Deep Convolution Neural Networks for Image Classification

Arun D. Kulkarni
Computer Science Department
The University of Texas at Tyler
Tyer, TX 75799, USA

***Abstract*—Deep learning is a highly active area of research in machine learning community. Deep Convolutional Neural Networks (DCNNs) present a machine learning tool that enables the computer to learn from image samples and extract internal representations or properties underlying grouping or categories of the images. DCNNs have been used successfully for image classification, object recognition, image segmentation, and image retrieval tasks. DCNN models such as Alex Net, VGG Net, and Google Net have been used to classify large dataset having millions of images into thousand classes. In this paper, we present a brief review of DCNNs and results of our experiment. We have implemented Alex Net on Dell Pentium processor using MATLAB deep learning toolbox. We have classified three image datasets. The first dataset contains four hundred images of two types of animals that was classified with 99.1 percent accuracy. The second dataset contains four thousand images of five types of flowers that was classified with 86.64 percent accuracy. In the first and second dataset seventy percent randomly chosen samples from each class were used for training. The third dataset contains forty images of stained pleura tissues from rat-lungs are classified into two classes with 75 percent accuracy. In this data set eighty percent randomly chosen samples were used in training the model.**

***Keywords*—Deep learning; convolutional neural networks; image classification; machine learning; object recognition**

I. Introduction

Image classification can be defined as categorizing images into predefined classes. Traditionally, image classification is conducted in two stages- low-level processing and high-level processing or recognition. Low level processing deals with image enhancement, filtering, detecting regions of interest, and extracting feature descriptors. High-level processing deals with classification, where feature descriptors are used to train the classifier into predefined categories. The two stages are often implemented sequentially. First, feature descriptors are obtained and subsequently are classified. The main disadvantage of this approach is that accuracy of the classifier is dependent on the design of the feature extraction stage. Many machine learning algorithms such as decision trees, Support Vector Machine (SVM), neural networks have been used to classify feature vectors obtained from images. Machine learning algorithms are known to learn the underlying relationship in data and make decisions. Deep Convolution Neural Networks (DCNNs) are one of the best learning algorithms for understanding image content and have shown exemplary performance in image segmentation, detection, and retrieval tasks [1]. In the recent years DCNNs are preferred for image classification. DCNNs use multiple layers consisting of nonlinear information processing units for low-level as well as high-level processing. DCNNs are feedforward networks. In general, DCNNs consist of convolution and pooling layers that are grouped into modules followed by one or more fully connected layers. Convolutional layers are used for extracting features from the input image. In a convolution layer, inputs are convolved with a weighted kernel and the output is sent via a nonlinear activation function to the next layer. The purpose of pooling layers is to reduce spatial resolution of the features maps and achieve spatial invariance to input distortions and translations [2]. In earlier days, the average pooling was used, but recently the max pooling has become a common practice. Several convolution and pooling layers are stacked together. The fully connected layers are used for classification. DCNNs were successfully used to classify images in the ImageNet Large Scale Visual Recognition Challenge [3]. Recent developments in DCNNs were possible because of two main factors a) availability of faster computing resources such as Graphical Processing Units (GPUs) and b) availability of large, labeled image datasets Also, there were algorithmic improvements in Deep Neural Networks (DNNs). DCNNs commonly use the gradient decent backpropagation algorithm. There are some drawbacks with DCNNs. The first drawback is the use of Sigmoid activation functions that leads to saturation resulting into slow convergence of gradient descent. The problem becomes more severe as we move away from the output layer to hidden layers. The compounded effect of saturation at multiple layers is known as vanishing gradient [4]. In the backpropagation algorithm the mean squared error at the output layer is propagated backwards to the hidden layers to calculate the change in weights. To avoid the vanishing gradient problem, recent DCNNs use the entropy loss function with Rectified Linear Units (ReLU) in the output layer. The second drawback with DCNNs is overfitting that occurs due to the substantial number of parameters that are updated in learning. Overfitting usually occurs when the dataset is of the small size. Various regularization techniques such as the dropout or bagging are used to overcome this problem. The third drawback is due to the non-convex shape of the error function. The backpropagation algorithm is sensitive to the randomly chosen initial values of weights. The gradient descent algorithm may get stuck at a local minimum. To avoid this problem the model is initially trained with a few parameters and then more parameters can be added during the training. DCNN models trained with a large dataset and can classify images with high accuracy. Many architectures for DCNN models have been used for image classification. In this paper we implement Alex Net using MATLAB deep learning toolbox. We use the model to classify three image datasets a) animal b) flower, and c) stained pleura tissue images. The outline of the paper is as follows. Section II describes related work. Section III deals with implementation Alex Net and results, and Section IV provides conclusions.

