

How to cite this paper:

Roy, S. S., Ahmed, M., & Akhand, M. A. H. (2018). Noisy image classification using hybrid deep learning methods. *Journal of Information and Communication Technology*, 17 (2), 233–269.

NOISY IMAGE CLASSIFICATION USING HYBRID DEEP LEARNING METHODS

**¹Sudipta Singha Roy, ²Mahtab Ahmed &
²Muhammad Aminul Haque Akhand**

*¹ Institute of Information and Communication Technology
Khulna University of Engineering & Technology, Khulna, Bangladesh*

*² Dept. of Computer Science and Engineering
Khulna University of Engineering & Technology, Khulna, Bangladesh*

*sudipta.singha.roy@iict.kuet.ac.bd; mahtab@cse.kuet.ac.bd;
akhand@cse.kuet.ac.bd*

ABSTRACT

In real-world scenario, image classification models degrade in performance as the images are corrupted with noise, while these models are trained with preprocessed data. Although deep neural networks (DNNs) are found efficient for image classification due to their deep layer-wise design to emulate latent features from data, they suffer from the same noise issue. Noise in image is common phenomena in real life scenarios and a number of studies have been conducted in the previous couple of decades with the intention to overcome the effect of noise in the image data. The aim of this study was to investigate the DNN-based better noisy image classification system. At first, the autoencoder (AE)-based denoising techniques were considered to reconstruct native image from the input noisy image. Then, convolutional neural network (CNN) is employed to classify the reconstructed image; as CNN was a prominent DNN method with the ability to preserve better representation of the internal structure of the image data. In the denoising step, a variety of existing AEs, named denoising autoencoder (DAE), convolutional denoising autoencoder (CDAE) and denoising variational autoencoder

(DVAE) as well as two hybrid AEs (DAE-CDAE and DVAE-CDAE) were used. Therefore, this study considered five hybrid models for noisy image classification termed as: DAE-CNN, CDAE-CNN, DVAE-CNN, DAE-CDAE-CNN and DVAE-CDAE-CNN. The proposed hybrid classifiers were validated by experimenting over two benchmark datasets (i.e. MNIST and CIFAR-10) after corrupting them with noises of various proportions. These methods outperformed some of the existing eminent methods attaining satisfactory recognition accuracy even when the images were corrupted with 50% noise though these models were trained with 20% noise in the image. Among the proposed methods, DVAE-CDAE-CNN was found to be better than the others while classifying massive noisy images, and DVAE-CNN was the most appropriate for regular noise. The main significance of this work is the employment of the hybrid model with the complementary strengths of AEs and CNN in noisy image classification. AEs in the hybrid models enhanced the proficiency of CNN to classify highly noisy data even though trained with low level noise.

Keywords: Image denoising, CNN, denoising autoencoder, convolutional denoising autoencoder, variational denoising autoencoder, hybrid architecture.

INTRODUCTION

In recent years, deep learning approaches have been extensively studied for image classification and image processing tasks such as perceiving the underlying knowledge from images. Deep neural networks (DNN) utilize their deep layer-wise design to emulate latent features from data and thus pick up the possibility to appropriately classify patterns. Arigbabu et al. (2017) combined Laplacian filters over images with the Pyramid Histogram of Gradient (PHOG) shape descriptor (Bosch, et al., 2007) to extract face shape description. Later, they used the Support Vector Machine (SVM) (Cortes & Vapnik, 1995) for face recognition tasks. One progressive feature of extracting variants of DNNs, the convolutional neural network (CNN) (LeCun et al., 1998; Krizhevsky et al., 2012; Schmidhuber, 2015), has surpassed the vast majority of the image classification methods. Different research work outcomes boldly indicate that feature selection from deep learning with CNN should be the primary candidate in most of the image recognition tasks (Sharif et al. 2014). The convolution and the following pooling (Scherer et al., 2010)

layers preserve the possession of the corresponding location of features and along these lines make the CNN empowered to preserve a better epitome of the input data. Current CNN works are concentrated on computer vision issues, for example 3D objects recognition, traffic signs and natural images classification (Huang and LeCun, 2006; Cireşan et al., 2011a; Cireşan et al., 2011b), image segmentation (Turaga et al., 2010), face detection (Matsugu et al., 2003), chest pathology identification (Bar et al., 2015), Magnetic Resonance Image (MRI) segmentation (Bezdek et al., 1993) and so on. However, the performance of deep CNN highly depends on the tremendous amount of pre-processed labeled data. Simonyan (2013) proposed an improved variant of the Fisher vector image encoding method and combined it with a CNN to develop a hybrid architecture that can classify images requiring a comparatively smaller computational cost than the traditional models, as well as assess the performance of the image classification pipeline with increased depth in layers.

Some variants of deep models, named unsupervised deep networks, learn underlying representation from input images overcoming the necessity of these input data to be labeled. One traditional model of this type is stacked autoencoders (SAE) (Bourlard and Kamp, 1988; Bengio, 2009; Rumelhart, 1985) in which the basic architecture holds a stack of shallow encoders which enable them to learn features from the data by means of encoding the input data into a vector and then decoding this vector to its native representation. Shin et al. (2013) pertained the stacked sparse autoencoders (SSAEs) for medical image classification task and achieved notable promotion in classification accuracy. Norouzi et al. (2009) introduced the stacked convolutional restricted Boltzmann machine (SCRBM) which incorporates dimensional locality and also weight sharing by maintaining the stack of the convolutional restricted Boltzmann machine (CRBM) to build deep models. Lee et al. (2009) introduced convolutional deep belief network (CDBN), which places the CRBM in each layer instead of RBM unlike the deep belief network (DBN), and utilization convolution structure to join the layers and thus build hierarchical models. Contrasted with the conventional DBN, it preserves spatial locality and enhances the performance of feature representation (Hinton et al., 2006). With comparable thoughts, Zeiler et al. (2010, 2011) proposed a deconvolutional deep model in view of the conventional sparse coding technique (Olshausen and Field, 1997). The deconvolution operation depends on the convolutional deterioration of information under a sparsity imperative. It is a modification of the traditional sparse coding methods. Contrasted with sparse coding, it can learn better feature representation.

Data subjected to noise is a hinder once to the success of the deep network-based image recognition systems in real world applications. Nonetheless, in most of the cases in real life scenarios, during transmission and acquisition, digital images are adulterated with noise resulting in degenerating the performance of image classification, medical image diagnosis, etc. One major issue originating from one of the intrinsic attributes of a DNN is its affectability to the input data. Because of being sensitive to little perturbation, DNNs may be misled and misclassify an image having a certain amount of imperceptible perturbation (Szegedy et al., 2013). As a result, when there is noise present in the input data, learned features by the DNN may not be vigorous. As examples, medical imaging techniques which are vulnerable to noise such as: MRI, X-rays, Computer Tomography (CT) can be considered (Sanches et al., 2008). Reasons fluctuate from the utilization of various image acquisition systems to endeavors at diminishing patients' introduction to radiation. As the measure of radiation is diminished, there is adulteration of the images with noise increments (Gondara, 2016; Agostinelli et al., 2013). A survey conducted by Lu and Weng (2007) investigated the image classification methods and suggested that image denoising prior to classification is efficient in case of remotely sensed data in a thematic map such as the geographical information system (GIS). Even if, the classifier is trained with noisy data, it does not show a much better performance in case of image classification. So, image denoising has become a compulsory requirement prior to feeding the image to the classifier in order to achieve a better classification result.

A notable number of researches have been directed over image denoising in the time period of the previous couple of years to make the deep learning-based image classification systems more compatible with practical applications. In the past, research in this field has conducted where denoising was accomplished on the premise of the wavelet transformation technique (Coifman and Donoho, 1995), the partial differential equation-based methods (Perona and Malik, 1990; Rudin and Osher, 1994; Subakan et al., 2007), and in addition conveyed scant coding approaches (Elad and Aharon, 2006; Olshausen and Field, 1997; Mairal et al., 2009). Singh et al. (2014) proposed an efficient classification model for multi-class object images subject to Gaussian noise. They applied wavelet transform-based image denoising techniques by means of employing the NeighShrink thresholding over the wavelet coefficients to eliminate wavelet coefficients causing noise in the image and picking up only useful ones.

Recent studies have effectively utilized deep learning-based approaches with the intention to accomplish image denoising (Krizhevsky et al., 2012; Bengio et al., 2007; Glorot et al., 2011). Burger et al. (2012) demonstrated that similar execution to the previously described strategies can be accomplished by applying plain multi-layer perception (MLP). Jain et al. (2009) employed CNN to denoise images which performed superior to wavelets notwithstanding utilizing a smaller set of training images. An assortment of autoencoders (AEs) has been employed to denoise images and these techniques have definitely surpassed the conventional denoising methods as they are less restrictive for details of noise generative mechanisms (Cho, 2013; Vincent et al., 2008; Vincent et al., 2010). Vincent et al. (2008) introduced the denoising autoencoder (DAE) which figures out how to recreate local images from adulterated forms by injecting arbitrary noise into the images of the training set amid the learning period. These DAEs are stacked to develop a deep unsupervised learning network called stacked DAE (SDAE) for adapting profound depiction (Vincent et al., 2010). Xie et al. (2012) deployed a combination of sparse coding along with DAE for tasks of image denoising and blind inpainting. It was designed to work with images subject to white Gaussian noise and superimposed text. Cho (2013) employed Boltzmann machines as well SDAEs for image denoising tasks in case of high level of noise injected in the images. He employed three distinct depth settings (one, two and four layers) for both the SDAEs and the Boltzmann machines to evaluate the performance of noise omission. Agostinelli et al. (2013) introduced the adaptive multi-column DNN with a combination of multiple-stacked sparse DAEs (SSDAE) that can denoise various types of noises in the images in a standalone manner. They computed optimal column weights using a nonlinear optimization program and later trained the individual networks to anticipate the optimal weights. One common disadvantage of these DAE-based models is that they learn the underlying hierarchical features from the image by reshaping the high dimensional data to vectors and thus discard the intrinsic structures of the images.

