

Deep networks in identifying CT brain hemorrhage

Abdulkader Helwan^{a,*}, Georges El-Fakhri^b, Hadi Sasani^a and Dilber Uzun Ozsahin^{a,b}

^a*Department of Biomedical Engineering, Near East University, Near East Boulevard, TRNC, Nicosia, Turkey*

^b*The Gordon Center for Medical Imaging, Radiology, Massachusetts General Hospital, Harvard Medical School, Boston, USA*

Abstract. Deep learning algorithms have recently been applied to solving challenging problems in medicine such as medical image classification and analysis. In some areas, those algorithms have outperformed the human medical experts experience in diagnosis. Thus, in this paper we apply three different deep networks to solve the problem of brain hemorrhage identification in CT images. The motivation behind this work is the difficulty that radiologists encounter when diagnosing a hemorrhagic brain CT image, in particularly in the early stages of the brain bleeding. Autoencoder (AE), stacked autoencoder (SAE), and convolutional neural network (CNN) are employed and trained to classify the CT images into hemorrhagic or non-hemorrhagic. Experimentally, it was found that all employed networks performed differently in terms of accuracy, error reached, and training time. However, stacked autoencoder has achieved a higher accuracy and lesser error compared to other used networks.

Keywords: Deep learning, deep networks, hemorrhage, autoencoder, stacked autoencoder, convolutional neural network

1. Introduction

Intracerebral hemorrhage (ICH) is an important cause of morbidity and mortality in the world [1]. Cerebral hemorrhage is a type of a stroke caused by an artery bursting and causing bleeding in the surrounded tissues. ICH is affecting 220 people out of every 100000 in Asia while 7 people out of every 100000 in West each year. Females are affected more than men by ratio of 3:2. ICH is a medical emergency to be treated. Rapid diagnosis and attentive management of patients with ICH is crucial because the mortality rate after the hemorrhage within in the first 30 days is high (up to 50%) [2]. Size of the hemorrhage region, its shape, and its location within the skull are important parameters

in diagnosis. Neuroimaging is highly important in establishing the diagnosis of ICH. As the first-line modality and due to its wide availability, lower cost and rapidity, CT is usually preferred and for detection of acute blood, non-contrast CT is highly sensitive and specific [3, 4].

Computer-Aided Diagnosis (CAD) system is very helpful for diagnosis and also treatment of diseases [5]. CAD systems are usually domain-specific that they can analyze different kinds of input such as symptoms, laboratory tests results, medical images, etc. They are mostly useful in reducing the workload of experts [5, 6]. With the use of CAD system, the demand for well-organized medical imaging data storage and retrieval techniques has increased [7]. Classifying the brain CT scan abnormalities is considered a tough task for radiologists. Hence, over the past decades, computer aided diagnosis (CAD) systems have been developed to extract useful information from brain CT to help doctors in having a

*Corresponding author. Abdulkader Helwan, Department of Biomedical Engineering, Near East University, Near East Boulevard, TRNC, Nicosia, 99138 Turkey. E-mail: abdulcadir.helwan@neu.edu.tr.

quantitative insight about a brain [8]. However, those CAD systems haven't achieved a high significance level to make decisions on the type of medical conditions found in a brain CT scan. Thus, their role was left as visualization functionality that helps doctors in making decisions.

Recently, a rapid rise of Deep learning was seen. Deep learning based networks have shown a great efficiency when applied to various medical areas such as medical image analysis [9], medical image classification [10, 11], medical organs detection [12], disease detection [13], etc... these networks own the paradigms of more biologically inspired structures than the traditional networks [9]. This rise in networks performance, due to their 'biologically inspired' deep structure, motivated many researchers to apply some types of deep networks to the brain hemorrhage CT images classification, which can be a tedious task, even for medical experts, due to the difficulty in visualizing the hemorrhage in its early stages. Autoencoder (AE), Stacked autoencoder (SAE), and Convolutional neural network (CNN) are employed in this study, aiming to accurately identify hemorrhage in brain CT images obtained from the Near East Hospital, Cyprus.

This paper is structured as follows: section one is an introduction, section two is a deep insight of deep learning approach. Section three describes the training of the three employed networks. Section four shows the performance of networks in addition to the results discussion. Section six is a conclusion of the presented work.

2. Related works

In a study conducted by Santosh et al. [14], the possibility of diagnosing brain hemorrhage was investigated using an image segmentation of CT scan images using watershed method and feeding of the appropriate inputs extracted from the brain CT image to an artificial neural network for classification. The authors found that automatic detection of hemorrhage was a very complex task; after using watershed algorithm the boundaries of each region were continuous and over segmentation problems were encountered. This process was little bit time consuming. On other hand, in testing process the percentage of true detection was 80% and while validating it was found to be 75%. The system had better accuracy (84.62%), better sensitivity (88.89%) and MCC gives the quality of

classification which is 95.97 in case of system with 25 hidden layer neurons. The proposed system could be taken in to the next level by implementing identification for epidural and subarachnoid hemorrhage.

In a similar study done by Mahajan et al. [15], authors studied brain hemorrhage in more refined manner by feeding CT images and identified the type of brain hemorrhage using watershed algorithm along with artificial neural network (ANN). The features were extracted by using Grey Level Co-occurrence Matrix (GLCM). The feed forward Back Propagation Neural Network was used to classify the type of hemorrhage. They concluded that various diagnosis techniques for brain hemorrhage which required high segmentation, noise removal, accuracy, etc; in their study these problems were overcome by using advanced neural network in terms of accuracy, speed and robustness.

In another study [16] to detect brain hemorrhage in CT images and classification, Otsu's method was used to extract hemorrhage region from images by segmenting them, then discrimination features of regions of interests are extracted. Images were classified based on computed feature of ROI. Weka tool was used for classification and testifying parts. The rate of diagnosing brain hemorrhage was 100% and the achieved accuracy was 92%.

Furthermore, Gong et al. [17] focused on dividing brain CT images into regions; where each region could either be normal or hemorrhagic. For images containing hemorrhages, the regions which did not include hemorrhage were treated as normal regions resulting in a highly imbalanced dataset. The researcher had utilized an image segmentation scheme that used ellipse fitting, background removal and wavelet decomposition technique. The weighted precision and recall value for this approach were approximately 83.6% and 88.5%, respectively.

3. Deep into deep learning

Deep learning methods can be considered as a bunch of machine learning techniques that gain the ability of learning features in a hierarchical way; from lower level to higher level through building of a deep architecture [18]. These deep learning techniques own the ability of learning features automatically at multiple levels. This, therefore, allows the system to learn some complex mapping function: $X \rightarrow Y$ directly from input data, which prevents the need

of human-crafting features. For the high level feature abstraction, this deep learning ability is very significant [19]. This is due the difficulty in describing the high-level features directly from training data.

Deep learning is called deep as it depicts neural network structures of multi-hidden layer; in contrast to the conventional networks which used to have a single hidden layer referred to as shallow networks [20]. These deep networks are mimicked by biological features that describe the structure of the human brain and its visualization principles. Such features include:

The distributed portrayal of cognition and learning at each hidden layer.

The extraction of distinct and different features by the hidden neurons in each layer.

Disparate neurons can be active simultaneously.