II. Related Work

Neural networks (NN) are biologically inspired and are used for object recognition, image classification. Neural networks have been used as associative memory to store and retrieve information. Associative memories function as content addressable memories. Neural networks learn from training samples and have been used for pattern recognition since the 1950s [5]. Feedforward networks with a backpropagation learning algorithm have been used as supervised classifiers [6]. Today several well-developed learning algorithms for multi-layer neural network models are available. These include a multi-layer Perceptron, feedforward networks with back-propagation learning, Boltzman machines, Hamming net, Hopfield net, neocognitron models [7, 8, 9, 10, 11] Huang and Lippmann [12] provide a comparative study of neural networks and conventional classifiers. Neural networks have been used to implement expert systems or knowledge-based system. The success of neural networks has led to deep neural networks (DNNs). Deep learning algorithms were available since late 1980s, However, DNNs were computationally expensive. Chellaplla [13] suggested using Graphical Processing Units (GPUs) to implement deep learning algorithms faster. Deep learning is a form of machine learning that enables computers to learn from experience and understand the world in terms of hierarchy of concepts [14]. DNNs are neural networks with multiple hidden layers. The multiple hidden processing layers has dramatically improved the state-of-the art visual object recognition. DNNs discover intricate structures in large datasets by using the backpropagation algorithm [15]. Convolutional neural networks (CNN) are special type of neural networks for processing data that have a known grid-like structure [16]. Various stages of a DCNN show topology resemblance to primate's ventral pathway of visual cortex [17]. DCNNs can learn internal representations from raw pixels. DCNNs are hierarchical learning models and can extract features [18,19,20]. Rawat and Wang [21] present a comprehensive review of DCNNs. DCNNs consist of a stack of convolution and pooling layers followed by fully connected layers. Convolution and pooling layers are used for features extraction. The max-pooling method is a wieldy accepted method in recent DCNNs. Scherer et al [22] have shown that the max-pooling method can capture the invariance and is effective in reducing the computational time. DCNNs have been used in many computer applications that include image and object classification, face detection, image segmentation, and gesture recognition. Recent DCNNs have ten layers of ReLU, hundreds of millions of weights, and billions of connections between units [15]. Machine learning community's interest in DCNNs grew after Image Net compaction in 2012, where Alex Net achieved record breaking results in classifying images from ILSVRC data set consisting of more than 1.2 million images in to one thousand classes. This was a landmark achievement that has revolutionized the computer vision field. Significant achievements in DCNNs are a) LeCun et al. [2] used a DCNN to classify 70,000 hadwritten images of digits in to ten classes. b) Fei-Fei et al. [23] used a DCNN to classify 9,146 color images from CALTECH-101 data set in to 101 classes. c) Krizhevsky [24] classified 60,000 images in CIFAR-100 data set into one hundred classes. d) Russakovsky et al [25] classified more than 1.2 million images from ILSVRC dataset into one thousand classes. LeNet was proposed by LeCun et al. [26]. Alex Net proposed by Krizevsky et al [3] was based on principles used in LeNet Simonyan et al. [27] proposed a DCNN model VGG Net that was made nineteen layers deep and used 3x3 filters. The use of small size filters could induce the effect of large size filters and provided computational simplicity by reducing the number of parameters. Nowadays, most new DCNN architectures are built upon the principle of simple and homogenous topology as introduced in VGG Net. Zhang et al, [28] provide the taxonomy of CNNs. Khan et al. [29] discuss intrinsic taxonomy present in the recent and prominent DCNN architectures reported from 2012-2020. They have classified DCNN architectures into seven categories, namely, spatial exploitation, depth, multi-path, width, feature-map exploitation, channel boosting, and attention-based. Stacking of multiple transformations deep and in parallel fashion showed good learning for complex problems [30, 31]. Google Net was the winner of the 2014-ILSVRC competition [32]. Google Net introduced the concept of inception block, which incorporates multi-scale convolutional transforms using split, transform, and merge idea. The textbook by Szelinski [33] describes deep learning techniques including deep feedforward networks, regularization, optimization algorithms, convolutional neural networks. We have implemented Alex Net using MATLAB deep learning toolbox and have analyzed three datasets.

