IJICIS, Vol.16 No. 4 ACTOBER 2016

International Journal of Intelligent Computing and Information Science

CLASSIFICATION OF LOW QUALITY IMAGES USING CONVOLUTIONAL NEURAL NETWORK AND DEEP BELIEF NETWORK

E. H. El-Ashmony M. A. El-Dosuky Samir Elmougy

Department of Computer Science, Faculty of Computers and Information,Mansoura University, Mansoura 35516, Egypt
Esraaelashmony16@yahoo.com mouh_sal_010@mans.edu.eg mougy@mans.edu.eg

Abstract: *Low quality images become more challenge and core problem in recent decade because of the ambiguity of contents of them. Convolutional deep neural networks are used for solving this problem. In this work, we used a combination of convolutional neural network and deep belief network to construct an efficient model able to classify low quality images. This model has the capability in extracting effective features from low quality images. Data augmentation is used through this model to increase the accuracy of the system. Scikit-Learn python library is used in implementation the system on STL-10 dataset. The results showed that the proposed model increase the accuracy of the system by 0.20%.*

Keywords*: Convolutional deep neural networks, Deep belief network, Low* quality images.

1. Introduction

Recently, Convolutional Deep Neural Networks (CDNN) has been attracting in many Fields such as image classification and object recognition. It depends on Convolutional Neural Networks (CNN) model that uses shared weights to reduce connections between nodes and provide a large solution space [1]. It consists of neurons with learnable weights and biases. It receives an input (like image in form a single vector) and transforms it to hidden layers (made up of set of neurons) in which all neurons in hidden layer are fully connected with neurons in the previous layer. The last fully-connected layer is called the output layer [2]. CDNN model, unsupervised, and supervised learning are more suitable for training a large amount of data. Image classification is one of the core problems in computer vision. It assigns given image to one label from a fixed set of categories [3].For example, a cat image that has a total of 784 pixels (28 horizontal 28 vertical) shown in Figure 1 can be represented by a number [4]. This number could be 255 for white image view and zero for black image. This means that this image is a vector of 784 dimensions. When applying CNN to this image, it detects that it represents a cat with probability 82%, 15% as a dog, 2% as a hat and 1% as mug and thus its final predicted class is the cat (the highest probability among the detected classes for this image).

Supervised learning is machine learning application is used for inferring a function from labeled data [5].It has two process, training process to train the labeled data by using the training data and testing process to test the target (output layer) by using unlabeled data to know the model accuracy.

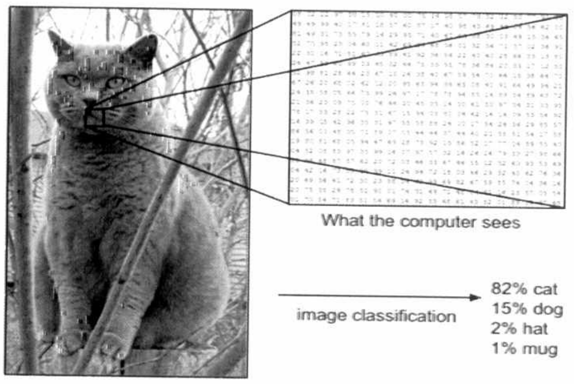

Figure 1: Image classification example [24]

Unsupervised learning is also machine learning application that used for inferring function for hidden structure; the output in unsupervised learning is unknown (no target) [6]. It is used for training unlabeled data and discovering information from data. The learning in unsupervised without teacher and it extracts the clusters to discover similarity between training data [7].

In this work, CNN is applied first for feature extraction and DBN is used next for training. Before extracting features, the input image is preprocessing using data augmentation that has several affects. The first effect is random flip of the image on horizontal and vertical sides, then image is cropped to 80*80 pixels and finally rotation transformation is applied. In our experimentation, STL-10 dataset is used. In Section 2, some previous works closed to our work are discussed. Section 3 presents the proposed work. Section 4 shows the experimentations and its results. Section 5 discusses conclusions and future work.