In deep networks, it is generally conceived that the different levels features are extracted layer by layer, for instance the first hidden layer extracts some primary or low-level features that are some functions of the input (or parts of input) such as corners and edges of the input image. Other hidden layers are to extract more distinctive and defined features and these features are all further combined into higher levels and well more defined features in the following layers such as objects in the image, and so on. This way of extracting different levels features can be seen as an abstraction of distinct levels of features which allows the deep network to gain a hierarchical representation of knowledge [21]. Thus, this helps a deep network to find more complex space functions as more hidden layers are added to the network.

Many deep networks models have been proposed by many researchers. In 2006, Hinton et al. proposed a new network model referred to as deep belief networks (DBNs) [22]. This model used a new unsupervised learning algorithm in which a deep network is greedily trained, layer by layer. Moreover, many other deep neural models have been proposed such as convolutional neural networks (CNN), recurrent neural networks (RNN), autoencoders (AE), and stacked autoencoders (SAE) etc. Note that these models were successfully employed in various areas such as prediction [22], medical diagnosis [23], image dimensionality reduction [24], natural language processing [25, 26], robotics [27], and many others. The crucial reasons of deep neural network success in various fields are mainly the significant increase of computer abilities and performances, the availability

of huge amount of data in some fields, and the drop in the hardware prices. be taken in to the next level by implementing identification for epidural and sub-arachnoid hemorrhage.

3.1. Autoencoder and stacked autoencoder

Autoencoders (AE) are mainly multilayer feed-forward networks that work on replicating their corresponding input at the output during training. Those networks have a different way of initializing weights. Here, the weights are initialized using a generative learning algorithm, which means that the network doesn't have to be deterministic of outputs classes since it uses unsupervised learning techniques in its primary training phases. This helps in providing good initiated weights for the network [27].

An auto encoder is a feedforward neural network that is trained first to learning the inputs features in an unsupervised manner; in other words, outputs are the same as inputs, no output labeling. This Training technique helps this network to learn the underlying features of the training data or images that are importantly need for the construction of same image at the output layer. This learning technique is called "Pre-training" and it is the same step in training a deep network. The outputs are the input themselves since it is unsupervised learning [28]. During pre-training the input-hidden layer weights are saved as well as the hidden-output layer activations. These saved weights are then used in the second training phase which is called fine-tuning. This learning technique is a supervised learning technique where the input data are labeled and the backpropagation learning technique is used for training the network. However, the input weights should be equal to the input weights that were saved during the pre-training; and the hidden-output weights can be initiated randomly.

The required time and cost that may be needed for labeling the training data beside the unavailability of labeled data, sometimes, are mainly the motivation factors behind using the auto encoders and therefore generative architectures. Hence, this type of generative learning suffices in situations where large unlabeled data are available, while the available labeled data are of small number. Accordingly, the network can be generatively trained using the unlabeled data as in case of auto encoders while the small labeled data can be used in the fine-tuning of the final network which is the deterministic approach of in

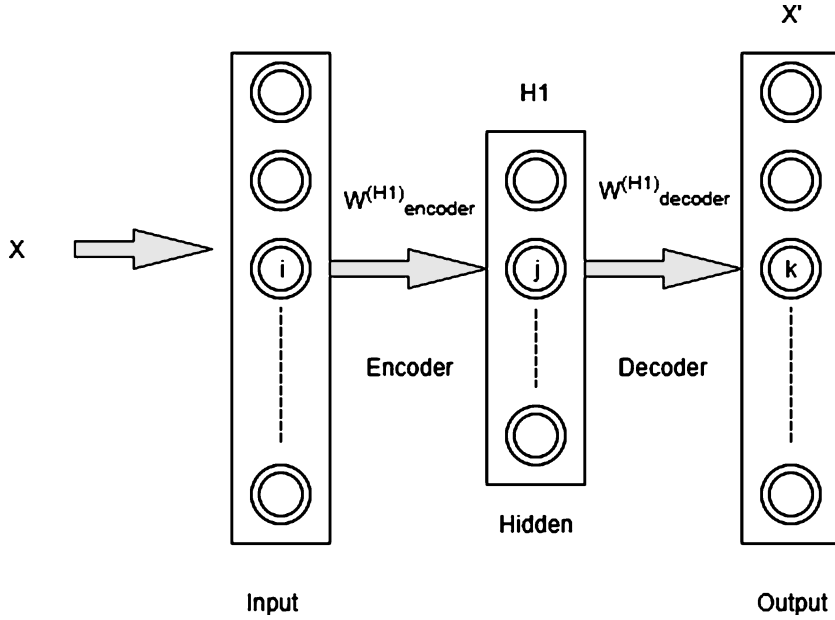


Fig. 1. Autoencoder.

which the deep network is discriminately fine-tuned, in a supervised manner, by the available small number of input data and targets.

As seen in Fig. 1, an auto encoder can be considered as an encoder-decoder system, where the input-hidden layer ($X-H1$) receives the input and extract some essential features to be then used in constructing the output in the hidden-output layer ($H1 - X'$). It should be noted that the number of neurons in the input (i) and output (k) layers are equal for the input and outputs are the same, no data labeling; unsupervised learning. However, the number of hidden neurons (j) is smaller than k , this allows the network to act as a sort of data compression system; feature extractor. Stacking more autoencoders result in a deeper and more hierarchical structure of knowledge extraction from the input data. Such deep structures are referred to as stacked autoencoder (SAE). These structures can be significantly efficient in extracting low to high levels feature abstractions of the input data layer (Fig. 2).

In generative network architectures, the used training algorithm is known as “greedy layer-wise pre-training”. This algorithm was first proposed by Hinton et al. [30] to train a Deep belief network layer by layer. The main idea of such approach is that each hidden layer can be separately trained as in the case of single hidden-layer networks. The whole network is later stacked and fine-tuning takes place; where the network is trained in a supervised manner using

the conventional backpropagation learning algorithm [30, 31].

- Since the auto encoder is fundamentally a feedforward network, the training is described below.

Encoder mode:

$$H1(x) = g(m(x)) = S(b^{(H1)} + W_{encoder}^{(H1)}x) \quad (1)$$

$$Y = X' = S(b^{(X')} + W_{encoder}^{(H1)}H1(x)) \quad (2)$$

Where, $m(x)$ and $n(x)$ are the pre-activations of the hidden and output layers $L1$ and y respectively; $b(H1)$ and $b(X')$ are the biases of the hidden and output layers $H1$ and respectively; S is the non-linearity such as the sigmoid; and finally y is the output X' .

Figure 2 shows a stacked autoencoder of 2 hidden layers $H1$ and $H2$. A greedy layer-wise training can be used to train this network, starting by feeding the input hidden layer $H1$ and $H2(x)$ as the output; note that $H2(x)$ has target data as the input; no data labeling. This network (Input-Hidden 2) is then trained in an unsupervised manner using backpropagation algorithm where the weights connection between the input layer and $H1$ are saved to be later on used [31].