III. Implementation and Results

The simplified architecture of Alex Net is shown in Fig. 1. It contains eight layers: five convolution and three fully connected layers. Convolution layers serve as feature extractors. Inputs are convolved with learned weights to compute feature maps and results are sent through a nonlinear activation function. The output of the *k*th feature map Y_k is given by (1).

$$Y_k = f\left(W_k * x\right) \tag{1}$$

Where x denotes the input image, W_k is the convolution filter. The '*' sign refers to the 2D convolution operator. The purpose of the pooling layer is to reduce the spatial resolution and extract invariant features [21]. The output of a pooling layer is given by (2).

$$Y_{kij} = \max_{(p,q)\in R_{ij}}\left(x_{kpq}\right) \tag{2}$$

Where X_{kpq} denotes elements at location (p, q) contained by the pooling region R_{ij}.

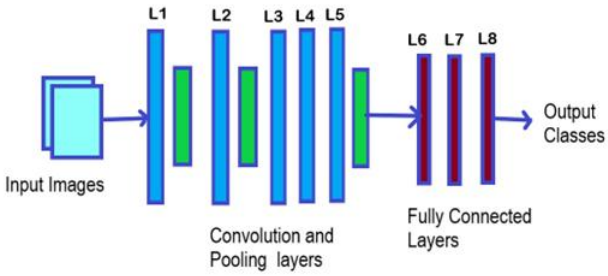

Fig. 1. Alex Net Architecture.

Fig. 2. Illustrates the max pooling operation. Inputs are mapped from a convolution layer to the pooling layer. With a 4x4 mask, the maximum value in each 2x2 sub-area is mapped. The fully connected layers follow the convolution layers that interpret extracted features and perform high level reasoning. DCNNs use learning algorithms to adjust the free parameters in the network to obtain the desired output. The most common algorithm is the backpropagation learning algorithm. The commonly experienced problem with DCNNs is overfitting. This is due to the substantial number of free parameters that are adjusted during learning. The layers in the Alex Net are shown in Table I.

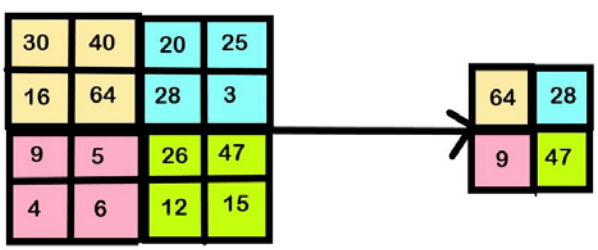

Fig. 2. Max Pooling.

TABLE I. LAYERS OF ALEX NET

	Input image	224x224x3 (Channels)
L1	Convolution Layer with a Pooling Layer	96 kernels of size 11x11x3
L2	Convolution Layer with a Pooling Layer	256 kernels of size 5x5x48
L3	Convolution Layer	384 kernels of size 3x3x256
L4	Convolution Layer	384 kernels of size 3x3x192
L5	Convolution Layer with a Pooling Layer	256 kernels of size 3x3x192
L6	Fully Connected Layer	4096 neurons
L7	Fully Connected Layer	4096 neurons
L8	Fully Connected Layer	4096 neurons

We implemented Alex Net using MATLAB deep learning toolbox. We analyzed three datasets. There are described below:

Example 1: In this example we have considered the subset of animal dataset [34]. The original dataset has 37 categories with 200 samples in each category. For this example, we have selected only two categories. The dataset consists of four hundred images of cats and dogs that are obtained with distinct backgrounds and with a variety of sizes and positions of these animals. We used seventy percent of randomly picked samples from each class to train Alex Net and thirty percent of samples were used for validation. Fig. 3 shows some randomly picked images from the dataset. Images in the dataset we resized to 224 rows x 224 columns. There were two units in the fully connected output layer. The DCNN was able to classify the dataset with 99.9 percent accuracy. Fig. 4 shows the graph for the accuracy and the loss function with iterations. Fig. 5 shows a few classified randomly chosen images with labels.