With the intention to solve this problem, Masci et al. (2011) proposed another variant of the autoencoder called convolutional AE (CAE) which trains itself for reconstructing images from the input image data in a convolutional manner. The stacked CAE forces the adjacent CAEs to learn the innate structure of the input image throughout the series of convolution and pooling operations. The kernels and other learning parameters of each layer are updated by backpropagation to convolve the feature maps of the input images into more abstract features of each layer. Compared to previously specified AEs it has

proved its capability to preserve more relating structural information. Xu et al. (2014) developed a deep CNN that can figure out the characteristics of blur degradation from an image. Gondara (2016) employed DAEs constructed with convolutional layers for denoising medical images. Du et al (2017) proposed stacked convolutional denoising autoencoders (SCDAE) by stacking DAEs in a convolutional way where each layer produces high dimensional feature maps by means of convolving features of the previous layer trained by a DAE. Recently, Kingma and Welling (2014) introduced the variational autoencoder (VAE), a hybrid of deep learning model along with variational inference that has prompted remarkable advances in generative modelling. The loss function used for training VAE is calculated by a variational upper bound on the log-likelihood of the data. It can figure out and preserve shape variability beyond the image set as well as reconstruct images given the manifold coordinates. Unlike other deterministic models, it is a probabilistic generative model which is trained all through with stochastic gradient descent. Unlike DAE that corrupts the input images by adding noise at the input level and later learns to reconstruct the clear image, VAE learns with noise added in its stochastic hidden layer. Im et al. (2017) proposed that adding noise in not only the stochastic hidden layer but also in the input layer is beneficial and empowers the VAE to perform image denoising tasks. They proposed a modified training criterion for denoising variational autoencoders (DVAE) that resemble a tractable bound, in case the input image is adulterated with noise.

The intention of this work was to build a few supervised image classifiers that can demonstrate better classification results across a noisy image set; thereby, contemplating DAE, CDAE, DVAE and proposing some hybrid models utilizing CNN along with these AEs. Initially a DAE, a CDAE and a DVAE were trained with image data subject to lower regular noise level so that they could omit noise from the input images and reconstruct a native form of it. To counter the massive noisy images, two hybrid structures (i.e. DAE-CDAE and DVAE-CDAE) were further investigated where for each of them two AEs were deployed in a cascaded manner. The reconstructed images from these AEs were fed to a following CNN for classification, where the CNN is trained with raw images having zero percent noise injected into it. The classification performance of this CNN is solely dependent on the quality of the reconstructed images from the conventional as well the hybrid AE structures. The DAE-CDAE-CNN as well as DVAE-CDAE-CNN models can work better with massive noisy images because of their cascaded architectures and thus omits the requirement of training with images corrupted by noise of different levels.

HYBRID DEEP LEARNING-BASED NOISY IMAGE CLASSIFICATION

Real world image classification tasks suffer from noise and other imperfections existing in the image data. So, denoising images prior to classification is compulsory. Noisy image classification tasks incorporate two steps, i.e. image denoising and image classification. This section first explains some conventional models for image denoising based on AEs as well as image classification with CNN. Then it presents the proposed hybrid methods consisting different cascaded AEs plus CNN.

CONVENTIONAL METHODS FOR IMAGE DENOISING AND CLASSIFICATION

Convolutional Neural Network (CNN) as Image Classifier

CNNs (LeCun et al., 1998) which are multiple-layered variants of artificial neural network (ANN) are well applied to classify images and perceive visual patterns straightforwardly from pixel images. In a CNN architecture, the information propagation throughout its multiple layers allows it to extract features from the perceived data at layers apiece by means of applying digital filtering techniques. CNNs perform on the basis of two main processes: convolution and subsampling. During the convolution process, a small-sized kernel is applied over input feature map (IFM) and produces a convolved feature map (CFM). The first set of CFMs are produced by applying the convolutional operation over the original input image. Here, a kernel is only an arrangement of weights and a bias. Every particular point in the CFM is gained by applying the same kernel over every small portion of the IFM, called a local receptive field (LRF). In this way, weights are shared among all positions throughout the convolutional process and spatial locality is preserved. The CFM computed from an IFM would be,

$$CFM_{(x,y)} = \mathfrak{f} \left(\sum_{i=1}^{K_h} \sum_{j=1}^{K_w} K_{(i,j)} * IFM_{(x+i,y+j)} + \mathfrak{B} \right) \quad (1)$$

where \mathfrak{B} and \mathfrak{f} represent the bias of the kernel $K_h \times K_w$ activation function respectively, whereas the 2-D convolution is symbolized by $*$. Throughout all the experiments here, the scaled sigmoid activation function as well as a single bias is used for every latent map used. While particular kernels may create distinct CFMs from the same IFM operations of numerous kernels are formed to deliver CFMs for different IFMs.

In CNN, each convolutional layer is followed by a subsampling layer to simplify the feature map gained from the convolution operation. This simplification process is done by selecting significant features from a region and discarding the rest (Du et al., 2017). Among various sub-sampling methods, max-pooling (Scherer et al., 2010) was used throughout our experiments. It takes the maximum incentive over non-overlapping sub-locales and can be defined as:

$$SFM(x, y) = d \left(\sum_{i=0}^{R-1} \sum_{j=0}^{C-1} CFM_{(xR-1+i, yC-1+j)} \right) \quad (2)$$

where R and C denote size of the pooling area as $R \times C$ matrix and d denotes the subsampling operation on the pooling area. The size of SFM becomes half of the CFM if $R \times C$ is 2×2 . In max-pooling, each point in the SFM is the maximum value computed from a particular 2×2 locale of the CFM (Akhand et al., 2016, 2017).

In CNN, the series of convolution-subsampling operation is followed by a hidden layer and then an output layer sequentially. Where nodes of a hidden layer and output layers are fully connected there lies a linear representation of terminal SFM values as a hidden layer. The error in the classification task can be measured from:

$$\mathbb{E}(d_o, y_o) = \frac{1}{2n} \sum_{i=1}^n (d_o(\mathcal{P}) - y_o(\mathcal{P}))^2, \quad (3)$$

where n is the product of the total number of patterns and the total number of output nodes in that particular classification task, every particular pattern \mathcal{P} , d_o and y_o denotes the desired output and obtained output respectively. The learning parameters are updated during backpropagation. Throughout our experiment, back-propagation (BP) (Liu et al., 2015; Bouvrie, 2006) was used for training the CNN. The CNN applied here in our experiment is demonstrated in Fig. 1. It consists of two convolutional layers (conv1 and conv2) and two subsampling layers (sub1 and sub2) each following a single convolutional layer. Throughout the experiments, the CNN used here was trained with noiseless raw images.

Denoising Autoencoder (DAE)

The DAE expands the conventional autoencoder alongside some stochastic augmentations keeping in mind the end goal to attain the ability to reproduce the native image from its noisy form (Vincent et al., 2008). This noise is usually included by physically utilizing deterministic distribution. The architecture of the DAE is demonstrated in Fig. 2.

For a given input $x \in [0,1]^{\text{dimension}}$, DAE adulterates x into $\tilde{x} \in [0,1]^{\text{dimension}}$ with some random noise. It is added with a certain probability \wp using a stochastic mapping.

$$\tilde{x} = \mathcal{D}(\tilde{x}|x, \wp) \quad (3)$$

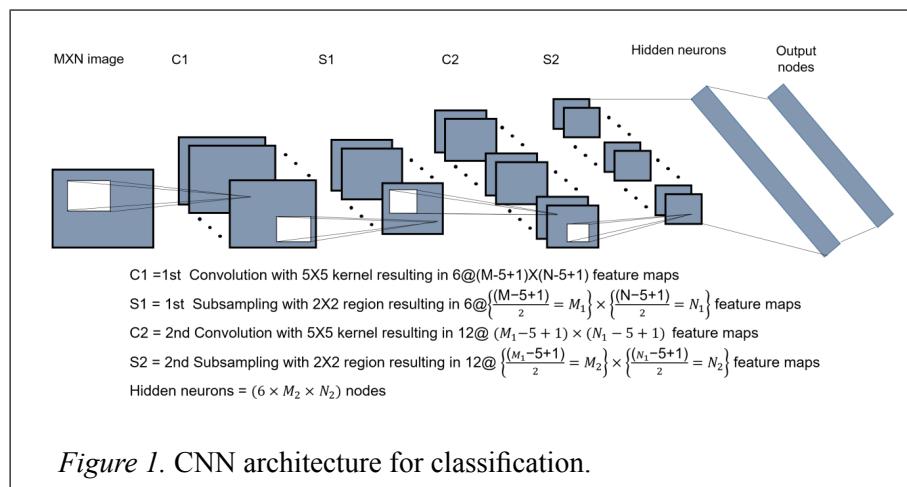
The type of distribution \mathcal{D} is regulated by the distribution of the original input x and the kind of arbitrary noise added to it. In practical cases, binomial noise is used for black and white images, whereas for color images uncorrelated Gaussian noise is better suited. However, the zero masking (binomial) noise as well as Gaussian noise were applied throughout the experiments here. Then, \tilde{x} was mapped to a underlying hidden representation y by means of a nonlinear deterministic function \mathcal{F} ,

$$y = \mathcal{F}(\mathcal{M}_1 \tilde{x} + \mathcal{B}_1) \quad (4)$$

In the very same way as in the traditional autoencoder, this hidden representation then mapped to the reconstructed feature, $z \in [0,1]^{\text{dimension}}$ of by original input applying another nonlinear deterministic function φ .

$$z = \varphi(\mathcal{M}_2 y + \mathcal{B}_2) \quad (5)$$

The construction error was assessed by computing the mean squared error Δ between input x and the reconstructed feature representation z . This is defined as:



$$\Delta(x, z) = \frac{1}{2n} \sum_{i=1}^n (x_i - z_i)^2 \quad (6)$$

The main aim of this reconstruction process is to minimize the construction error and this is done by optimizing the model parameters in such a way that:

$$optimal(\mathcal{M}_1, \mathcal{M}_2, \mathcal{B}_1, \mathcal{B}_2) = \arg \min_{\mathcal{M}_1, \mathcal{M}_2, \mathcal{B}_1, \mathcal{B}_2} \Delta(x, z) \quad (7)$$

For our experiment, the DAE was trained with images corrupted by 20% noise.