Upon training, the input layer is then removed and $H1$ becomes the input, $H2$ the hidden layer, and output follows last (i.e. layer Y). Again, the network

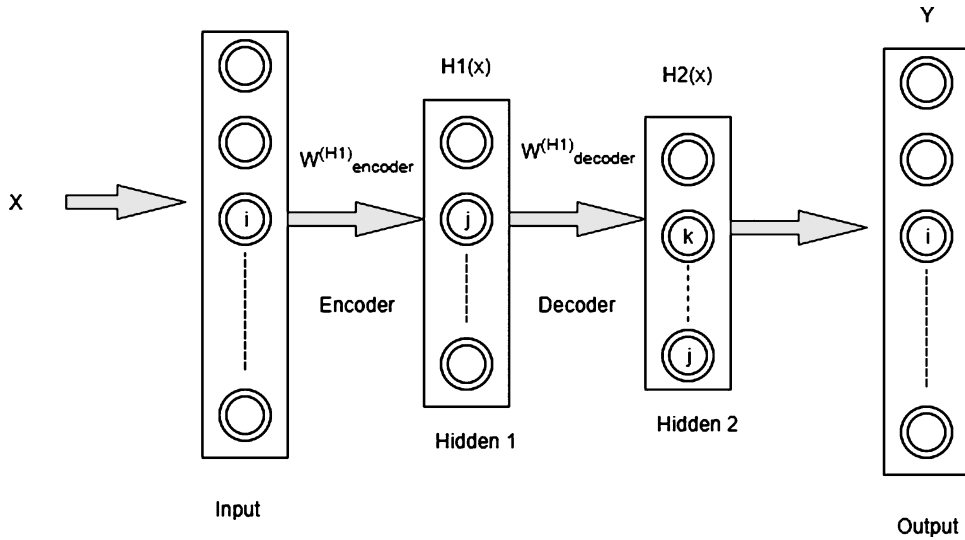


Fig. 2. Stacked autoencoder.

(Hidden1-Output) is trained, however the activation values of H1 are now considered as inputs to the hidden layer H2(x), and the output layer made the same as input H1(x). Note that the weights between H1(x) and H2(x) are also saved here.

Finally, the whole network layers are stacked all together and the whole network is fine-tuned as in supervised backpropagation learning algorithm; where outputs are labeled. Note that the previously saved weights between the first and second hidden layers are used here while last hidden layer and output layers weights are randomly initialized.

3.2. Convolutional neural network

Convolutional neural network (CNN) has been employed successfully for several tasks in medical image classification and analysis [10, 13], medical image segmentation [32, 33], biomedical text classification [34]. In such networks, features extraction from input images is the principal purpose of convolution as this mathematical operation preserves the spatial relationship between pixels. Generally, a convolutional neural network relies on architectural features which include the receptive field, weight sharing and pooling operation to take into account the 2D characteristic of structured data such as images [35]. The concept of weight sharing for convolution maps drastically reduce model parameters; this has the important implications that the model is less prone to over-fitting as compared to fully connected models of comparable size. The pooling operation essentially

reduces the spatially dimension of input maps and allow the CNN to learn some invariance to moderate distortions in the training; this feature enhances the generalization of the CNN at test time as model is more tolerant to moderate distortion in the test data [36, 37].

Figure 3 shows a typical CNN architecture. Basically, it consists of convolution layers, pooling layers and the fully connected layers as shown in Fig. 3. In the first layer, n convolutional filters of size $a \times a$ are used to generate n convolution or feature maps (C1) of size $i \times i$ by sliding the filter over the image and convolving it with the square input data that fits the kernel. Note that filters act as feature detectors from the original input image. The following layer is called pooling or sub-sampling in which the dimensionality of features maps generated in the first layer is reduced.

This operation requires a selection window size of $b \times b$ from each feature map and taking the largest element from the rectified feature map within that window. Instead, we may take the average (Average Pooling) or sum of all elements in that selected window; however, taking the maximum (Max Pooling) has been more common and shown to work better [38]. Thus, pooling layer (S1) is composed of n feature maps of size $j \times j$; where, $j = i/b$ [20]. Convolution and pooling layers may be repeated many times depending on the CNN desired architecture, but finally, they reach to the final layer of the network named, The Fully Connected layer. This layer is a traditional Multi-Layer Perceptron

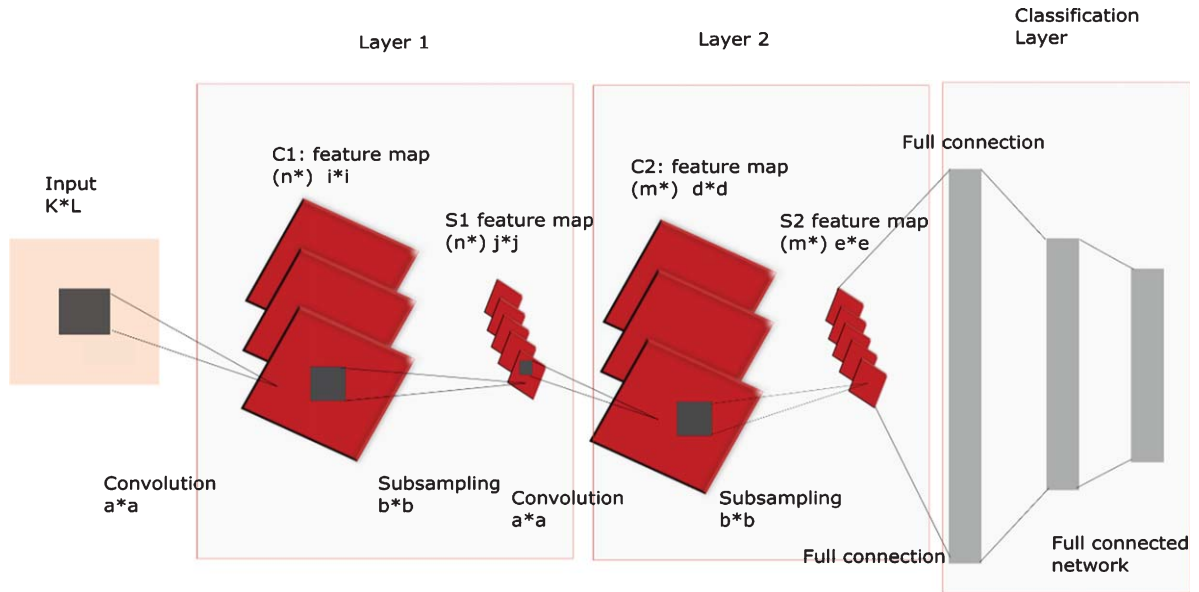


Fig. 3. Convolutional neural network.

that uses a Softmax activation function in the output layer. The features from the previous layers are forward-propagated through the network and fed into this fully connected layer with an output layer of Softmax units. For learning the classifier model, the conventional backpropagation learning algorithm can be used here to train the fully connected network and update these model parameters via gradient descent update rule [39].

4. Deep models training

This work presents a deep approach for the brain hemorrhage identification. This approach is based on extracting the low to high levels of abstractions of features from two different types of brain CT images of various hemorrhagic medical conditions using deep learning based neural networks. Those features are then what distinguish the class of the brain images, i.e., hemorrhagic or not. Note that the images used in this research are of different types of hemorrhage; however, in this work, we attempt to identify whether the CT slice contains hemorrhage or not, regardless of the hemorrhage type. A total number of 2527 images collected from the Near East Hospital [40] were used for training and testing the employed deep networks. Figure 4 shows some normal and hemorrhagic CT slices of the brain. Table 1 shows the learning scheme that is used in this work.