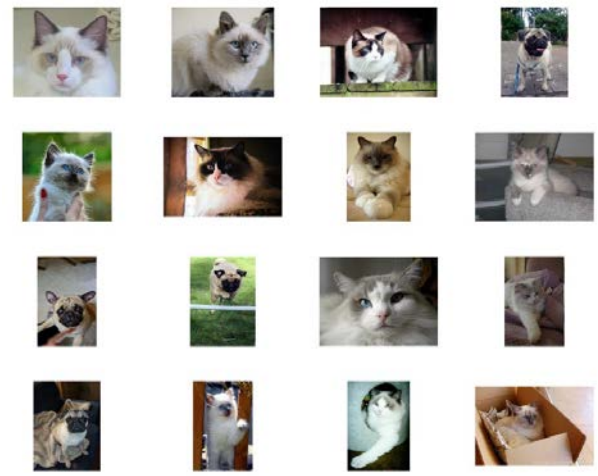

Fig. 3. Sample Images from Animal Dataset.

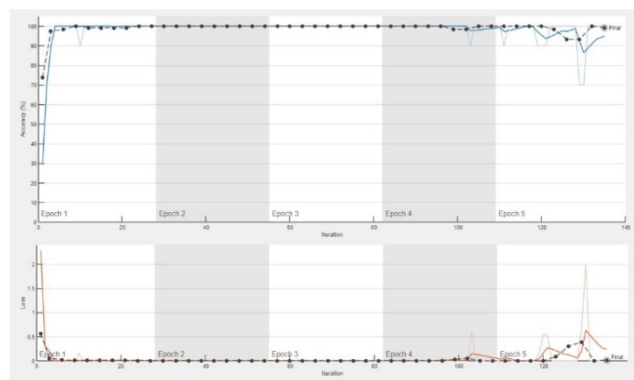

Fig. 4. Accuracy and Loss Function for Animal Dataset.

Fig. 5. Classified Sample Images with Class Labels.

Example 2: In this example we have considered a subset of the flower dataset [35]. The subset consists of four thousand images of five types of flowers: daisy, rose, dandelion, sunflower, and tulip. The dataset contains images that are obtained with distinct color background and with a variety of sizes and colors of flowers. We used seventy percent of randomly picked samples from each class to train Alex Net and thirty percent of samples were used for validation. Fig. 6 shows some randomly picked images from the dataset. Images in the dataset were resized to 224 rows x 224 columns. There were five units in the fully connected output layer. The network took 256 minutes for training on Dell Pentium processor. The DCNN was able to classify the dataset with 86.6 percent accuracy. Fig. 7 shows the graph for the accuracy and the loss function with iterations. Fig. 8 shows a few classified images with labels.

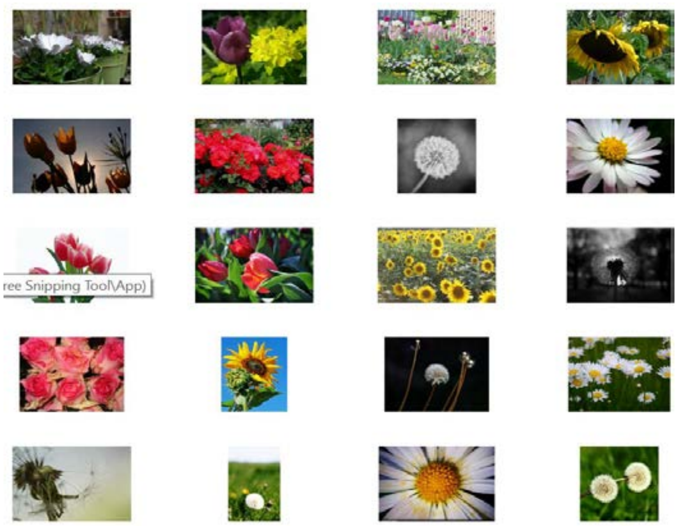

Fig. 6. Sample Images from Flower Dataset.