2. The Related Works

Coates and Ng [18] designed an algorithm for automatically choosing a receptive fields and selecting similar groups of features by using similarity metric. During training, they used unsupervised learning for adjusting weights and the connected layers. For the testing, they used CIFAR and STL-10 datasets and their results showed 60.1 % accuracy. Dundar et al. [19] applied modified *K*-means for minimizing the correlated parameters through training a DCNN. They used labeled data for training and supervised for learning before unsupervised learning for extracting useful features. The 3*96*96 pixelsinput layer is passed to 5*5 filters to obtain the feature maps (96*96*92) and add the Connection matrix between the parts of model to improve the accuracy by 67.1%

Dong and Tan [20] used a single- layer of K-mean networks with multiple layers of unlabeled data and learning features by unsupervised learning with random selecting 20.000 unlabeled image and the size of square pooling is 4*4.It depends on the number of clusters and shallow learning (SIFT-representation instead on deep learning). They overcome the cluttered background of objects in the dataset and improved the classification accuracy by 68.23%. Alexey et al [21] used unlabeled data of some transformations on a set of surrogate classes (samples of image patches) for training raw data images that is represented by data augmentation (translation, scaling, rotation, contrast). They used supervised feature learning for the large data and unsupervised learning for extracting features. The classification accuracy on STL-10 dataset is 72.8%.

3. The Proposed Model

In the previous works, some researchers used CNN alone for image classification, but in this paper, we combine CNN and DBN, It is called CDNN model. All images were trained and tested by DBN and all the parameters of training sent through the RBM. The output data is reshaped as new images andused as input for the CNN [17].

The proposed model is an adaptation of CDNN model. This model based on CNN and DBN .The output of CNN is a regular, fully connected with on-linearity softmax layer, the output provides an estimate of the conditional probability of each class. It depends on local connectivity, parameters sharing and subsampling hidden units. It consists of mainly two stages. The first stage depends on extracting useful features from the input images [10].In the second stage, the training data by using Deep Belief network (DBN) algorithm for STL-10 dataset. The goal of the second stage is to assist and enhance the first stage for solving classification problem. The unsupervised learning is applied to use only the inputs $x^{(t)}$ for learning and automatically extracting meaning features for Raw data. DBN is a generative model that mixes undirected and directed connections between variables. The top two layers' are an RBM and the other layers are a Bayesian network.

Unsupervised training algorithm is applied with the follower steps. First it trains one layer each time starting from first to last layer then it fixes the parameters of previous hidden layers and the previous layers viewed as feature extraction. In the hidden units, the first layer searches for hidden unit's features that is more common in training inputs than in random inputs. In the second layer searches combinations of hidden unit features that are more common than random hidden unit features. The third layer searches combinations of the previous layers.

The main stage in any classification model is extracting features from raw data images as shown in Figure 2. The CNN model consists of an input image of 96*96 pixels and the images in RGB mode that consist of three color maps and the size of input image is 96*96* 3 pixel [11]. Before extracting features, we used the data augmentation for the input image for increasing the size of the dataset where it gives more probability for the same image and reduces the overfitting for images [12]. The first effect is random flip on horizontal and vertical sides then image cropping to 80*80 pixel and rotation transformation for it.

CNN model, as shown in Figure 3, consists of convolutional layer with input pixel x_i associated with the output feature map Y_i by filter k_{ij} computed by equation (1).

$$y_i = b_i + \sum_i k_{ij} * x_i \tag{1}$$

If $l$is the loss function, then for convolutional operationy_i=x_i*k_i, the gradient for x_i is given by equation (2).

$$\nabla_{x_i} l = \sum_j (\nabla_{y_i} l) * (w_{ij}) \tag{2}$$

The gradient for w_{ij} is given by equation (3).

$$\nabla_{w_{ij}} l = (\nabla_{y_i} l) * \tilde{X}_i \tag{3}$$

where $*$ is the convolution withZero padding and $\tilde{X}_i$is the row/column flipped version of x_i.