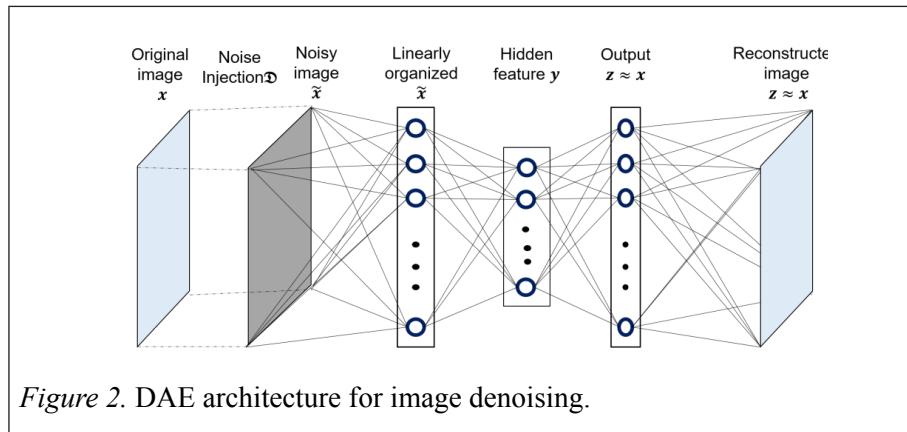
Convolutional Denoising Autoencoder (CDAE)

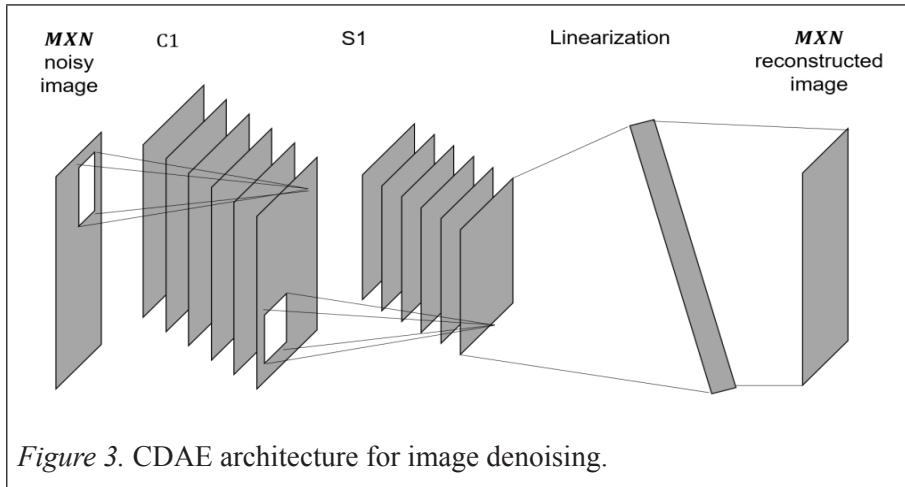
The fundamental contrast between CDAE (Masci et al., 2011) and conventional autoencoders is unlike others. CDAE shares weights among all positions in the input and consequently it conserves spatial locality. Subsequently, the consequent reconstruction process is finished by a linear combination of all-important IMAGE PATCHES on the premise of the latent code. For a single channel input x the latent representation of the k^{th} feature map would be:

$$h^k = \varrho(x * \omega^k + \mathcal{B}^k) \quad (8)$$

where \mathcal{B} denotes the bias, ϱ represents the activation function and the 2-D convolution is symbolized by $*$. The scaled hyperbolic tangent activation function and a single bias were used for every latent map during the experiments. The reconstruction was achieved by applying:

$$y = \varrho \left(\sum_{n \in H} h^k * \tilde{\omega}^k + \beta \right) \quad (9)$$





As in the previous step, for every input channel, one bias β was also used here also. H denotes the group of underlying feature maps, the flip operation over one and the other dimensions of the weights are identified by $\tilde{\omega}$. The error function used here is defined as:

$$\varepsilon(x, y) = \frac{1}{2n} \sum_{i=1}^n (x_i - y_i)^2 \quad (10)$$

The gradient of this error function is computed during the backpropagation (Liu et al., 2015; Bouvrie, 2006) step. The overall architectural description is illustrated in Fig. 3. The convolution operation employed here were uniform to the convolution operation depicted in the CNN section. Amid the training period the native image was utilized as the output label with a specific end goal to update the kernel weights and different parameters so that in times of testing the CDAE could reproduce a noise-omitted picture given a noise-injected one. In this experiment, the CDAE was trained with 20% noisy images.

Denoising Variational Autoencoder

The denoising variational autoencoder (DVAE) (Ciresan et al., 2011c; Kingma and Welling, 2013), a modern variant of AE, is a deep directed graphical model that interprets the output of the encoder by means of variational inference. There are basically three components as the building block of a DVAE: an encoder, the following decoder and finally a loss function. The structure of the DVAE used all through this experiment is demonstrated in Fig. 4. Both the encoder and the decoder can be any variant of the neural network. It computes probability distribution $p_\theta(x, y)$ and thus finds out the probability distribution of data x by employing the following equation:

$$p_{\theta}(x) = \int p_{\theta}(x, y) dy = \int p_{\theta}(x|y) p(y) dy \quad (11)$$

where θ denotes the weights and biases of the decoder, $p(y)$ is the probability distribution of the latent variable y which is often the standard normal distribution $\mathcal{N}(0, I)$, and $p_{\theta}(x|y)$ is the decoder's output under noise rumination in terms of probability distribution of the reconstructed data given latent features.

The encoder neural network takes data point x as input and translates it to a hidden representation y which has significantly less dimension than x . As the encoder learns to compress the data into a significantly stochastic less dimensional space, it produces output parameters which is a Gaussian probability density $q_{\phi}(y|x)$. ϕ represents the weights and biases of the encoder. This posterior $q_{\phi}(y|x)$ is the uncorrelated multivariate normal determined by the encoder:

$$p_{\theta}(x) = \int p_{\theta}(x, y) dy = \int p_{\theta}(x|y) p(y) dy \quad (12)$$

where \mathcal{N} represents the standard normal, μ_{ϕ} and σ_{ϕ} denote the mean and the standard deviation respectively. The decoder neural network takes the latent feature representation y as input and its outputs are the parameters to the probability distribution of the data $p_{\theta}(x|y)$. As the decoder tries to reconstruct from the real-valued numbers in y with less dimensionality to real-valued numbers in x of higher dimensionality, some information may be lost. This reconstruction loss is calculated using log-likelihood .

$$\log p_{\theta}(x|y)$$

Unlike other conventional autoencoders, the loss function used in DVAE is the negative log-likelihood with an additional regularizer. As all the data points do not share global representation, the loss function is decomposed into just terms that rely on a single data point. The loss function ℓ_i for a single data point x_i is computed by:

$$\ell_i(\phi, \theta) = -E_{y \sim q_{\phi}(y|x_i)} [\log p_{\theta}(x_i|y)] + KL(q_{\phi}(y|x_i) || p_{\theta}(y)) \quad (13)$$

Thus, for total data points the overall loss would be:

$$L = \sum_{i=1}^N \ell_i(\phi, \theta) \quad (14)$$

This DVAE is trained to reconstruct native images from their 20% noisy form.

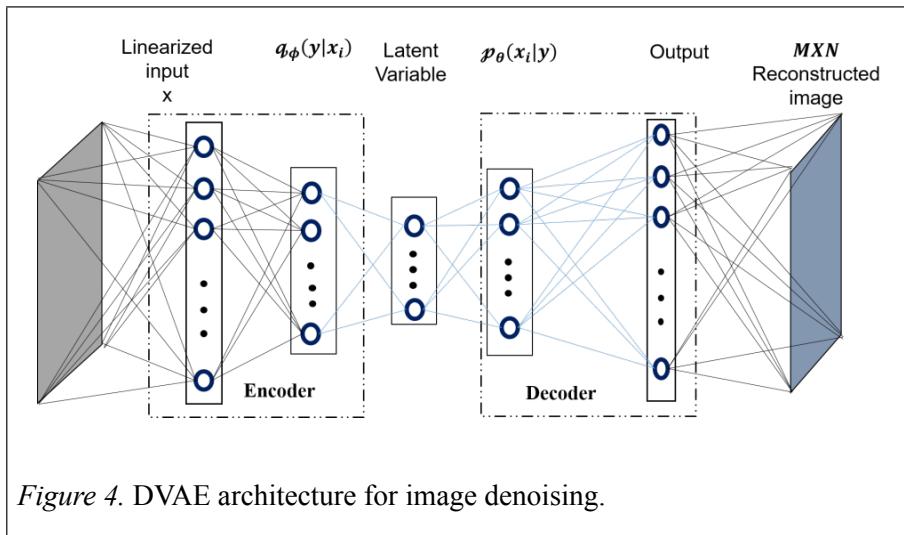


Figure 4. DVAE architecture for image denoising.

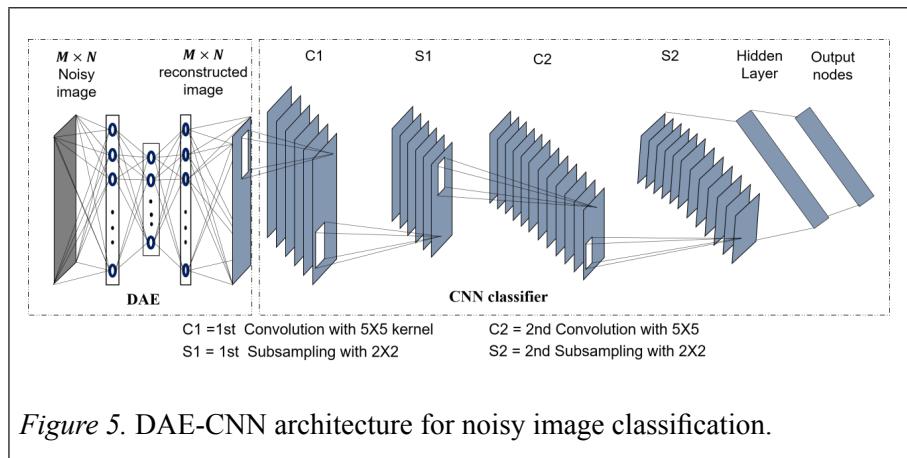
Proposed Hybrid Models for Noisy Image Classification

This section explains the proposed hybrid models DAE-CNN, CDAE-CNN, DVAE-CNN, DAE-CDAE-CNN and DVAE-CDAE-CNN for noisy image classification. The common feature of all these models is that a CNN is used as a classifier which takes denoised image (i.e. reconstructed) from the prior AE of a particular model. Conventional AE(s) of a model are trained individually with regular noise and CNN is trained with noise-free image. Finally, AE(s) and CNN are cascaded to form a particular hybrid model and no further training is performed. The following subsections explain the architectural description as well as the working procedures of each individual model.