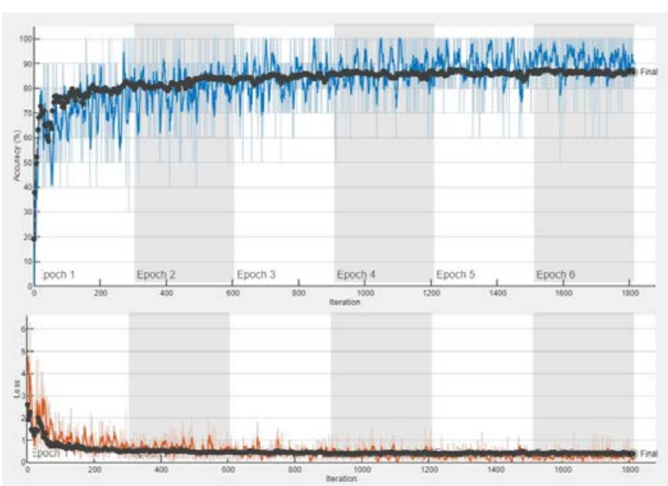

Fig. 7. Accuracy and Loss Functions for Flower Dataset.

Fig. 8. Classified Output with Class Labels.

Example 3: In this example we have considered the stained pleura images of rat-lungs. These are pathological images representing positive and negative cancer cases. The dataset consists of forty images, twenty images in each class. We used eighty percent of randomly picked samples from each class to train Alex Net and twenty percent of samples were used for validation. Fig. 9 shows some randomly picked images from the dataset. Images in the dataset we resized to 224 rows x 224 columns. There were two units in the fully connected output layer. The DCNN was able to classify the dataset with 75 percent accuracy. Fig. 10 shows the graph for the accuracy and the loss function with iterations. Fig. 11 shows a few classified images with labels.

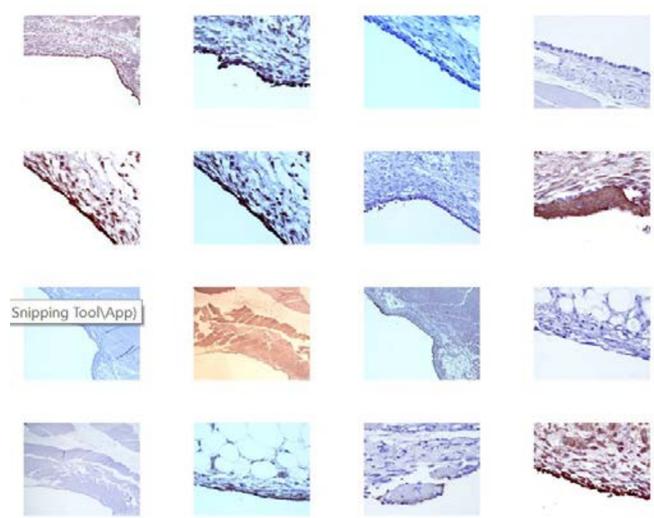

Fig. 9. Sample Images Pleura Dataset.

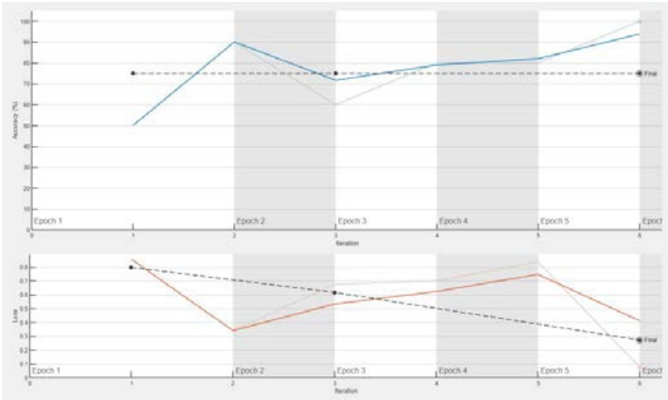

Fig. 10. Accuracy and Loss Functions Pleura Dataset.