The follower layer to convolutional layer is subsampling layer for decreasing the size of feature map [21]. In the convolutional layer and subsampling layer, all biases had initial values zeros and the first three convolutional layers were padded with two, these means adding two zero rows to the top and to the bottom ,also adding two columns to the left and the right. The stride for all the convolutional layers have one value and for all subsampling layers have two values [22]. The filter parameters takes 10-4 as an initial value for a specific factor for convolutional layer in the first time, in the second time takes 0.01 by a layer specific factor and for the fully connected layer takes 0.1.

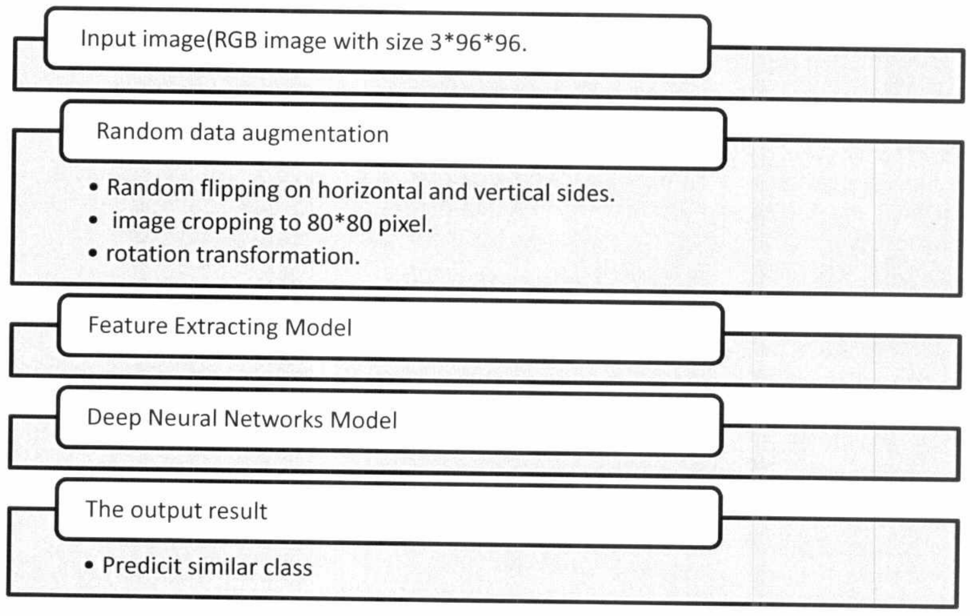
Input image(RGB image with size 3*96*96.
Random data augmentation
• Random flipping on horizontal and vertical sides.
• image cropping to 80*80 pixel.
• rotation transformation.
Feature Extracting Model
Deep Neural Networks Model
The output result
• Predicit similar class

Figure 2: The main model for classification of STL-10 dataset.

The number of features effects in the visibility of the output image, as shown in Figure 5.a where the dog image after 22nd features map is a humble result, while Figure 5.b shows the same image after 45th feature map which becomes more visible than Figure 5.a.

We used for the classification problem unsupervised learning for inferring function for hidden structure where the output in unsupervised learning is unknown(no target) and the training data is unlabeled [10]. The learning in unsupervised without teacher. It creates the clusters by discovers similarity between training data. In this stage we trained data by using DBN.