Hybrid Model 1: DAE-CNN Architecture

The proposed DAE-CNN is a supervised deep network designed in order to perform image classification regardless of the possibility of they being noisy. With layer-wise training, the whole architecture of the DAE-CNN is optimized. Fig. 5 shows the all-inclusive architecture of the proposed DAE-CNN model. This model is a fusion of DAE and a two-layered CNN. In the first place, the noisy image is refined by the DAE, and afterward the reconstructed image is fed to the accompanying CNN. DAE filters the noises from the input images via the reconstruction process. All the encoder and decoder parameters (the input-hidden and the hidden-output weights) are initialized by the weights of the DAE trained before (discussed in the DAE section). The following CNN is designed with two convolution-subsampling layers; at first, a following dense layer and finally an output layer. All the parameters of the CNN (the

hidden-output weights, local averaging parameters, and kernels) are set to the corresponding parameters used in the pre-trained CNN as discussed in the CNN section. In the end, only via a forward pass, this architecture does the noisy image classification task.

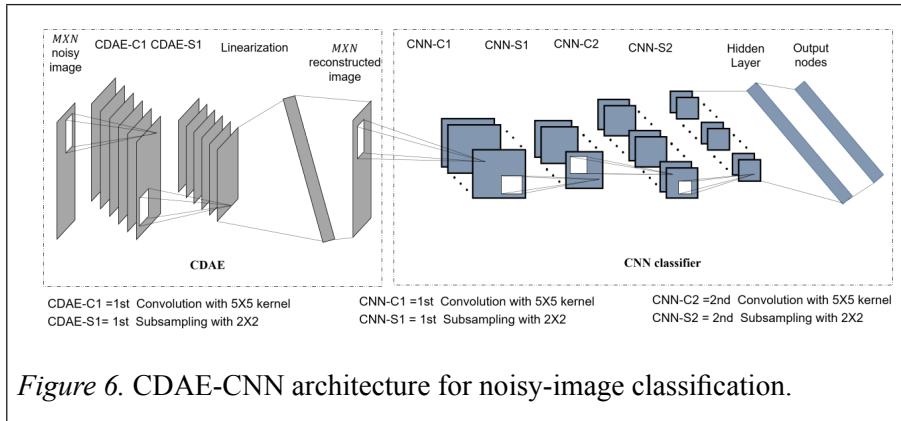


Hybrid Model 2: CDAE-CNN Architecture CDAE-CNN is another supervised deep network used in this study for classifying noisy images as shown in Fig. 6. It is a combination of a CDAE at first and follows a two-layered CNN in the very same manner. DAE-CNN architecture incorporates a DAE as an image reconstructor and a CNN as a classifier. Serving as a filter as well as a reconstructor, the CDAE reconstructs noise-free images from the noisy version fed to it and then passes it to the following CNN. The kernel weights along all the parameters of both CDAE and the CNN used here were initialized with the value of the corresponding parameters of the pre-trained CDAE and CNN (discussed in the CDAE and CNN section).

Hybrid Model 3: DVAE-CNN Architecture

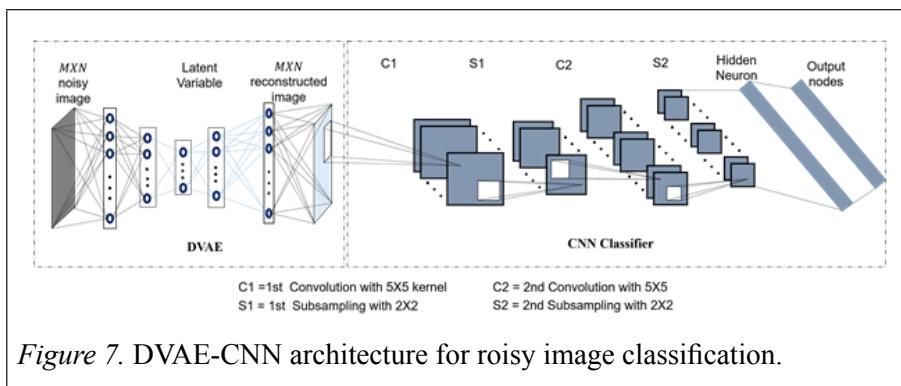
The DVAE-CNN architecture incorporates one image reconstructor and a following classifier like DAE-CNN and CDAE-CNN architecture. In this model DVAE serves as the noise filter as well as the image reconstructor. At first the noisy image is fed to the DVAE. Like DAE and CDAE it also reconstructs noise-free native images from the noisy input images but in a variational inference manner. Moreover, it uses an additional regularizer along with the negative log-likelihood which is common in all other traditional autoencoders. The following two-layered CNN takes this reconstructed and less noisy image as input and classifies it. The inclusive architecture is optimized via layer-wise training. Fig. 7 gives a proper demonstration of this

model. All the weights between the input and hidden layers as well as between the hidden and output layers of the DVAE are initialized with the corresponding weights of the pretrained DVAE (discussed in the DVAE section). After this, the classifier CNN is initialized containing two convolution-subsampling layers, a dense layer and in the end, an output layer. All the parameters of this CNN are initialized with the ones of the very same parameters used in the pre-trained CNN (discussed in the CNN section). A simple forward pass would then employ DVAE-CNN in the classification task.



Hybrid Model 4: DAE-CDAE-CNN Architecture

The hybrid DAE-CDAE-CNN-supervised image classifier incorporates both the denoising and convolutional approaches (DAE and CDAE) for filtering noisy images and reconstructing noise-free raw



images from them. The all-embracing structure of the DAE-CDAE-CNN is exhibited in Fig. 8. It has three basic components: first a DAE, then a CDAE,

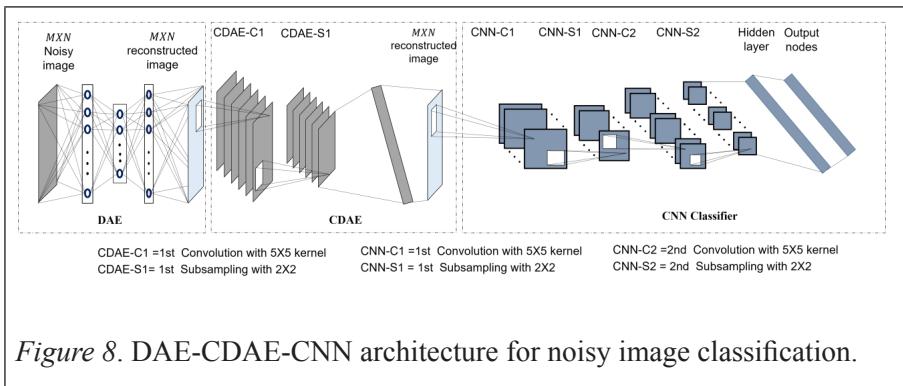


Figure 8. DAE-CDAE-CNN architecture for noisy image classification.

both serving as image reconstructors, and finally a two-layered CNN serving as an image classifier. The fundamental point of this method is to enhance the accuracy of the image classification with better reconstructions of the noisy images by having a good quality. The first image reconstructor DAE's input-hidden weights as well as the hidden-output weights are set to the value of the same pre-trained DAE's corresponding weights as in DAE-CNN architecture. DAE tries to reconstruct the raw image emitting the noise from the noisy input image serving as a filter and outputs a reconstructed image with less noise. This reconstructed intermediate image is then fed to the CDAE for further denoising. Compared to DAE, CDAE yields a better reconstruction in case of images. As this CDAE is fed with less noisy images than the original input, it outputs a better intermediate representation of the image for the following classifier. The kernels and other performance parameters of this CDAE are regulated uniformly to the pre-trained CDAE discussed in the CDAE section. The two-layered CNN is also regulated uniformly to the CNN, trained with zero noise added images for classification purpose (discussed in the CNN section).

Hybrid Model 5: DVAE-CDAE-CNN Architecture

DVAE-CDAE-CNN (shown in Fig. 9) works in the same manner as the DAE-CDAE-CNN architecture (discussed in the section on Hybrid Model 4: DAE-CDAE-CNN Architecture) and contains two image reconstructors: at initial point, a DVAE, and then a CDAE. As DVAE performs better image reconstruction than the traditional DAE (Im et al., 2017) the input image for CDAE is better in quality here compared to the DAE-CDAE-CNN architecture. As a result, the hybrid image reconstructor DVAE-CDAE outputs better images for the following CNN classifier compared to the DAE-CDAE-CNN architecture resulting in a better image classification in case the image is noisy. All the parameters in this DVAE are tuned to the corresponding

parameters' value of the pre-trained DVAE (as specified in the DVAE section). The kernels, hidden-output weights along with the local averaging parameters used in this structure are initialized with corresponding parameter values of the CDAE and CNN previously trained (discussed in the CDAE and the CNN section).

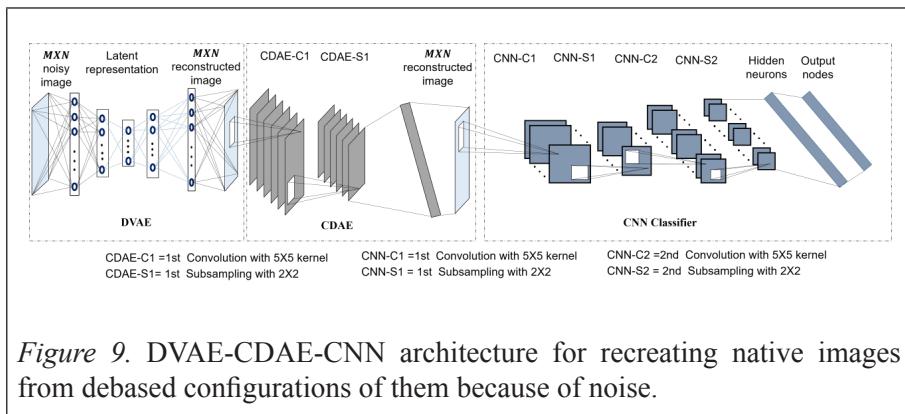


Figure 9. DVAE-CDAE-CNN architecture for recreating native images from debased configurations of them because of noise.