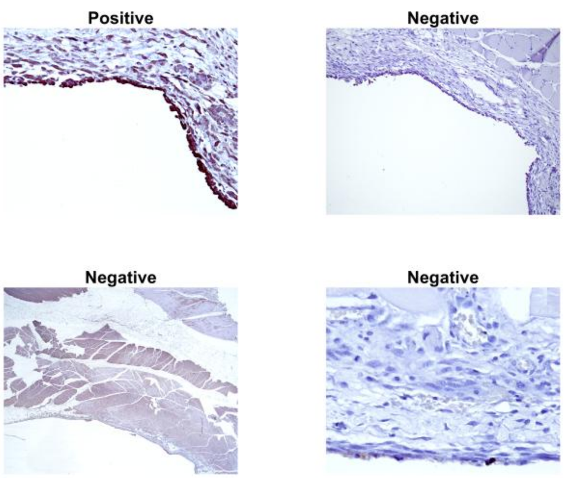

Fig. 11. Classified Output with Labels -Pleura Dataset.

IV. Conclusion

In this research work, we have implemented Alex Net using MATLAB deep learning toolbox and have analyzed three datasets. The classification accuracy for the animal dataset was 99.01 percent. The classification accuracy for the flower dataset was 86.64 percent. Images of several types of flowers were obtained with various backgrounds, sizes, colors. In the flower dataset, many images in two categories sunflower and dandelion are similar. It is hard to distinguish between the two classes as these classes could be non-separable or overlapping classes in the feature space. In the case of pleura image classification, we obtained the classification accuracy of 75 percent. This because the sample size for the dataset was too small. The dataset contained only forty images, out of which 80 percent were used for training. Due to the substantial number of free parameters in the model and the small sample size of the training set, there is possibility of model overfitting. This may result in less accuracy. Also, training and testing images were selected randomly and sometimes we got 100 percent accuracy. We need a large dataset to evaluate this application. The DCNN presented in the paper can be used for other practical applications such as object recognition, military reconnaissance, medical image diagnosis, etc.

Many DCNN models are proposed in the literature. These models vary in the depth, width, activation functions, kernel size, and hardware implementation. It is possible to add layers to a DCNN to extract transformed domain features with the ring- and wedge-shaped filters to extract texture as well as scale and rotation invariant features. Also, it is not yet clear how image features or properties of image categories are represented in the DCNN models. In the case of neural networks, researchers have been able to extract knowledge in term of classification rules by tracing links in the neural net. However, in the case of DCNNs, due to the large number of free parameters, researchers have not been yet able to decode the DCNN models or extract the knowledge as how DCNNs make decision. It is an open area for the future research.

Acknowledgment

The author is thankful to UT Health North, Tyler, TX for providing pleura image dataset.

References

[1] D. Cireşan, U. Meier, J. Schmidhuber, "Multi-column Deep Neural Networks for Image Classification". Computer Vision and Pattern Recognition, 2012, pp. 3642-3649 https://doi.org/10.48550/arXiv.1202.2745.

[2] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner "Gradient-based learning applied to document recognition". Proceedings of the IEEE, 1998, vol. 86, no. 11, pp. 2278–2324.

[3] A. Krizhevsky, I. Sutskever, G. Hinton, "ImageNet classification with deep convolutional neural networks". Adv Neural Inf Process Syst. 2012 https://doi.org/10.1061/(ASCE)GT.1943-5606.0001284.

[4] M. Tan and Q. V. Le, "Efficient Net: Rethinking Model Scaling for Convolutional Neural Networks," Proceedings of the 36 th International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019.

[5] F. Rosenblatt, F, "The Perceptron: a probabilistic model for information storage and organization in brain," Psychology Review, 1958, vol. 65, pp386-408.G. Eason, B. Noble, and I. N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955. *(references).*

[6] A. D. Kulkarni, Computer vision and fuzzy-neural systems. Prentice Hall PTR, Upper Saddle River, NJ, 2001.