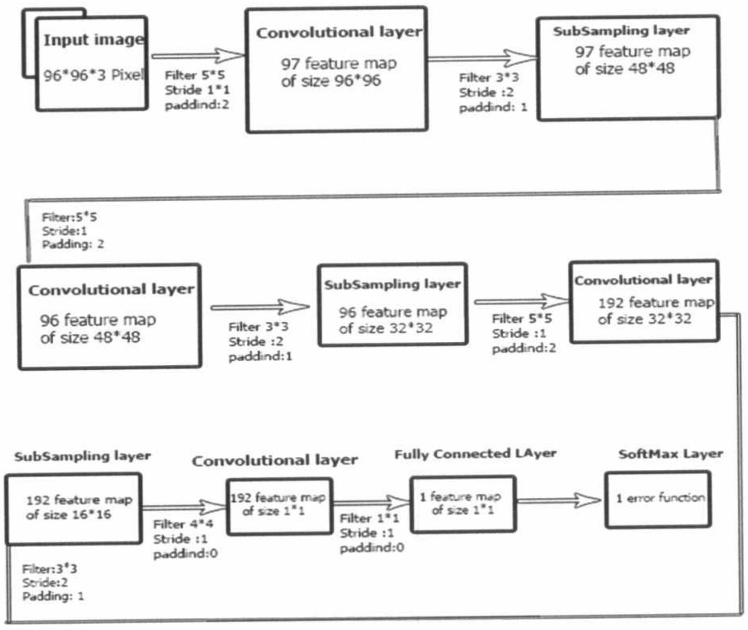

Figure 3: CNN model for extracting useful features from input image.

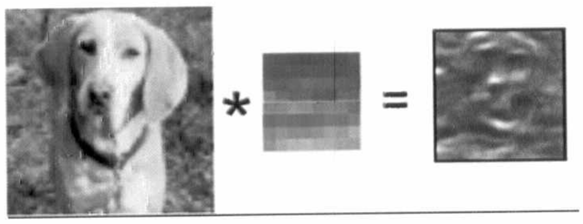

Figure 4.a: Dog image after 22nd features map.

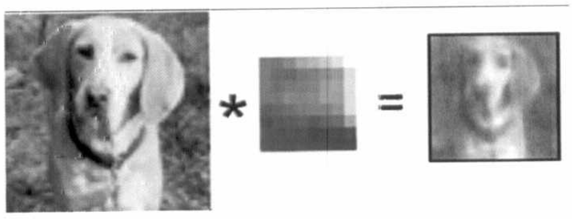

Figure 4.b: Dog image after 45th feature map

DBN is a directed acyclic graph contains with multi-layers from hidden layers and it uses non-linear operation in output [14].It uses unsupervised approach in learning process. It is composed of stacked of

Restricted Boltzmann Machine (RBM) as shown in Figure 5. It can learn deep net by using millions of parameters and RBM for each layer. It isa generative stochasticthat deal with labeled and unlabeled images [12].The important for using RBM is learning process Collaborate with DBN, equation (4) represents the relation between visible and hidden units.

$$E(v,h) = -\sum_i \sum_j w_{ij} h_i v_j - \sum_j b_i v_i - \sum_i c_i h_i \quad (4)$$

wherew_{ij} : weights of visible and hidden units,v_i: binary vector of visible units,h_i: binary vector of visible units,c_i: bias weights for the visible units and b_i:bias weights for the hidden unit.

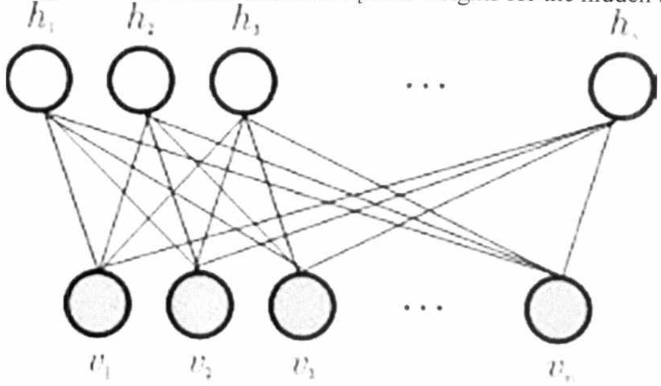

Figure 5: RBM simple mechanism [12]

The probability of an array in RBM follows the Boltzmann distribution, the probability for any array v and $h$is given by equations (5) and equation(6).

$$p(v,h) = \frac{1}{Z} e^{-E(v,h)} \quad (5)$$

$$p(v,h) = \sum_{v,h} e^{-E(v,h)} \quad (6)$$

Where Z is the partitionfunction, let generalized (normalization factor). Now that the above joint distribution is defined [16]. Hence, the conditional probability distribution and the opposite of the situation for a given v, h is given by equation (7) andequation (8).