4.1. Autoencoder and Stacked autoencoder training

As discussed in Section 3, the autoencoders are trained using an algorithm known as a greedy layer-wise training, in which each layer can be separately trained, and then the whole network is coupled back. It is important to note that the trained network is still generative at this stage (Fig. 1), and fine tuning the whole stacked network (pre-trained network) makes it discriminative (Fig. 2), then it can be used for classification purposes. The used networks are trained and simulated on a dual core, intel (R) Core (TM) (3.6 GHz) GPU with 32 GB RAM, with programs written and run in a MATLAB environment. Also, note that the aim of training different deep networks is to investigate the one that performs better and discuss the possible reasons. Thus, an autoencoder was first pre-trained on 1004 normal images; have no hemorrhage, and 1138 images with hemorrhage conditions. The three-layer autoencoder is then fine-tuned using backpropagation algorithm (supervised learning) to gain the power of classifying the brain images into two classes: hemorrhagic and non-hemorrhagic images. Figure 5 shows the two phases of an autoencoder where the network is first trained to generate the inputs in the output layer from the distinct features extracted in the hidden layer. Hence, the number of outputs neurons is equal to the number of input neurons in this

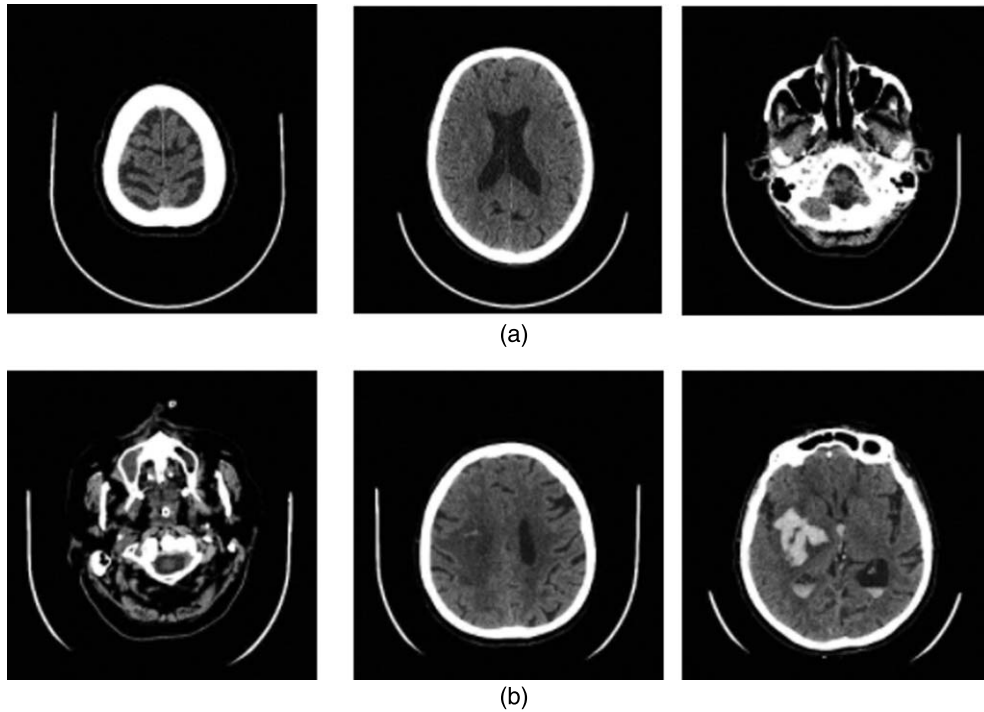


Fig. 4. Sample of normal and hemorrhagic CT slices; (a) show some normal brain images, (b) shows some hemorrhagic brain.

Table 1
Learning scheme

	Normal Brain Images	Hemorrhagic Brain Images	Total
Train	1004	1138	2142
Test	136	249	385
Total	1140	1387	2527

phase Fig. 5(a). Figure 5(b) shows the structure of the autoencoder in the fine-tuning phase in which the outputs are labeled and the whole network is trained, using gradient descent, to classify the brain CT images into hemorrhagic and non-hemorrhagic images. Note that the number of input neurons of the autoencoder (Fig. 5b) is 65536 for the input images size is 512×512 pixels, and the number of output neurons is 2, i.e. the number of classes: hemorrhagic and non-hemorrhagic.

Figure 6 shows the stacked autoencoder that was created by stacking two pre-trained autoencoders to each other. Similarly, the network was trained (fine-tuned) on the same images used for pre-training the autoencoders. Same images were used to train this SAE, hence same number of input neurons it has in its input layer. It also comprises of two hidden layers of 45 and 35 neurons, respectively; as

we used two autoencoders to build up this stacked autoencoder.

Table 2 shows the learning parameters of both autoencoder and stacked autoencoder. It can be seen that AE and SAE have different parameters values, in terms of number of hidden layers, learning rate, and number of iterations. As seen, the SAE was capable of achieving a lower mean square error when learning; however, that was achieved in a longer time and with more iterations compared to those of AE. Obviously, this is due to the number of hidden layers found in the SAE which makes it a deeper structure than that of AE. Consequently, this requires more time and iterations to reach global optima during learning.

Figure 7(a&b) shows the error variations with respect to the Epochs increasing during fine-tuning of both.

AE and SAE, respectively. It is noted that both networks were trained well; however, the increase of depth of SAE makes it more difficult to train, i.e. SAE required longer time to reach the minimum square error (MSE) and also more number of iterations. Moreover, it is important to mention that this difference in time and iterations number of SAE ends up with a lower MSE than that reached by AE.

