada JM, Nasraoui O (2016) Deep learning of part-based representation of data using sparse autoencoders with nonnegativity constraints. *IEEE Trans Neural Netw Learn Syst* 27(12):2486–2498
- Bengio Y, Yao L, Alain G, Vincent P (2013) Generalized denoising auto-encoders as generative models. In: *Advances in neural information processing systems* pp 899–907
- Jia K, Sun L, Gao S, Song Z, Shi BE (2015) Laplacian auto-encoders: an explicit learning of nonlinear data manifold. *Neurocomputing* 160:250–260
- Zhang N, Ding S, Shi Z (2016) Denoising Laplacian multi-layer extreme learning machine. *Neurocomputing* 171:1066–1074
- Wang W, Cui Z, Chang H, Shan S, Chen X (2014) Deeply coupled auto-encoder networks for cross-view classification. *arXiv preprint arXiv:14022031*
- Liu W, Ma T, Tao D, You J (2016) HSAE: a Hessian regularized sparse auto-encoders. *Neurocomputing* 187:59–65
- Chorowski J, Zurada JM (2015) Learning understandable neural networks with nonnegative weight constraints. *IEEE Trans Neural Netw Learn Syst* 26(1):62–69
- Hong C, Yu J, Wan J, Tao D, Wang M (2015) Multimodal deep autoencoder for human pose recovery. *IEEE Trans Image Process* 24(12):5659–5670
- Nishino K, Inaba M (2016) Bayesian AutoEncoder: generation of Bayesian networks with hidden nodes for features. In: *AAAI* pp 4244–4245
- Rifai S, Mesnil G, Vincent P, Muller X, Bengio Y, Dauphin Y, et al. (2011) Higher order contractive auto-encoder. In: *Joint European conference on machine learning and knowledge discovery in databases*. Springer. pp 645–660
- Vincent P, Larochelle H, Lajoie I, Bengio Y, Manzagol PA (2010) Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *J Mach Learn Res* 11:3371–3408
- Qi Y, Wang Y, Zheng X, Wu Z (2014) Robust feature learning by stacked autoencoder with maximum correntropy criterion. In: *2014 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE. pp 6716–6720
- Lemme A, Reinhart RF, Steil JJ (2010) Efficient online learning of a non-negative sparse autoencoder. In: *ESANN*. Citeseer
- Wu P, Hoi SC, Xia H, Zhao P, Wang D, Miao C (2013) Online multimodal deep similarity learning with application to image retrieval. In: *Proceedings of the 21st ACM international conference on multimedia*. ACM, pp 153–162
- Mescheder L, Nowozin S, Geiger A (2017) Adversarial variational bayes: Unifying variational autoencoders and generative adversarial networks. *arXiv preprint arXiv:1701.04722*
- Liu Y, Feng X, Zhou Z (2016) Multimodal video classification with stacked contractive autoencoders. *Sig Process* 120:761–766
- Rifai S, Vincent P, Muller X, Glorot X, Bengio Y (2011) Contractive auto-encoders: explicit invariance during feature extraction. In: *Proceedings of the 28th international conference on international conference on machine learning*. Omnipress, pp 833–840
- Nawi NM, Ransing RS, Salleh MNM, Ghazali R, Hamid NA (2010) An improved back propagation neural network algorithm on classification problems. In: *Database theory and application, bio-science and bio-technology*. Springer, pp 177–188
- Wahid F, Ismail LH, Ghazali R, Aamir M (2019) An efficient artificial intelligence hybrid approach for energy management in intelligent buildings. *KSII Trans Internet Inf Syst* 13(12):5904–5927
- Mohd Nawi N, Hamzah F, Hamid NA, Rehman MZ, Aamir M, Azhar AR (2017) An optimized back propagation learning algorithm with adaptive learning rate. *Int J Adv Sci Eng Inf Technol* 7(5):1693–1700
- Snášel V, Nowaková J, Xhafa F, Barolli L (2017) Geometrical and topological approaches to Big Data. *Future Gener Comput Syst* 67:286–296
- Wahid F, Ghazali R (2018) Hybrid of firefly algorithm and pattern search for solving optimization problems. *Evol Intel* 1–10