$$p(h|v) = \prod_i p(h_i|v) \quad (7)$$

$$p(h|v) = \prod_j p(v_j|h) \quad (8)$$

The distribution of a specific node in the other layer is given by equation (9) and equation (10).

$$P(h_i = 1|v) = \sigma(\sum_j w_{ij} v_j + c_i) \quad (9)$$

$$P(v_j = 1|h) = \sigma(\sum_i w_{ij}h_j + b_j) \quad (10)$$

4. Experimentations and Results

STL-10 is an object recognition dataset used for by researchers to aim to improve the image classification problem and develop unsupervised learning. It has size 96*96 pixels (RGB image) and it has the same number and types of classes as CIFAR-10 dataset. It consists of labeled and unlabeled images and has 10 classes (airplane, bird, car, cat, deer, dog, horse, monkey, ship, truck). Also, it composed of two sections, one for training (500 images*10 classes) and other for testing (800 images * 10 classes)as shown as in Figure 2 [23].

Figure 6: STL-10 Dataset samples [23]

This model is implemented using Scikit-learn machine learning tool [18]. It is an open source, simple and efficient library for solving the machine learning problems. The proposed model depends on creating data augmentation for adding more shapes to the same image and reducing the overfitting as shown in Figure 7 then we used the CDNN model. It consists of two stages. First, CNN architecture model is applied for extracting extra useful features and for filter learning as shown in Figure 8. The second stage isfor training data by DBN, equation (11) is used to measure the accuracy of the proposed model.An example of applying the convolutional features of car image is shown in Figure9.The fine tuning is used for adjusting weights and improving the performance of all layers .After unsupervised learning by using supervised gradient descent of the cost function on the output based the last hidden layer for classify the input image.

$$Accuracy = \frac{Number\ of\ correct\ classification}{Total\ number\ of\ test\ cases} \quad (11)$$

The proposed system is compared with some other related systems as shown in Table (1) and Figure 10 in which the results show that CDNN and unlabeled data give better accuracy than shallow learning and labelled data that take a long time for training. The output from the first stage take as input for the next stage that is consider more effective for enhancement the accuracy about the raw data in the past.