PERFORMANCE EVALUATION

This section investigates the performances of the proposed hybrid models on the benchmark datasets of two different categories: MNIST numeral images and CIFAR-10 object images. This section first describes the datasets and the experimental setups used to work over these datasets. Experiments were conducted at different noise levels and the proficiency of the models were compared against existing models. These models were implemented in Matlab R2015a. The performance analysis was conducted on MacBook Pro Laptop (CPU: Intel Core i5 @ 2.70 GHz and RAM: 8.00 GB) in OS-X Yosemite environment.

Data Description

Image data corrupted with noise occurs while dealing with real life practical applications. Even when a well-established system is employed on real-life data that system might fail only because of the inappropriateness of the data. Therefore, it is highly required to preprocess those image data prior to applying them in the practical application plot. With the intention to cope with this type of scenario, and at the same time to show the significance of the proposed models we considered two benchmark datasets: MNIST (LeCun et al. 2010) and CIFAR-10 (Coates et al. 2011), in this study. A large number

of recent studies utilized these two datasets considering the image data as a source (LeCun et al., 1998; Vincent et al., 2008; Vincent et al., 2010; Masci et al., 2011).

MNIST Dataset: The dataset contains 70000 28x28-sized sample images with a large variety of distinct numeral images from various individuals rehearsing distinctive individual writing patterns. The images are divided into training and test sets. The test set holds 10000 images having 1000 samples for each of the 10 numerals and the training set contains 60000 images having 6000 images for every individual digit. Fig. 10(a) displays few sample images of every handwritten numeral

CIFAR-10 Dataset: This dataset contains 32x32-sized 60000 samples of colored images of ten different

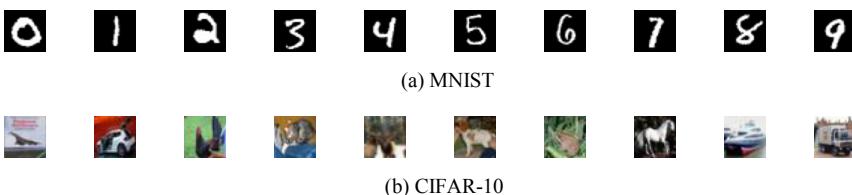


Figure 10. Samples of some images from MNIST and CIFAR-10 datasets.

objects (airplane, automobile, bird, cat, deer, dog, frog, horse, ship and truck). The training set consists of 50000 distinct images having 6000 samples of every class. The remaining 10000 sample images are used as test set. In the test set, each of every object has exactly 1000 sample images which are mutually exclusive. A few images from each class are shown in Fig. 10(b). For the experiments, these images were converted to grey-scale.

Experimental Setup

We experimented the proposed hybrid models over MNIST and grey-scaled CIFAR-10 datasets. Images in these two datasets were different in size. MNIST contained images of size 28x28 whereas images in CIFAR-10 were of size 32x32 forcing us to apply different architecture for the proposed models. This section describes the actual architectural setup used to work with MNIST and CIFAR-10 datasets.

A uniform experimental environment was set up for fair investigation among the proposed and the existing methods. As the images from the dataset were of size 28×28 (MNIST) and 32×32 (CIFAR-10), each of these classifiers

had 784 (and 1024) input units so as to take the linearized version of the data. As the data was divided into 10 classes, each of these classifiers had 10 units in the output layer. The intermediate portion of each of the network varied based on its architecture. DAE and DVAE had hidden layer size of 500 (and 700). Additionally, DVAE had an additional latent representation layer of size two. On the other hand, CDAE had kernel size of 5×5 and a subsample window of 2×2 local averaging area. Throughout the experiments, a two-layered CNN was used with all of the AEs (conventional and hybrid) having two convolution-subsampling layers. For both convolutional layers, the kernel size remained fixed and was 5×5 , in both subsampling layers; the size of the pooling area was 2×2 .

Due to the large-sized training set, batch-wise training was performed; and all of the experiments were conducted with a fixed batch size of 50. Weights of each of these networks were updated once for a batch of image patterns and batch size (BS), i.e. the number of patterns in a batch, was considered as a user-defined parameter in such a way that the total training patterns were completely divisible by the BS value. For the experiments, the learning rate (i.e. eta) values were varied in the range of 0.1 to 1.0.

EXPERIMENTAL RESULTS AND ANALYSIS

As these models were validated against two datasets, the experimental results and analysis are presented in two different subsections.

Result on MNIST Dataset

This section illustrates the performance of the proposed models over the MNIST data set with noise of different proportions injected in it. Fig. 11 delineates the result of the noise removal step utilizing DAE, CDAE, DAE-CDAE, DVAE-CNN, DVAE-CDAE-CNN separately on 50% noisy image data and these reconstructed images were fed to the following CNN classifier. Initially, the images in the dataset were pre-processed and without any additional noise added in it. With a specific end goal to assess the performance of these proposed hybrid classifiers on noisy images, noise was added manually to the images in the dataset. Zero masking noise was used for conducting the experiments in which an arbitrary matrix with the equal size of training image data was initialized where some of the pixels being arbitrarily OFF having the probability of 20% for both training and test cases and then 50% only for the test case. It can be clearly seen that the reconstructed image from DAE-CDAE and DVAE-CDAE were much better than the reconstructed ones from the

standalone AEs. Because of DVAE using the variational upper bound on the log-likelihood of the data as loss function rather than normal reconstruction error as DAE, it produces better representation of the reconstructed images than DAE. However, DVAE produced a little blurry image but it kept the shapes of the objects more accurate than DAE. So, when a CDAE was used in a cascaded manner after DVAE this blurriness also got omitted resulting in the DVAE-CDAE architecture to output better reconstructed images than DAE-CDAE in terms of 50% noisy input images.

	Original Image					
	With 50% noise					
Reconstructed Image	DAE					
	CDAE					
	DVAE					
	DAE - CDAE					
	DVAE-CDAE					

Figure 11. Sample of original images from MNIST dataset with and without noise and their reconstruction using different AEs.

The test set classification performance of all five models proposed in this study along with a simple CNN for both scenarios when the images were corrupted with 20% noise as well as 50% noise are portrayed in Fig. 12. The classification accuracy notes up to 400 interactions and Fig. 12(a) gives evidence that the DVAE-CNN architecture surpasses all other architecture in terms of 20% noisy image classification with 98.84 % accuracy. CDAE-CNN confirms the second position with 98.69% accuracy. The accuracy recorded for DAE-CNN, DAE-CDAE-CNN, DVAE-CDAE-CNN and simple CNN architectures are 98.01%, 97.40%, 97.43% and 97.76% respectively. In Fig. 12(b) it is clearly visible that whenever the same test set images are corrupted with 50% noise, the DVAE-CDAE-CNN architecture surpasses all other models attaining 96.74% accuracy whereas it shows least accurate classification result in the previous case. The reason behind this contradictory scene is that if we compare 50% noisy images against the 20% noisy image

data set, a larger number of pixels are found to be forcefully turned ON/OFF for 50% noisy images. That's why, whenever the frontier DVAE trained to work with 20% noisy images is fed with 50% noisy images it forces a portion of the turned OFF pixels due to zero mask noise to get turned ON. As the following CDAE works with this intermediate image it reconstructs other affected pixels completely making the classification task for the CNN easier. For the very same reasons, the DAE-CDAE-CNN architecture performs better than DAE-CNN, CDAE-CNN, DVAE-CNN architectures and achieves classification accuracy of 96.34%. The 50% noisy image recognition accuracy obtained by DAE-CNN, CDAE-CNN, DVAE-CNN are 95.01%, 94.22% and 95.63% respectively. When these 50% noisy data are fed to the CNN classifier without any additional denoising and reconstruction process, the performance shown by the simple CNN is the worst and attains the lowest classification accuracy (85.15%) compared to the other models. Fig. 12(b) supports the fact that, whenever each of these hybrid supervised classifiers give more than 95% accuracy with just 50 iterations, a simple CNN's accuracy was less

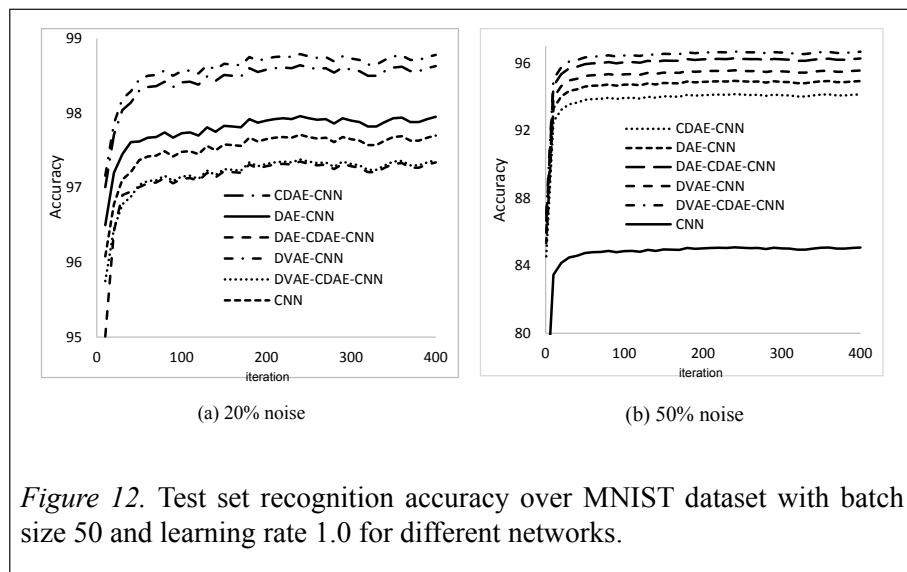


Figure 12. Test set recognition accuracy over MNIST dataset with batch size 50 and learning rate 1.0 for different networks.

than 85% at that time. In times of few initial iterations, the test set accuracy was lower compared to later iterations. This incident was not unexpected as these samples were not conspicuous by these hybrid networks during training period. Still, the classification accuracy improved significantly for test image sets quickly at a lower number of iterations (e.g. up to 100).