- Izenman AJ (2013) Linear discriminant analysis. In: *Modern multivariate statistical techniques*. Springer. pp 237–280
- Wahid F, Ghazali R, Fayaz M, Shah AS (2017) Statistical features based approach (SFBA) for hourly energy consumption prediction using neural network. *Networks* 8:9
- Aamir M, Mohd Nawi N, Wahid F, Mahdin H (2019) An efficient normalized restricted Boltzmann machine for solving multiclassclassification problems. *Int J Adv Comput Sci Appl* 10(8):416–426
- Wahid F, Ghazali R, Shah H (2018) An improved hybrid firefly algorithm for solving optimization problems. In: *International conference on soft computing and data mining*. Springer, pp 14–23
- Larochelle H, Erhan D, Courville A, Bergstra J, Bengio Y (2007) An empirical evaluation of deep architectures on problems with many factors of variation. In: *Proceedings of the 24th international conference on Machine learning*. ACM, pp 473–480
- Chollet F, et al (2018) Keras: the python deep learning library. *Astrophysics Source Code Library*
- Abadi M, Barham P, Chen J, Chen Z, Davis A, Dean J, et al (2016) Tensorflow: a system for large-scale machine learning. In: *OSDI*. vol 16, pp 265–283
- Montavon G, Müller KR, Cuturi M (2016) Wasserstein training of restricted Boltzmann machines. In: *Advances in neural information processing systems*, pp 3718–3726
- Liang J, Liu R (2015) Stacked denoising autoencoder and dropout together to prevent overfitting in deep neural network. In: *2015 8th International congress on image and signal processing (CISP)*. IEEE, pp 697–701
- Sankaran A, Vatsa M, Singh R, Majumdar A (2017) Group sparse autoencoder. *Image Vis Comput* 60:64–74
- Chan TH, Jia K, Gao S, Lu J, Zeng Z, Ma Y (2015) PCANet: A simple deep learning baseline for image classification? *IEEE Trans Image Process* 24(12):5017–5032
- Zi Y, Xie F, Jiang Z (2018) A cloud detection method for Landsat 8 images based on PCANet. *Remote Sens* 10(6):877
- Zhu W, Miao J, Qing L, Huang GB (2015) Hierarchical extreme learning machine for unsupervised representation learning. In: *2015 International joint conference on neural networks (IJCNN)*. IEEE, pp 1–8
- Liu R, Lu T (2016) Character recognition based on PCANet. In: *2016 15th International symposium on parallel and distributed computing (ISPDC)*. IEEE, pp 364–367

38. Lee JN, Byeon YH, Pan SB, Kwak KC (2018) An EigenECG network approach based on PCANet for personal identification from ECG Signal. *Sensors*. <https://doi.org/10.3390/s18114024>
39. Wang Z, Chang S, Ling Q, Huang S, Hu X, Shi H, et al (2016) Stacked approximated regression machine: a simple deep learning approach. arXiv preprint [arXiv:1608.04062](https://arxiv.org/abs/1608.04062).
40. Soon FC, Khaw HY, Chuah JH, Kanesan J (2018) PCANet-Based Convolutional Neural Network Architecture For a Vehicle Model Recognition System. *IEEE Trans Intell Transp Syst* 99:1–11
41. Liao L, Jin W, Pavel R (2016) Enhanced restricted Boltzmann machine with prognosability regularization for prognostics and health assessment. *IEEE Trans Industr Electron* 63(11):7076–7083
42. Biglari F, Ebadian A (2015) Limited memory BFGS method based on a high-order tensor model. *Comput Optim Appl* 60(2):413–422
43. Jae-Neung Lee YHB, Kwak KC (2018) An EigenECG network approach based on PCANet for personal identification from ECG signal. *Micromachines* 9(4):411
44. Lee JN, Byeon YH, Kwak KC (2018) Design of ensemble stacked auto-encoder for classification of horse gaits with MEMS inertial sensor technology. *Micromachines* 9(8):411

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.