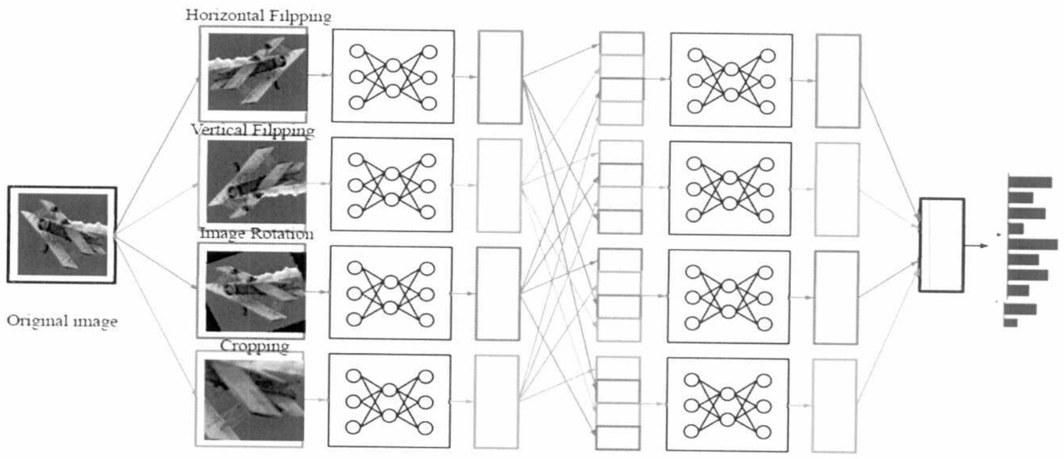

Figure 7: Architecture for data augmentation

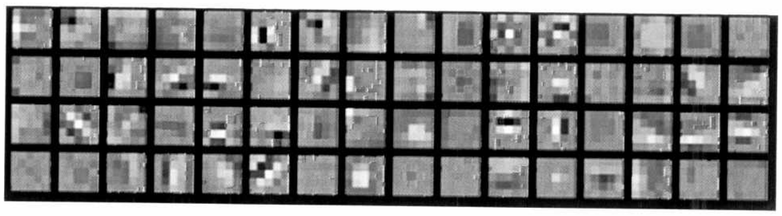

Figure 8: The learned filters for the first training layer

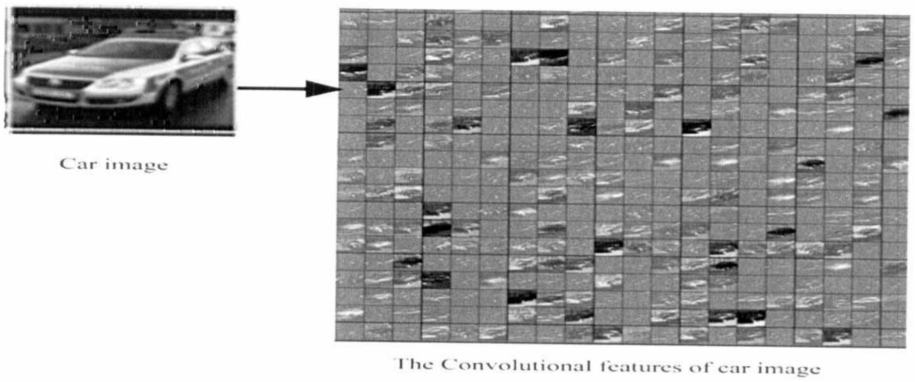

Figure 9: the output of Convolutional features of Car image

Table (1): The Comparison between some other systems and the proposed system

Algorithm	Accuracy
Selecting Receptive Fields in Deep Networks(A)	60.10 %
Convolutionalclusteringfor unsupervised learning(B)	67.10 %
Unsupervised Feature Learning with C-SVDDNet(C)	68.23 %
Discriminative Unsupervised Feature Learning with Convolutional Neural Networks(D)	72.80 %
Multi-stage CDNN(our model)(E)	73.00 %

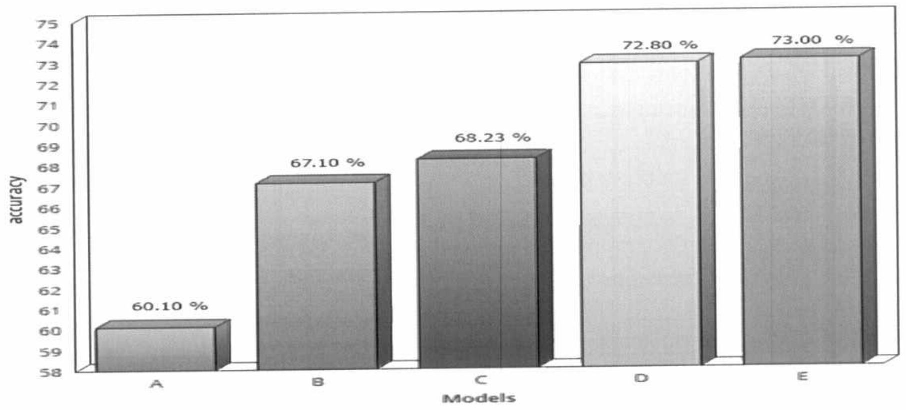

Figure 10: Comparison between the proposed system accuracy and some other previous systems

4. Conclusion

In this paper, we applied a model for classification of low quality images depending on DCNN architecture and unsupervised learning. It consists of two main stages: the first stage is for extracting features and the second stage is for training data by deep belief network. Scikit-Learn machine learning library was useful for implementation. The total accuracy for the proposed model gives 73% in which it is better than the other compared related models. As future works, we hope to increase and improve the accuracy and apply this model on other datasets that is more complex.