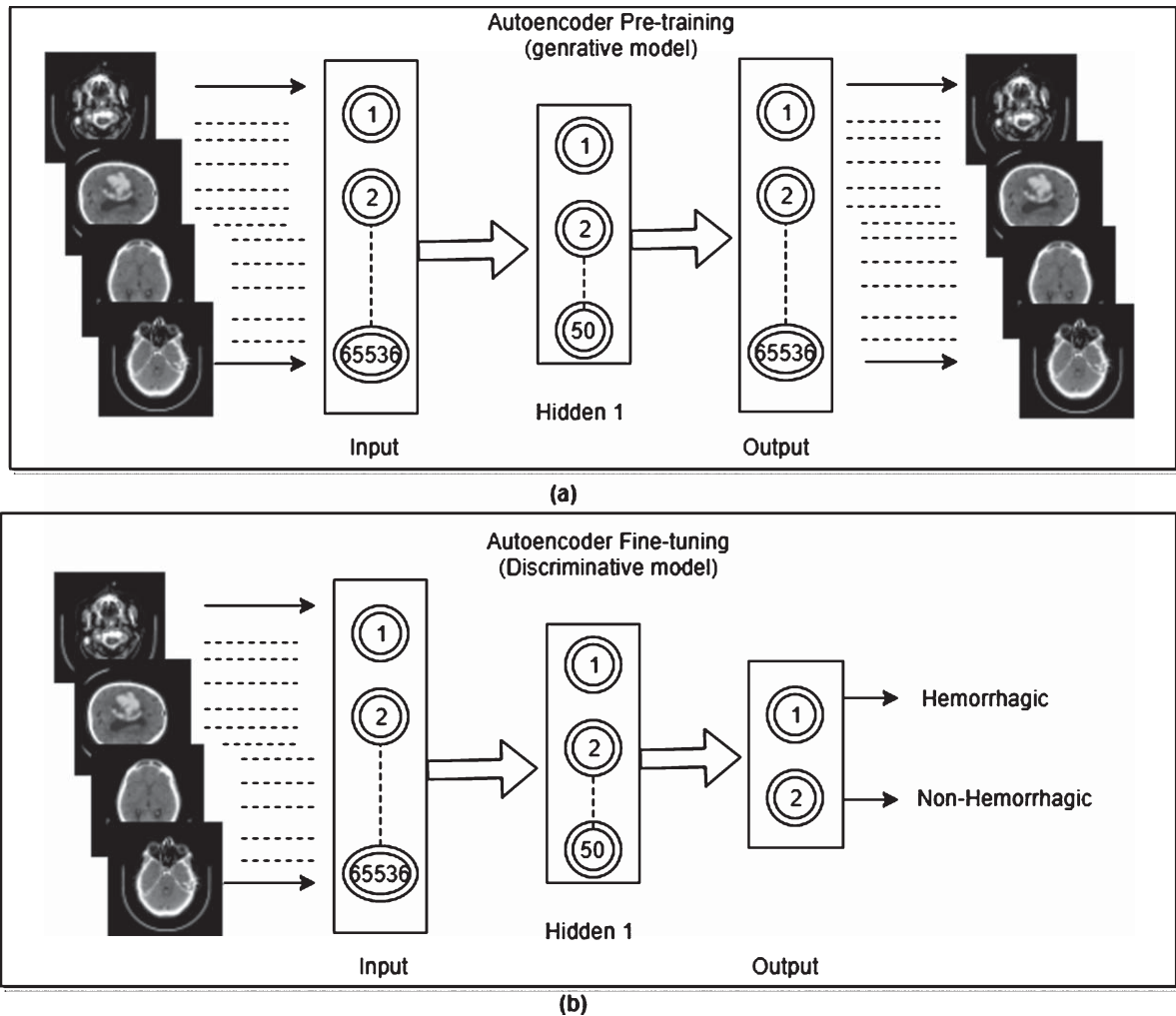


Fig. 5. Autoencoder training stages.

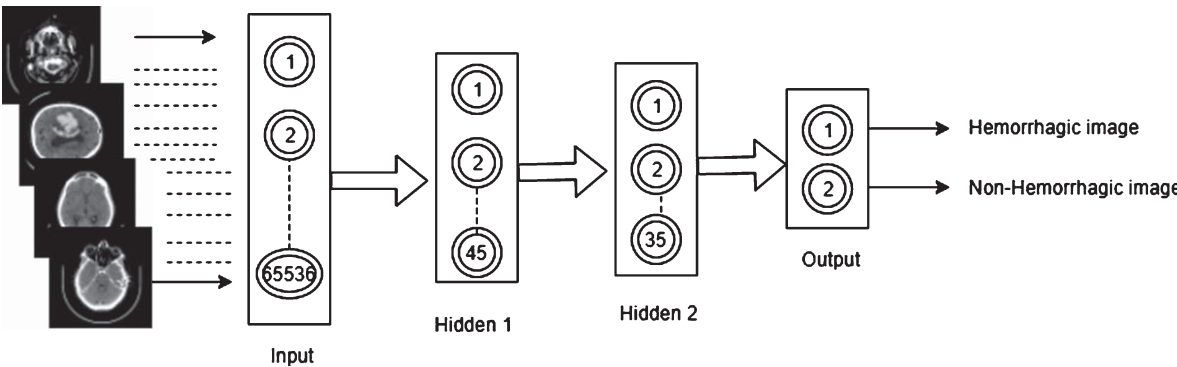


Fig. 6. Proposed stacked autoencoder architecture.

Figure 7(c) shows the learned kernels by neurons in the first hidden layer of the SAE. It is seen that those neurons are relatively active in extracting some

special representations of the input data. Generally, those learned filters may be some low-level features of the training inputs; for instance, they can be some

edges, as they are extracted by the neurons in the first hidden layer.

4.2. CNN training

In this paper, a convolutional neural network for the hemorrhage identification was employed. This section presents the model architecture and training results of the employed CNN. The network architecture is described briefly in Fig. 8, which shows an example of the working paradigm of our CNN. It shows the three principal layers of a typical convolutional network, starting by feeding a brain image into

input layer and convolving it with a selected kernel to make up a multiple of feature maps (convolution layer), then subsampled by a selected window which also produces some smaller feature maps (Max pooling layer). Finally, all extracted features are fed into the Fully connected layer, which is in this work, a feedforward neural network trained using a stochastic gradient descent, to classify images as hemorrhagic images or not.

Table 3 shows a detailed description of the employed CNN model. As seen, the network comprises of three hidden layers in which each layer contains a convolution layer, batch normalization, ReLU and pooling layer, where ‘conv’ denotes convolution layer, ‘BN’ denotes batch normalization where a batch size of 10 is used to achieve stochastic gradients computations. ‘ReLU’ is the Rectified Linear Units referred to the neurons with linearity, and ‘FC’ denotes Fully connected layer. Note that using ‘ReLU’s in deep convolutional neural networks makes the train several times faster than their equivalents of ‘tanh’ units as stated in [35].

Finally, the classifier layer is a full connected feed-forward network of one hidden layer with 20 neurons, and output layer of 2 neurons considering the number of brain image classes which are hemorrhagic and non-hemorrhagic images.

Table 2
Learning parameters of the AE and SAE

Parameters	Autoencoder	Stacked Autoencoder
Number of training samples	2142	2142
Hidden layer 1- neurons	50	45
Hidden layer 2- neurons	–	35
Activation function	Log-sigmoid	Log-sigmoid
Learning rate for pre-training	0.27	0.15
Learning rate for fine-tuning	0.25	0.3
Iterations for pre-training	65	120
Iterations for fine-tuning	100	400
Training time (s)	58.3	66.8
Mean squared error (MSE)	0.028	0.0021

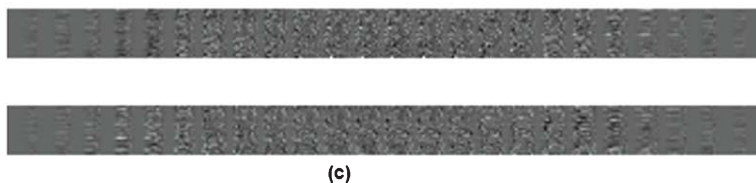
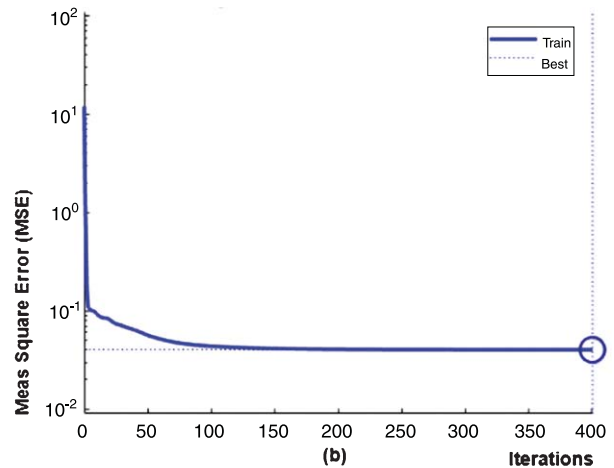
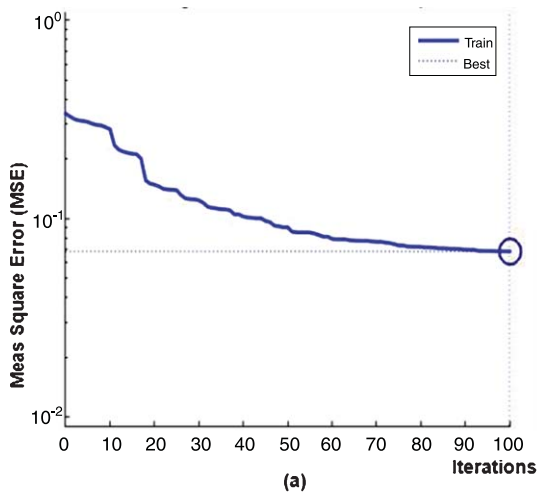


Fig. 7. Training curves and learned kernels of the networks.