Table 1 details the classification of each class individually by all the proposed hybrid models for test set samples after fixed 400 epochs with 20% noise. For

Table 1

Classification Performance of the Hybrid Models in Case of Individual Objects from MNIST Dataset

Noise Level	Models	Accurate Classification (out of 1000 test sample of each class)									Accuracy (%)
		0	1	2	3	4	5	6	7	8	
20%	DAE-CNN	995	992	983	976	968	957	983	985	978	98.01
	CDAE-CNN	997	994	989	984	983	976	988	987	994	977
	DVAE-CNN	999	990	990	986	984	978	989	988	995	980
	DAE-CDAE-CNN	997	991	981	965	940	938	985	980	984	979
50%	DVAE-CDAE-CNN	997	990	981	965	944	940	985	980	983	980
	DAE-CNN	981	961	957	938	936	929	948	965	947	959
	CDAE-CNN	979	955	951	928	925	917	941	960	938	928
	DVAE-CNN	982	965	960	944	947	937	956	968	956	948
	DAE-CDAE-CNN	987	971	970	955	958	944	962	975	958	954
	DVAE-CDAE-CNN	988	973	972	959	958	955	966	977	966	960

DAE-CNN and CDAE-CNN, it is obviously noticeable from the table that they most exceedingly horrendously performed for the numeral “5” and out of 1000 test cases, they classified accurately 957 and 976 cases respectively. The DVAE-CNN classifier performed worst while classifying “5” as well. Still, it performed better than all the other models. For numeral “0” it showed the best classification result. In 999 cases, out of 1000 cases it classified “0” correctly. The DAE-CDAE-CNN and DVAE-CDAE-CNN architecture also misclassified the same digit 62 and 60 times respectively. These perplexities in a couple of manually written numeral images are a result of various handwriting styles of individuals, and furthermore, the arbitrary noise injected in the images slightly misconstrue the patterns with each other. In any case, the proposed models have accomplished best classification for “0” by classifying it correctly 995, 997, 999, 997, 997 cases out of 1000 experiments for DAE-CNN, CDAE-CNN, DVAE-CNN, DAE-CDAE-CNN and DVAE-CDAE-CNN individually. Among the majority of the cases DVAE-CNN accomplished a decent noisy image classification task misclassifying just 116 cases though DAE-CNN, CDAE-CNN, DAE-CDAE-CNN and DVAE-CDAE-CNN misclassified 199, 131, 260 and 257 cases respectively. In the case of 50% noisy images, the worst case occurred with all the models while classifying numeral character “5”. It was misclassified in 71 cases throughout the experiments. The accuracy achieved by the CDAE-CNN, DVAE-CNN, DAE-CDAE-CNN and DVAE-CDAE-CNN architectures in case of classifying “5” are 91.7%, 93.7%, 94.4% and 95.5% respectively. The accuracies calculated in the case of classifying “3” are quite similar: 93.8% for DAE-CNN, 92.8% for CDAE-CNN, 94.4% for DVAE-CNN, 95.5% for DAE-CDAE-CNN and 95.9% for DVAE-CDAE-CNN architectures. These models performed best for classifying numeral “0”. In this case, the classification precisions for DAE-CNN, CDAE-CNN, DVAE-CNN, DAE-CDAE-CNN and DVAE-CDAE-CNN models were 98.1%, 97.9%, 98.2%, 98.7% and 98.8% respectively. Moreover, in most of the cases the occurrences of other numerals misclassified as “0” were few. For all models, the error that occurred mostly were for numerals “8”, “9”, “5” and “3” in the descending order. As this time test set images were corrupted with 50% noises, the shapes of the handwritten numerals were almost totally distorted resulting in less accurate classification performances for the hybrid models than in the case of 20% noisy images because when the images were adulterated with 50% noise it was quite difficult to recognize them even with clear eyes.

Table 2

Sample Handwritten Numeral Images along with their Original and Predicted Class Labels

Sample Image	Actual Label	Classification using				
		DAE-CNN	CDAE-CNN	DVAE-CNN	DAE-CDAE-CNN	DVAE-CDAE-CNN
	2	2	2	2	8	8
	4	6	4	4	8	8
	3	3	5	3	3	3
	7	2	2	2	9	9

Table 2 demonstrates some handwritten numeral images and their corresponding class labels in the original as well as in the reconstructed form. It is clearly seen that the first image was classified correctly as “2” when reconstructed with DAE, CDAE and DVAE, but the reconstruction using DAE-CDAE as well as DVAE-CDAE distorted the pattern, thereby causing the CNN to classify it as “8”. Numeral “4” was classified correctly when reconstructed using CDAE and DVAE, but misclassified as “6” in the case of DAE-CNN and “8” by DAE-CDAE-CNN as well as DVAE-CDAE-CNN. However, the third pattern was classified correctly using all the models expect CDAE-CNN which misclassified it as “5”. On the other hand, the fourth pattern from the table was misclassified by all of the networks. It is important to note that all of these patterns are pretty difficult to identify even by humans because of the diverse writing styles of different persons and adding noise with these ambiguous patterns makes their classification even more difficult.

Table 3 portrays the consequences of the proposed techniques with different prominent works. It, moreover, displays specific feature(s) of individual procedures. It is striking that the proposed models did not use any feature extraction procedure while the vast majority of the current techniques use possibly more than one or maybe a couple of feature extraction techniques. Without utilizing any extra technique for feature extraction, the proposed DAE-CNN, CDAE-CNN, DVAE-CNN, DAE-CDAE-CNN and DVAE-CDAE-CNN models seem to beat the existing strategies. According to the

table, test set classification accuracies when they are corrupted with 50% noise are 95.01%, 94.22%, 95.63%, 96.34%, and 96.74% and with 20% noisy test set images, the accuracies are 98.01%, 98.69%, 98.84%, 97.40 % and 96.74% for DAE-CNN, CDAE-CNN, DVAE-CNN, DAE-CDAE-CNN and DVAE-CDAE-CNN architectures individually. It is clearly visible from the table that DAE-CNN, CDAE-CNN, DVAE-CNN outperform the other two models along with the existing models in case of classifying images subject to regular noise. But whenever it comes to classifying massive noisy images the DAE-CDAE-CNN and DVAE-CDAE-CNN play the frontier role.

Classification Result on CIFAR-10 Dataset

In this section, the proposed methods were tested over the CIFAR-10 data set. From the grey-scaled CIFAR-10 dataset, 50000 sample images were considered for training purpose and 10000 for testing. Sample images were uniformly distributed over the elementary 10 classes. Noiseless initial data were adulterated with 20% Gaussian noise for the training of the autoencoders, whereas 50% noise was injected for the testing purpose only with the intention to check the performances of the proposed models in the noisy environment. Fig. 13 displays some reconstructed images after the noise removal steps using DAE, CDAE, DVAE, DAE-CDAE, DVAE-CDAE respectively in case the images were corrupted with 50% noise. Without any doubt, the DVAE-CDAE provides the best reconstruction in the case of 50% noisy images and the reconstructed images displayed in the figure show the evidence of the statement.

Table 3

A Comparative Description of the Proposed DAE-CNN, CDAE-CNN, DVAE-CNN, DAE-CDAE-CNN and DVAE-CDAE-CNN with Some Contemporary Methods

Work reference	Classification	Noise	Recog. acc.
Bengio et al. (2007)	DBN	0%	98.50%
	Deep net	0%	98.4%
	Shallow net	0%	95%
Glorot (2011)	Sparse rectifier neural network	25%	98.43%
Vincent et al. (2008)	SDAE-3	10%	97.20% %
Vincent et al. (2010)	SVM	25%	98.37%
	SDAE-3	25%	98.5%
Proposed DAE-CNN	CNN	20%	98.01%
		50%	95.01%

(continued)

Work reference	Classification	Noise	Recog. acc.
Proposed CDAE-CNN	CNN	20%	98.69%
		50%	94.22%
Proposed DVAE-CNN	CNN	20%	98.84%
		50%	95.63%
Proposed DAE-CDAE-CNN	CNN	20%	97.40 %
		50%	96.34%
Proposed DVAE-CDAE-CNN	CNN	20%	97.43 %
		50%	96.74%

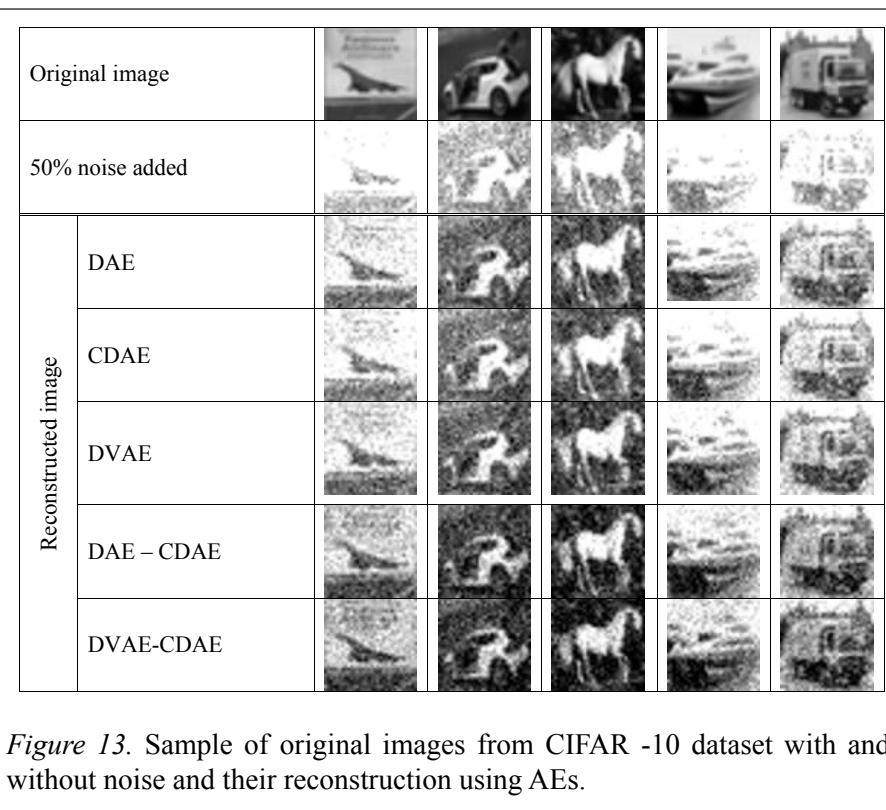


Figure 14 shows the noisy-image classification accuracy of the different proposed models along with a simple CNN in case the images were corrupted with 20% noise as well as 50% noise. The reported values were captured up to 400 iterations. Figure 14(a) depicts the fact that DVAE-CNN performs the best with 20% noisy images achieving an accuracy of 62.8%. The classification performance of DAE-CNN, CDAE-CNN, DAE-CDAE-CNN, DVAE-CDAE-CNN and the simple CNN are 62.37%, 62.69%, 61.88%, 61.93% and 62.04%