References

1. You Zhining and Pu Yunming, "The Genetic Convolutional Neural Network Model Based on Random Sample", International Journal of u-and e-Service, Science and Technology, Vol.8, No. 11, pp.317-326, 2015.
2. Igor Aizenberg and Claudio Moraga, "Multilayer Feedforward Neural Network Based on Multi-valued Neurons (MLMVN) and a Backpropagation Learning Algorithm", Soft Computing, Vol. 11, No. 2, pp. 169-183, 2011.
3. https://en.wikipedia.org/wiki/Pixel, Last Access: 20/4/2016.
4. Shi Zhong, "Semi-supervised Sequence Classification with HMMs", International Journal of Pattern Recognition and Artificial Intelligence (IJPRAI), Vol. 19, No. 2, pp. 165-182, 2005.

5. James T. Kwok, Zhi-Hua Zhou, Lei Xu. Machine Learning. Springer Handbook of Computational Intelligence. J. Kacprzyk and W. Pedrycz. Berlin, Heidelberg, Springer Berlin Heidelberg: 495-522, 2015.

6. http://www.wow.com/wiki/Deep_Learning, Last Access: 3/3/2016.

7. Geoffrey E. Hinton, Simon Osindero, Yee-Whye, "A fast learning algorithm for deep belief nets", Journal of Neural Computations, MIT, Vol. 18, No. 7, pp. 1527 – 1554, 2006.

8. Bengio, Yoshua, Aaron Courville, and Pierre Vincent. "Representation learning: A review and new perspectives", IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 1798-1828, 2013.

9. https://www.techopedia.com/definition/28033/data-augmentation/, Last Access: 6/1/2016.

10 .http://danielnouri.org/notes/category/python/, Last Access: 14/3/2016.

11. http://cs231n.github.io/neural-networks-1/, Last Access: 4/2/2016.

12. Lopes, Noel, and Bernardete Ribeiro. "Towards adaptive learning with improved convergence of deep belief networks on graphics processing units." Pattern Recognition, pp. 114-127, 2014.

13. Hinton, Geoffrey E. "Learning to Represent Visual Input" Philosophical Transactions of the Royal Society of London B: Biological Sciences 365.153, pp. 177-184, 2010.

14. Hinton, G. E. A Practical Guide to Training Restricted Boltzmann Machines. Neural Networks: Tricks of the Trade: Second Edition. G. Montavon, G. B. Orr and K.-R. Müller. Berlin, Heidelberg, Springer Berlin Heidelberg, pp. 599-619, 2012.

15 .Lichun Fan, Hongyan Li, Dengfeng Ke, Bo Xu, "A Novel Noise-Robust ASR Method by Applying Partially Connected DNN Model and Mixed-Bandwidth Concept", 2nd International Symposium on Computer, Communication, Control and Automation (3CA 2013), pp. 182-185, Singapore, December 1-2, 2013.

16. http://scikit-learn.org/stable/tutorial/: Last Access 4/1/2016.

17. Adam Coates and Andrew Y. Ng, "Selecting Receptive Fields in Deep Networks", Advances in Neural Information Processing Systems, pp.2528-2536, 2011.

18 Aysegul Dundar, Jonghoon Jin, Eugenio Culurciello, "Convolutional Clustering for Unsupervised Learning",

19. Dong Wang, Xiaoyang Tan, "Unsupervised Feature Learning with C-SVDDNet", https://arxiv.org/pdf/1412.7259.pdf, Last Access: 15-4-2016.a

20. Alexey Dosovitskiy, Jost T. Springenberg, Martin Riedmiller and Thomas Brox, "Discriminative Unsupervised Feature Learning with Convolutional Neural Networks", Advances in Neural Information Processing Systems, pp.766-774, 2014.

21. Dan Claudiu Ciresan,Ueli Meier,Jonathan Masci,Luca Maria Gambardella Luca Maria Gambardella, Jürgen Schmidhuber, "Flexible, High Performance Convolutional Neural Networks for Image Classification", Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Volume 2, Barcelona, Catalonia, Spain, pp. 1237-1242 , July 16-22, 2011.

22 .http://deeplearning.net/software/theano/library/tensor/nnet/conv.html, Last Access: 2/3/2016.

23. https://cs.stanford.edu/~acoates/stl10/, Last Access: 1/5/2016.

24. http://cs231n.github.io/classification/, Last Access: 6/4/2016.