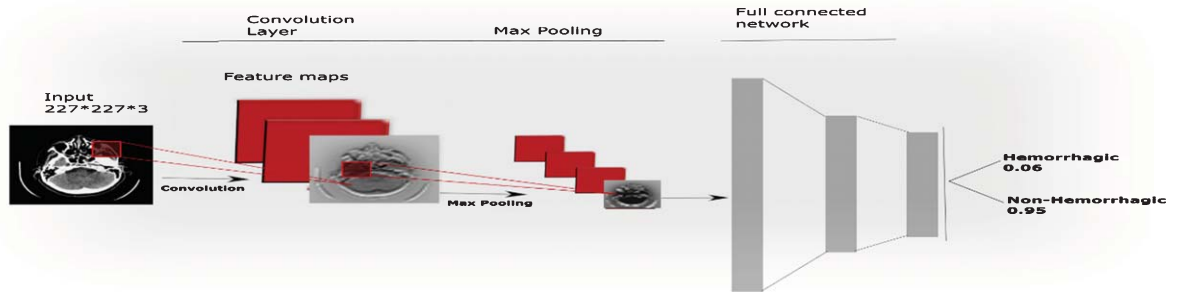


Fig. 8. Proposed Convolutional neural network.

Table 3
CNN parameters

Input	Image Input	227 × 227 × 3 images with 'zerocenter' normalization
Hidden Layer 1	Conv1 + BN+ReLU Pool1	16 feature maps of size 10 × 10 2 × 2 kernel size with stride of 2
Hidden Layer 2	Conv2 + BN+ReLU Pool2	32 feature maps of size 10 × 10 2 × 2 kernel size with stride of 2
Hidden Layer 3	Conv3 + BN+ReLU	64 feature maps of size 10 × 10
Classification layer	FC Softmax	2 fully connected layers 2 units

Table 4
CNN learning parameters

Network	CNN
Training samples	2143
Learning rate	0.8
Iterations	12000
Activation function	Log-Sigmoid
Training time (s)	320
Mean square error (MSE)	0.099

The designed convolutional network was trained on 2143 images containing hemorrhagic and non-hemorrhagic images as shown in Table 1. The input images were of size 227*227 (51529 pixels). Note that the initialization of weights was first at random, while the final training parameters were heuristically obtained during training.

Table 4 shows the training values of the CNN parameters. As seen, the network achieved a mean square error of 0.099, which was used as the cost function of learning, in 320 seconds and 12000 iterations as training time and maximum number of iterations, respectively.

Figure 9 shows the learned filtered at convolution layer 1; it is seen that different levels of features are

extracted. Figure 10 shows the extraction of features of one brain image with hemorrhage in both convolution and pooling layer 1. It shows the different levels of abstractions that can be extracted during each layer as well as during the same layer.

5. Results and discussion

The networks were tested on 385 images containing both output classes. Equation 2 shows the calculation of the performance of the networks, where R is the testing recognition rate of the networks, CC denotes the number of correctly classified samples, and TS denotes the total number of samples or images.

$$R = \frac{CC}{TS} \quad (3)$$

Table 5 shows the recognition rates of each network during training. It can be seen that the autoencoder with one hidden layer achieved lower training recognition rate (93.1%) than that of stacked autoencoder (97.9%) which consists of two hidden layers. However, the higher performance of the SAE required longer training time (66.8 secs) than that of AE (58.3 secs). This may be related to the vanishing gradient, i.e., the depth of the SAE. Furthermore, the table shows the CNN needed 320 seconds to achieve 96.6% recognition rate which is still lower than that of SAE. It is concluded that the larger depth (more number of hidden layers) of the autoencoder can engender better performance in terms of training recognition ratio and error; however, it requires longer time.

Table 6 shows the recognition rates obtained by each network during testing on 385 samples. From this table, it is seen that the SAE performed the best in classifying the brain images into hemorrhagic or not; where it achieved 90.9% recognition rate during

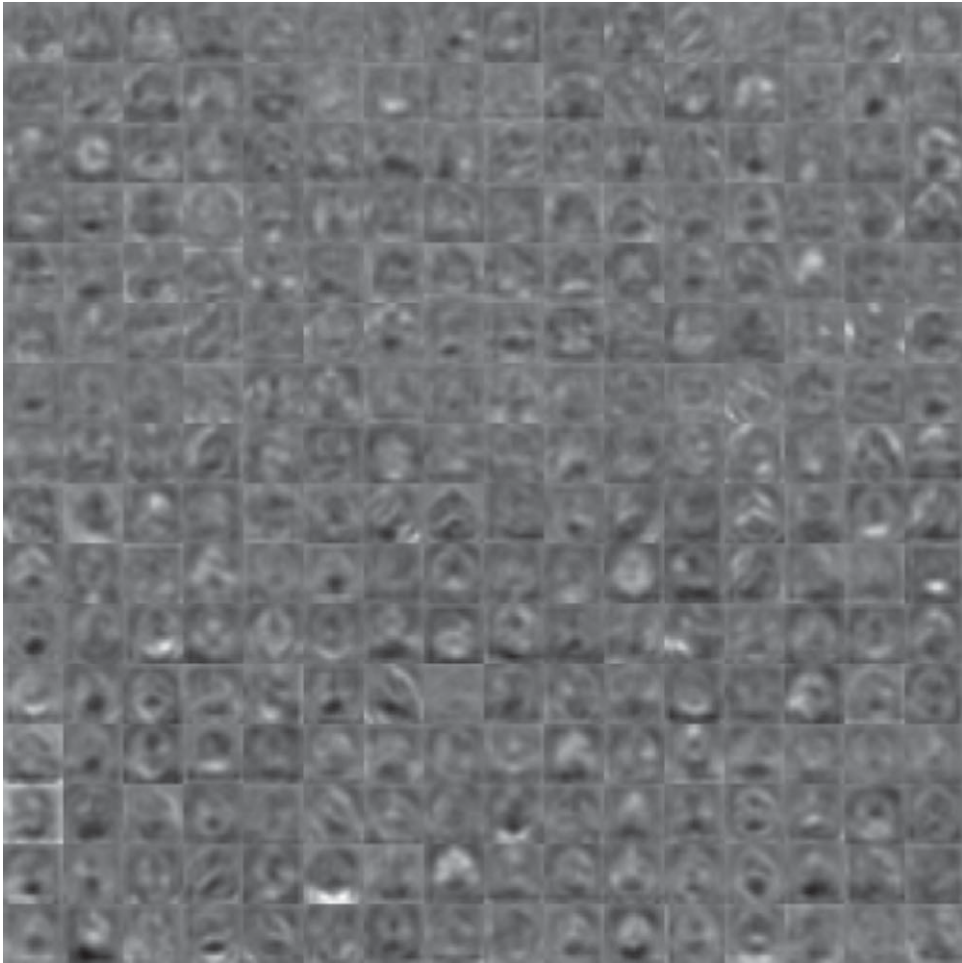


Fig. 9. Learned Kernels at convolution layer 1.