respectively. Fig. 14(b) shows the classification accuracy of the proposed models in the case of classifying 50% noisy image data up to 400 epochs. This time, the DVAE-CDAE-CNN architecture achieved the first place with 53.91% accuracy. The DAE-CNN, CDAE-CNN, DVAE-CNN and DAE-CDAE-CNN showed a reasonable classification accuracy (52.95%, 52.5%, 52.64% and 53.63% respectively.). It is clearly visible that the classification accuracy of these models degraded while working with the CIFAR-10 dataset compared to the performance of the models over the MNIST data set. The main reason behind this issue was that in the tiny pictures in the CIFAR-10 data set (32x32 sized) do not give a clear representation of the objects in the image within such a small region. Moreover, the objects were captured in images with different orientationTable 4 details the classification accuracy for each individual object for the test set images after 400 epochs with 20% as well as 50% noise. All the models showed best classification accuracy for the object “Frog”. The DAE-CNN, CDAE-CNN and DVAE-CNN recognized it correctly 704, 702 and 707times respectively. Both the DAE-CDAE-CNN and the DVAE-CDAE-CNN accurately classified it in 699 cases. The worst classification happened while classifying the object “Deer”. Even, the DVAE-CNN architecture that showed the best performance while classifying 20% noisy images misclassified it in 49% cases. In case of classifying 50% noisy images, all the models performed worst for the object “Deer”. The CDAE-CNN architecture misclassified it 626 times which was the highest misclassification result. The DAE-CNN architecture classified it correctly only 7 times more than the CDAE-CNN architecture. The DVAE-CNN, DAE-CDAE-CNN and DVAE-CDAE-CNN architecture showed 42.9%, 37.6% and 39.4% accuracy respectively. The classification result was not up to the mark for the object “Dog”. The DAE-CNN, CDAE-CNN, DVAE-CNN, DAE-CDAE-CNN and DVAE-CDAE-CNN models’ classification accuracies in the case of classifying “Dog” were 39.8%, 39.4%, 39.5%, 40.9% and 41.1% respectively. These models performed best for the object “Frog” with 63%, 62.8%, 62.8%, 63.5% and 63.8% accuracies achieved by DAE-CNN, CDAE-CNN, DVAE-CNN, DAE-CDAE-CNN, DVAE-CDAE-CNN respectively.

Table 5 demonstrates some sample images from the CIFAR-10 datasets and their corresponding class labels in the original as well as in the reconstructed form. The first image was of “Airplane”. All the models misclassified it as “Bird”. All the models except the DAE-CNN classified the second image correctly as “Horse”, whereas DAE-CNN classified it as “Dog”. The third image of a “Ship” was classified accurately by all the models. The last image was of “Truck”. Only DAE-CDAE-CNN and DVAE-CDAE-CNN classified it correctly.

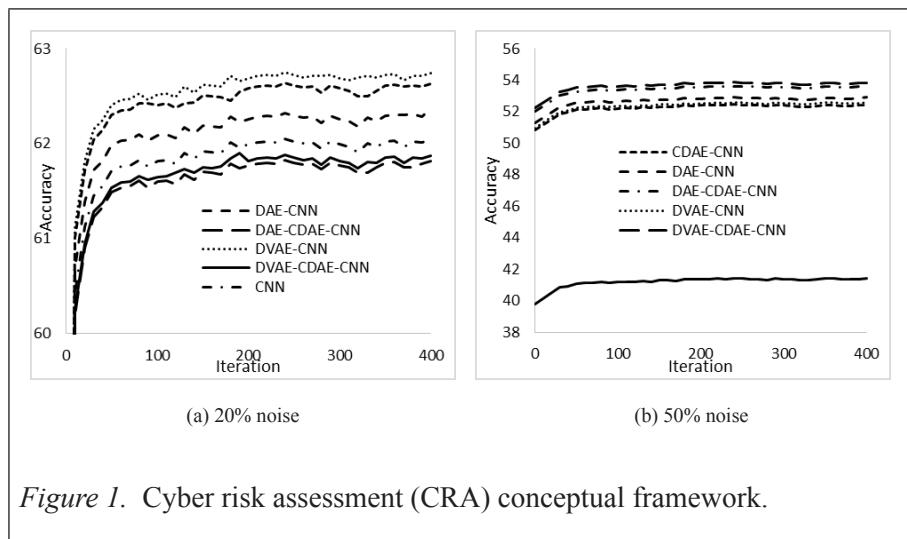


Figure 1. Cyber risk assessment (CRA) conceptual framework.

Table 5

Sample Objects from CIFAR-10 Dataset along with their Original and Predicted Class Labels

Original image	Actual label	Classified with hybrid methods				
		DAE-CNN	CDAE-CNN	DVAE-CNN	DAE-CDAE-CNN	DVAE-CDAE-CNN
	Airplane	Bird	Bird	Bird	Bird	Bird
	Horse	Dog	Horse	Horse	Horse	Horse
	Ship	Ship	Ship	Ship	Ship	Ship
	Truck	Automobile	Automobile	Automobile	Truck	Truck

Table 6 compares the result of the proposed hybrid noisy image classifiers with other prominent works while working over the CIFAR-10 data set along with the particular feature(s) of those models. As per the table, test set accuracies with 50% noise were 52.95%, 52.5%, 52.64%, 53.68% and 53.91%, while with the 20% noise test set, accuracies were 62.37%, 62.69%, 62.8%, 61.88%

and 61.93% for DAE-CNN, CDAE-CNN, DVAE-CNN, DAE-CDAE-CNN and DVAE-CDAE-CNN models respectively. From this table, it is clearly visible that our proposed models outperform some of the existing prominent models in the case of classifying noisy images, especially when the images were subject to massive noises. Moreover, the classifier was the CNN. All the autoencoders and their hybrid models served only for the image denoising task. From the table, it is clearly observable that without prior image denoising by the autoencoders, the performance of the classifier would be disastrous. It is also notable that our models do not need to be trained with images corrupted with noises of different proportions. The DAE, CDAE, DVAE were trained with 20% noisy images only and the CNN was trained with noise-free raw images. Still, the DAE-CDAE-CNN and the DVAE-CDAE-CNN models classified 50% noisy images with very good classification accuracy omitting the necessity for the noisy-image classifiers to be trained with 50% noisy images. The cascading structures of the DAE-CDAE-CNN and DVAE-CDAE-CNN enabled them to show such great performances over the massive noisy data.

Table 6

A Comparative Description of the Proposed DAE-CNN, CDAE-CNN, DVAE-CNN, DAE-CDAE-CNN and DVAE-CDAE-CNN with some Contemporary Methods while Experimenting over CIFAR-10.

Work reference	Classification	Noise	Recog. acc.
Glorot et al. (2011)	Sparse rectifier neural network	25%	50.48%
Traditional CNN (LeCun et al., 1998)	CNN	20%	62.04%
		50%	41.42%
Proposed DAE-CNN	CNN	20%	62.37%
		50%	52.95%
Proposed CDAE-CNN	CNN	20%	62.69%
		50%	52.5%
Proposed DVAE- CNN	CNN	20%	62.8 %
		50%	52.64%
Proposed DAE-CDAE-CNN	CNN	20%	61.88%
		50%	53.68%
Proposed DVAE-CDAE-CNN	CNN	20%	61.93%
		50%	53.91%

Significance of the Proposed Hybrid Models

There are several significant differences between the proposed hybrid methods and the existing ones in terms of noisy-image classification. Conventional models train AEs for denoising in a stacked or standalone way, whereas AEs are trained independently and then cascaded for denoising in any of the proposed hybrid models. The proposed methods use CNN as a classifier rather than MLP or other classifiers as CNN performs well for image classification. The experimental results on benchmark datasets revealed the effectiveness of the proposed hybrid models for both regular and massive noise.

Deep learning-based models have the dependency over training data; therefore, existing models perform well only when they work with the images corrupted with the very same proportion of noise as in the training data and performance degrades when noise level increases. Our proposed hybrid models DAE-CDAE-CNN and DVAE-CDAE-CNN have overcome this problem. Both the architectures are very good at classifying images injected with massive noisy data even if they are trained with images corrupted with regular noise. The underlying cascaded structures of these two models make it possible for them to perform well in this case. These two models use two AEs as image denoiser and both the AEs are trained to reconstruct native images from images subject to the same level of regular noises. So, in both cases, the frontier AE omits a proportion of noise from the input image and the reconstructed image is passed to the following AE for further filtering. As a result, whenever the percentage of noise is massive in the input images these two noisy-image classifiers perform better than other models.

On the other hand, the proposed DAE-CNN, CDAE-CNN and DVAE-CNN models performed well for regular noise. As single AEs in these three models are trained with regular level noisy images, their standalone structure is sufficient to denoise regular noisy images which are later easy to classify with CNN.

CONCLUSION

Conventional image classifiers perform really well with preprocessed data generated in the laboratory. But when they are employed to classify real world data, most often these images are corrupted with noise during acquisition and transmission. As a result, there is a high chance that they would fail drastically when applied in real life tasks. The solution to this problem is to denoise the images prior to feeding to the classifier. This research work proposed five supervised deep architectures named DAE-CNN, CDAE-

CNN, DVAE-CNN, DAE-CDAE-CNN and DVAE-CDAE-CNN among which the first three perform well when the images are subjected to a small amount of noise, whereas, the last two are for classifying massive noisy data. These models utilize the ideas of various autoencoders and along with CNN construct classifiers for noisy image data. These deep models have the ability to filter noise from the image data and classify them by learning latent feature representations from them. These models' classification accuracy over MNIST and CIFAR-10 datasets (corrupted with noise of different proportions) gives evidence that they have the capability to learn hierarchical representations of the images. Still, there are scopes for further developments in future. The different hybrid models proposed here are good with a different level of noises. Our future research work would focus on building a standalone model using these techniques that would be able to classify images adulterated with any proportion of noise.

REFERENCES

- Agostinelli, F., Anderson, M. R., & Lee, H. (2013). Adaptive multi-column deep neural networks with application to robust image denoising. *Proceedings of the Advances in Neural Information Processing Systems*, 1493-1501.
- Akhand, M. A. H., Ahmed, M., & Rahman, M. H. (2016). Convolutional neural network based handwritten Bengali and Bengali-English mixed numeral recognition. *International Journal of Image, Graphics and Signal Processing*, 8(9), 40-50. doi: 10.5815/ijigsp.2016.09.06
- Akhand, M. A. H., Ahmed, M., Rahman, M. H. & Islam, M. M. (2017). Convolutional neural network training incorporating rotation based generated patterns and handwritten numeral recognition of major Indian scripts. *IETE Journal of Research (TIJR)*, Taylor & Francis, 63(Online), 19 pages. doi: 10.1080/03772063.2017.1351322
- Arigbabu, O. A., Ahmad, S. M. S., Adnan, W. A. W., Yussof, S., & Mahmood, S. (2017). Soft biometrics: Gender recognition from unconstrained face images using local feature descriptor. *arXiv Preprint arXiv:1702.02537*.
- Bar, Y., Diamant, I., Wolf, L., & Greenspan, H. (2015, March). Deep learning with non-medical training used for chest pathology identification. *Proceedings of Society for Optics and Photonics*, 94140V-94140V. doi: 10.1117/12.2083124.

- Bengio, Y., Lamblin, P., Popovici, D., & Larochelle, H. (2007). Greedy layer-wise training of deep networks. *Proceedings of the Advances in Neural Information Processing Systems*, 153-160.
- Bengio, Y. (2009). Learning deep architectures for AI. *Foundations and Trends® in Machine Learning*, 2(1), 1-127.
- Bezdek, J. C., Hall, L. O., & Clarke, L. (1993). Review of MR image segmentation techniques using pattern recognition. *Medical Physics*, 20(4), 1033-1048.
- Bosch, A., Zisserman, A., & Munoz, X. (2007, July). Representing shape with a spatial pyramid kernel. *Proceedings of the 6th ACM International Conference on Image and Video Retrieval*, 401-408.
- Bourlard, H., & Kamp, Y. (1988). Auto-association by multilayer perceptrons and singular value decomposition. *Biological Cybernetics*, 59(4), 291-294.
- Bouvie, J. (2006). *Notes on convolutional neural networks*, Center for Biological and Computational Learning, Department of Brain and Cognitive Sciences, MIT, Cambridge.
- Burger, H. C., Schuler, C. J., & Harmeling, S. (2012, June). Image denoising: Can plain neural networks compete with BM3D?. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2392-2399.
- Cho, K. (2013). Boltzmann machines and denoising autoencoders for image denoising. *arXiv Preprint arXiv:1301.3468*.
- Cireşan, D. C., Meier, U., Masci, J., Gambardella, L. M., & Schmidhuber, J. (2011). High-performance neural networks for visual object classification. *arXiv Preprint arXiv:1102.0183*.
- Cireşan, D., Meier, U., Masci, J., & Schmidhuber, J. (2011, July). A committee of neural networks for traffic sign classification. *Proceedings of the 2011 International Joint Conference on Neural Networks (IJCNN)*, 1918-1921.
- Ciresan, D. C., Meier, U., Gambardella, L. M., & Schmidhuber, J. (2011, September). Convolutional neural network committees for handwritten

- character classification. *Proceedings of the 2011 International Conference on Document Analysis and Recognition (ICDAR)*, 1135-1139.
- Coates, A., Ng, A., & Lee, H. (2011, June). An analysis of single-layer networks in unsupervised feature learning. *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, 215-223.
- Coifman, R. R., & Donoho, D. L. (1995). Translation-invariant de-noising. *Wavelets and Statistics*, 103, 125-150.
- Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3), 273-297.
- Du, B., Xiong, W., Wu, J., Zhang, L., Zhang, L., & Tao, D. (2017). Stacked convolutional denoising auto encoders for feature representation. *IEEE Transactions on Cybernetics*, 47(4), 1017-1027.
- Elad, M., & Aharon, M. (2006). Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image Processing*, 15(12), 3736-3745.
- Glorot, X., Bordes, A., & Bengio, Y. (2011, June). Deep sparse rectifier neural networks. *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, 315-323.
- Gondara, L. (2016, December). Medical image denoising using convolutional denoising autoencoders. *Proceedings of the 2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*, 241-246.
- Hinton, G. E., Osindero, S., & Teh, Y. W. (2006). A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7), 1527-1554.
- Huang, F. J., & LeCun, Y. (2006). Large-scale learning with svm and convolutional nets for generic object recognition. *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1-8.
- Im, D. J., Ahn, S., Memisevic, R., & Bengio, Y. (2017). Denoising criterion for variational auto-encoding framework. *Association for the Advancement of Artificial Intelligence*, 2059-2065.

- Jain, V., & Seung, S. (2009). Natural image denoising with convolutional networks. *Proceedings of the Advances in Neural Information Processing Systems*, 769-776.
- Kingma, D. P., & Welling, M. (2013). Auto encoding variational bayes. *arXiv Preprint arXiv:1312.6114*.
- Kingma, D. P., & Welling, M. (2014). Stochastic gradient VB and the variational auto encoder. *Proceedings of the Second International Conference on Learning Representations (ICLR)*, 1-14.
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Proceedings of the Advances in Neural Information Processing Systems*, 1097-1105.
- LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278-2324. doi: 10.1109/5.726791
- LeCun, Y., Cortes, C., & Burges, C. J. (2010). MNIST handwritten digit database. *AT&T Labs [Online]*. Retrieved from: <http://yann.lecun.com/exdb/mnist>
- Lee, H., Grosse, R., Ranganath, R., & Ng, A. Y. (2009). Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. *Proceedings of the 26th Annual International Conference on Machine Learning*, 609-616.
- Liu, T., Fang, S., Zhao, Y., Wang, P., & Zhang, J. (2015). Implementation of training convolutional neural networks. *arXiv Preprint arXiv:1506.01195*.
- Lu, D., & Weng, Q. (2007). A survey of image classification methods and techniques for improving classification performance. *International Journal of Remote Sensing*, 28(5), 823-870.
- Mairal, J., Bach, F., Ponce, J., & Sapiro, G. (2009). Online dictionary learning for sparse coding. *Proceedings of the 26th Annual International Conference on Machine Learning*, 689-696.
- Masci, J., Meier, U., Cireşan, D., & Schmidhuber, J. (2011). Stacked convolutional auto-encoders for hierarchical feature extraction. *Artificial Neural Networks and Machine Learning (ICANN)*, 52-59.

- Matsugu, M., Mori, K., Mitari, Y., & Kaneda, Y. (2003). Subject independent facial expression recognition with robust face detection using a convolutional neural network. *Neural Networks, 16*(5), 555-559.
- Norouzi, M., Ranjbar, M., & Mori, G. (2009, June). Stacks of convolutional restricted Boltzmann machines for shift-invariant feature learning. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVRP)*, 2735-2742.
- Olshausen, B. A., & Field, D. J. (1997). Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision Research, 37*(23), 3311-3325.
- Perona, P., & Malik, J. (1990). Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 12*(7), 629-639.
- Rudin, L. I., & Osher, S. (1994). Total variation-based image restoration with free local constraints. *Proceedings of the IEEE International Conference on Image Processing, 1*, 31-35.
- Sanches, J. M., Nascimento, J. C., & Marques, J. S. (2008). Medical-image noise reduction using the Sylvester–Lyapunov equation. *IEEE Transactions on Image Processing, 17*(9), 1522-1539.
- Scherer, D., Müller, A., & Behnke, S. (2010). Evaluation of pooling operations in convolutional architectures for object recognition. *Proceedings of the International Conference on Artificial Neural Networks (ICANN)*, 92-101.
- Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks, 61*, 85-117.
- Sharif Razavian, A., Azizpour, H., Sullivan, J., & Carlsson, S. (2014). CNN features off-the-shelf: An astounding baseline for recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 806-813.
- Shin, H. C., Orton, M. R., Collins, D. J., Doran, S. J., & Leach, M. O. (2013). Stacked autoencoders for unsupervised feature learning and multiple organ detection in a pilot study using 4D-patient data. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 35*(8), 1930-1943.

- Singh, A. K., Shukla, V. P., Biradar, S. R., & Tiwari 1, S. (2014). Multiclass noisy image classification based on optimal threshold and neighboring window denoising. *International Journal of Computer Engineering Science (IJCES)*, 4(3), 1-11.
- Simonyan, K., Vedaldi, A., & Zisserman, A. (2013). Deep fisher networks for large-scale image classification. *Advances in Neural Information Processing Systems*, 163-171.
- Subakan, O., Jian, B., Vemuri, B. C., & Vallejos, C. E. (2007). Feature-preserving image-smoothing using a continuous mixture of tensors. *Proceedings of the 11th International Conference on Computer Vision (ICCV)*, 1-6.
- Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., & Fergus, R. (2013). Intriguing properties of neural networks. *arXiv Preprint arXiv:1312.6199*.
- Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning internal representations by error propagation. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, 1, 318-362.
- Turaga, S. C., Murray, J. F., Jain, V., Roth, F., Helmstaedter, M., Briggman, K., et al. (2010). Convolutional networks can learn to generate affinity graphs for image segmentation. *Neural Computation*, 22(2), 511-538.
- Vincent, P., Larochelle, H., Bengio, Y., & Manzagol, P.A. (2008). Extracting and composing robust features with denoising autoencoders. *Proceedings of the 25th International Conference on Machine Learning*, 1096-1103.
- Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., & Manzagol, P. A. (2010). Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 11(Dec), 3371-3408.
- Xie, J., Xu, L., & Chen, E. (2012). Image denoising and inpainting with deep neural networks. *Proceedings of the Advances in Neural Information Processing Systems*, 341-349.
- Xu, L., Ren, J. S., Liu, C., & Jia, J. (2014). Deep convolutional neural network for image deconvolution. *Proceedings of the Advances in Neural Information Processing Systems*, 1790-1798.

- Zeiler, M. D., Krishnan, D., Taylor, G. W., & Fergus, R. (2010). Deconvolutional networks. *Proceedings of the 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2528-2535.
- Zeiler, M. D., Taylor, G. W., & Fergus, R. (2011). Adaptive deconvolutional networks for mid- and high-level feature learning. *Proceedings of the 2011 IEEE International Conference on Computer Vision (ICCV)*, 2018-2025.