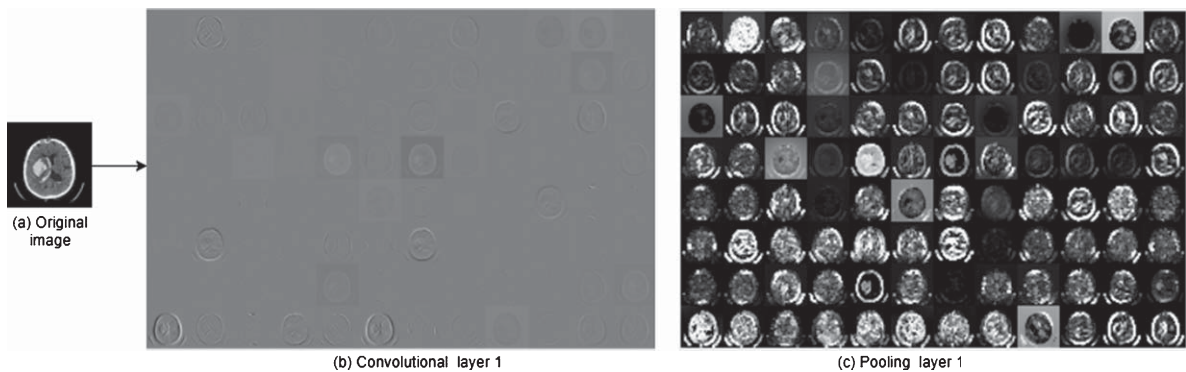


Fig. 10. An example hemorrhagic brain image during convolution and pooling layer.

testing. It is noteworthy to mention that the SAE was expected to perform better than that of AE due to difference in depth of both networks, which allows the SAE to extract more useful features than the AE,

and this result in a better performance. Moreover, SAE has achieved a lower MSE (0.0021) compared to that of AE (0.028); but this required a longer training time (66.8 secs) and more iterations (400). On the

Table 5
Recognition rates of networks during training

Network	AE	SAE	CNN
Number of train images	2142	2142	2142
Number of correctly classified images	1995	2099	2070
Recognition rates	93.1%	97.9%	96.6%
Train time (s)	58.3	66.8	320

contrary, CNN was expected to outperform other used networks, but it didn't. CNN couldn't achieve a lower MSE than that obtained by SAE, although its training time was longer (320 secs) and maximum number of iterations was roughly higher (12000). Thus, it can be concluded that the performance of the three models are relatively closed in terms of correct classification, while the SAE achieved the highest classification rate during training and testing. This outperformance of SAE over CNN is usually due to the small number of data used for training, as stated in many previously published papers [21, 41]. Overall, for a CNN to outperform AE and SAE, it should be trained on a larger database of images in order to learn the different levels of abstractions of different classes. Otherwise, SAE would perform better.

Figure 11 shows some images which were misclassified during the testing phase of the networks. The

Table 6
Recognition rates of networks during testing

Network	AE	SAE	CNN
Number of test images	385	385	385
Number of correctly classified images	340	350	348
Recognition rates	88.3%	90.9%	89.6%

first row (a) shows some normal brain images that were classified as hemorrhagic. Second row (b) shows some hemorrhagic brain images that were classified as non-hemorrhagic.

6. Conclusion

In this research, AE, SAE, and CNN were employed for the classification of brain hemorrhage in CT images. The motivation behind this study is the difficulty that radiologists might encounter when diagnosing brain images; whether they are hemorrhagic or healthy. Thus, we believe that intelligent systems can help medical experts who are most likely exposed to fatigue and eye conditions such as color appreciation deficiency; consequently, this reduces the error of diagnosis and hemorrhage identification made by the medical experts.

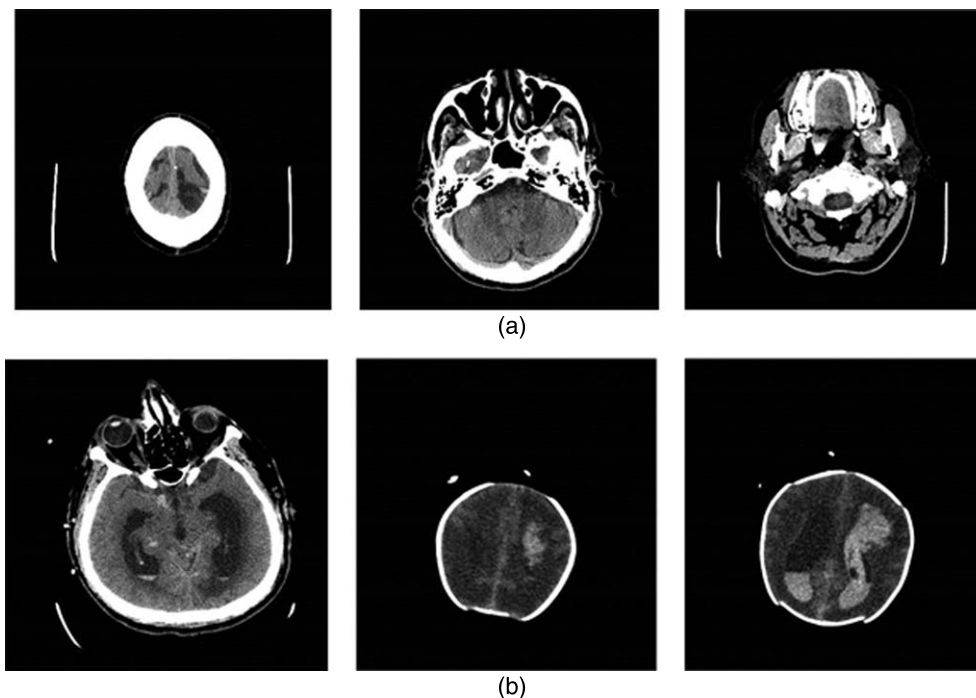


Fig. 11. Sample of misclassified brain images.

All employed networks were trained and tested on same number of images in order to evaluate the performance of each network in addition to investigating the network that outperforms the hemorrhage classification task in terms of recognition rate and minimum square error achieved.

Overall, it is notable that the SAE outperformed other employed networks; where it achieved the highest classification rate and the lowest MSE.

Conflicts of interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

- [1] U. Balasooriya and M.S. Perera, Intelligent brain hemorrhage diagnosis using artificial neural networks. In *Business Engineering and Industrial Applications Colloquium (BEIAC)*, IEEE, 2012, pp. 128–133.
- [2] R. Badenes and F. Bilotta, Neurocritical care for intracranial haemorrhage: A systematic review of recent studies, *BJA: British Journal of Anaesthesia* **115**(Suppl 2) (2015), ii68–ii74.
- [3] L.B. Morgenstern, J.C. Hemphill, C. Anderson, K. Becker, J.P. Broderick, E.S. Connolly and P.H. Mitchell, Guidelines for the management of spontaneous intracerebral hemorrhage. A guideline for healthcare professionals from the American Heart Association/American Stroke Association, *Stroke* (2010).
- [4] F. Aloheid and R.I. Aviv, Emergency imaging of intracerebral haemorrhage, In *New Insights in Intracerebral Hemorrhage*, Karger Publishers, Vol. 37, 2016, pp. 13–26.
- [5] M. Sridevi, S. Naaz and K. Anjum, Diagnosis and classification of brain hemorrhage using CAD system, *Indian Journal of Scientific Research* (2015), 121–126.
- [6] C.M. Chen, Y.H. Chou, N. Tagawa and Y. Do, *Computer-aided detection and diagnosis in medical imaging*, 2013, Computational and mathematical methods in medicine, 2013.
- [7] K. Doi, Current status and future potential of computer-aided diagnosis in medical imaging, *The British Journal of Radiology* **78**(Suppl 2) (2005), s3–s19.
- [8] S. Yu and L. Guan, A CAD system for the automatic detection of clustered microcalcifications in digitized mammogram films, *IEEE Transactions on Medical Imaging* **19**(2) (2000), 115–126.
- [9] S.U. Akram, J. Kannala, L. Eklund and J. Heikkilä, Cell segmentation proposal network for microscopy image analysis, In *International Workshop on Large-Scale Annotation of Biomedical Data and Expert Label Synthesis*, Springer International Publishing, 2016, pp. 21–29.
- [10] A. Akselrod-Ballin, L. Karlinsky, S. Alpert, S. Hasoul, R. Ben-Ari and E. Barkan, A region based convolutional network for tumor detection and classification in breast mammography, In *International Workshop on Large-Scale Annotation of Biomedical Data and Expert Label Synthesis*, Springer International Publishing, 2016, pp. 197–205.
- [11] O.K. Oyedotun, E.O. Olaniyi, A. Helwan and A. Khashman, Hybrid auto encoder network for iris nevus diagnosis considering potential malignancy, In *Advances in Biomedical Engineering (ICABME), 2015 International Conference on*, 2015, pp. 274–277.
- [12] A. Helwan and D. Uzun Ozsahin, Sliding Window Based Machine Learning System for the Left Ventricle Localization in MR Cardiac Images, 2017, *Applied Computational Intelligence and Soft Computing*, 2017.
- [13] S. Albarqouni, C. Baur, F. Achilles, V. Belagiannis, S. Demirci and N. Navab, Aggnet: Deep learning from crowds for mitosis detection in breast cancer histology images, *IEEE Transactions on Medical Imaging* **35**(5) (2016), 1313–1321.
- [14] S.H. Suryawanshi and K.T. Jadhao, Smart brain hemorrhage diagnosis using artificial neural networks, *International Journal of Scientific & Technology Research* **4**(8) (2015), 267–271.
- [15] R. Mahajan and P.M. Mahajan, Survey on diagnosis Of brain hemorrhage by using artificial neural network, *International Journal of Scientific Research Engineering & Technology* **5**(6) (2016), 378–381.
- [16] M. Sridevi, S. Naaz and K. Anjum, Diagnosis and classification of brain hemorrhage using CAD system, *Indian Journal of Scientific Research* (2015), 121–126.
- [17] T. Gong, R. Liu, C.L. Tan, N. Farzad, C.K. Lee, B.C. Pang and Z. Zhang, Classification of CT brain images of head trauma, In *IAPR International Workshop on Pattern Recognition in Bioinformatics*, Springer, Berlin, Heidelberg, 2007, pp. 401–408.
- [18] Y. LeCun, Y. Bengio and G. Hinton, Deep learning, *Nature* **521**(7553) (2015), 436–444.
- [19] J. Schmidhuber, Deep learning in neural networks: An overview, *Neural networks* **61** (2015), 85–117.
- [20] O.K. Oyedotun and A. Khashman, Deep learning in vision-based static hand gesture recognition, *Neural Computing and Applications* **28**(12) (2017), 3941–3951.
- [21] H. Chang, J. Han, C. Zhong, A. Snijders and J.H. Mao, Unsupervised transfer learning via multi-scale convolutional sparse coding for biomedical applications, *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2017).
- [22] G. Carneiro, L. Oakden-Rayner, A.P. Bradley, J. Nascimento and L. Palmer, Automated 5-year mortality prediction using deep learning and radiomics features from chest computed tomography, 2016. arXiv preprint arXiv:1607.00267.
- [23] Y. Anavi, I. Kogan, E. Gelbart, O. Geva and H. Greenspan, A comparative study for chest radiograph image retrieval using binary texture and deep learning classification, In *Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE*, 2015, pp. 2940–2943.
- [24] G.E. Hinton and R.R. Salakhutdinov, Reducing the dimensionality of data with neural networks, *Science* **313**(5786) (2006), 504–507.
- [25] R. Collobert and J. Weston, A unified architecture for natural language processing: Deep neural networks with multitask learning, In *Proceedings of the 25th International Conference on Machine Learning*, 2008, pp. 160–167. ACM.
- [26] A. Mnih and G.E. Hinton, A scalable hierarchical distributed language model, In *Advances in Neural Information Processing Systems*, 2009, pp. 1081–1088.

- [27] R. Hadsell, A. Erkan, P. Sermanet, M. Scoffier, U. Muller and Y. LeCun, Deep belief net learning in a long-range vision system for autonomous off-road driving, In *Intel-ligent Robots and Systems*, 2008. *IROS 2008. IEEE/RSJ International Conference on*, 2008, pp. 628–633.
- [28] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio and P.A. Manzagol, Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion, *Journal of Machine Learning Research* **11**(Dec) (2010), 3371–3408.
- [29] X. Cheng, L. Zhang and Y. Zheng, Deep similarity learning for multimodal medical images, *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization*, 2016, pp. 1–5.
- [30] G.E. Hinton, S. Osindero and Y.W. Teh, A fast learning algorithm for deep belief nets, *Neural Computation* **18**(7) (2006), 1527–1554.
- [31] Y. Bengio, P. Lamblin, D. Popovici and H. Larochelle, Greedy layer-wise training of deep networks, In *Advances in Neural Information Processing Systems*, 2007, pp. 153–160.
- [32] M.R. Avendi, A. Kheradvar and H. Jafarkhani, A combined deep-learning and deformable-model approach to fully automatic segmentation of the left ventricle in cardiac MRI, *Medical Image Analysis* **30** (2016), 108–119.
- [33] A. Alansary, K. Kamnitsas, A. Davidson, R. Khlebnikov, M. Rajchl, C. Malamateniou, . . . and B. Kainz, Fast fully automatic segmentation of the human placenta from motion corrupted MRI, In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer International Publishing, 2016, pp. 589–597.
- [34] A. Rios and R. Kavuluru, Convolutional neural networks for biomedical text classification: Application in indexing biomedical articles, In *Proceedings of the 6th ACM Conference on Bioinformatics, Computational Biology and Health Informatics*, 2015, pp. 258–267. ACM.
- [35] A. Krizhevsky, I. Sutskever and G.E. Hinton, Imagenet classification with deep convolutional neural networks, In *Advances in Neural Information Processing Systems*, 2012, pp. 1097–1105.
- [36] K. Simonyan and A. Zisserman, Very deep convolutional networks for large-scale image recognition, 2014, arXiv preprint arXiv:1409.1556.
- [37] J. Weston, F. Ratle, H. Mobahi and R. Collobert, Deep learning via semi-supervised embedding, In *Neural Networks: Tricks of the Trade*, Springer, Berlin, Heidelberg, 2012, pp. 639–655.
- [38] P. Baldi, Autoencoders, unsupervised learning, and deep architectures, In *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*, 2012, pp. 37–49.
- [39] D. Erhan, Y. Bengio, A. Courville, P.A. Manzagol, P. Vincent and S. Bengio, Why does unsupervised pre-training help deep learning? *Journal of Machine Learning Research* **11**(Feb) (2010), 625–660.
- [40] Near East University Hospital. From <http://neareasthospital.com>
- [41] O.K. Oyedotun and A. Khashman, Prototype-Incorporated Emotional Neural Network, *IEEE Transactions on Neural Networks and Learning Systems*, 2017.