[7] J. J. Hopfield, "Neurons with graded response have collective computational properties like those of two state neurons," Proceedings of the National Academy of Sciences, 1984, vol 81, pp. 3088-3092.

[8] D. E. Rumelhart, J. L. McClelland, and the PDP Group, Parallel Distributed Processing, vol I. MIT Press, Cambridge, MA, 1986.

[9] R. P. Lippmann, "An introduction to computing with neural nets," IEEE Transactions on Acoustics, Speech, and Signal Processing, 1987, vol. 32, pp. 4-22.

[10] K. Fukushima, "Neural networks for visual pattern recognition," Computer, March 1988, pp. 65-74.

[11] R. Hecht-Nielsen, Neurocomputing. Addison-Wesley, Reading, MA, 1990.

[12] W. Y. Huang and R. P. Lippmann, R. P, "Comparison between neural net and conventional classifiers," Proceedings of Joint International Conference on Neural Networks, San Diego, 1988, vol. IV, pp. 485-493.

[13] K. Chellapilla, S. Puri, and P. Simard, "High performance convolutional neural networks for document processing," In Tenth international workshop on frontiers in handwriting recognition. Suvisoft. October 2006, pp. 1-6.

[14] K. G. Kim, "Deep Learning Book Review", Healthcare Information Research, 2016, vol. 22, no. 4. pp 351-354. http://doi.org/10.4258/hir.2016.22.4.351.

[15] Y. LeCun, Y. Bengin, and G. Hinton, "Deep learning", Nature, 2015, vol. 521, pp. 436-444.

[16] I. Goodfellow, Y. Bengin, and A. Courville, Deep Learning. The MIT Press, Cambridge, MA, USA, 2016.

[17] M. Laskar, L. Giraldo, O. Schwartz, Correspondence of deep neural networks and the brain for visual textures, 2018, pp. 1–17.

[18] Q. Abbas, M. Ibrahim, M. Jaffar (2019) A comprehensive review of recent advances on deep vision systems. Artif Intell Rev, 2019, vol. 52, pp 39–76. https://doi.org/10.1007/s10462-018-9633-3.

[19] Y. Guo, Y. Liu, A. Oerlemans, S. Lao, S. Wu, and M. S. Lew, “Deep learning for visual understanding: A review”, Neurocomputing, 2016, vol. 187, pp. 27–48.

[20] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. F. Alsaadi, “A survey of deep neural network architectures and their applications”, Neurocomputing, 2017, vol. 234, pp.11-26.

[21] W. Rawat, and Z. Wang, “Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review”, Neural Computation, 2017, vol. 29, pp. 2352–2449, 2017.

[22] D. Scherer, A. Müller, S. Behnke, “Evaluation of pooling operations in convolutional architectures for object recognition”, In: Artificial Neural Networks-ICANN 2010. Springer, pp. 92–101.

[23] L. Fei-Fei, “Knowledge transfer in learning to recognize visual objects classes”, In Proceedings of the 4th International Conference on Development and Learning, 2006, pp 11-17.

[24] A. Krizhevsky, I. Sutskevee, and G. Hinton, “ImageNet classification with deep convolutional neural networks”, Communications of the ACM 2017, vol. 60, no. 6, pp. 84-90. Doi 10.1145/3065386.

[25] O. Russakovsky ”. Int J Comput Vis. https://doi.org/10.1007/s11263-015-0816-y J. Deng, H. Su (2015) “ImageNet large scale visual recognition challenge.

[26] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, W., and L. D. Jackel. “Handwritten digit recognition with a back-propagation network”. In D. S. Touretzky (Ed.), Advances in neural information processing systems, 1989, vol. 2, pp. 396–404, Cambridge, MA: MIT Press.

[27] K. Simonyan, A. Zisserman, “Very deep convolutional networks for large-scale image recognition”, ICLR 2015, vol. 75, pp. 398–406. https://doi.org/10.2146/ajhp170251.

[28] S. Zhang, L. Yaq, A. Sun, Y. Tay, “Deep Learning based Recommender System: A Survey and New Perspectives”, ACM Computing Surveys, 2018, vol. 1, no. 1, pp. 1-35.

[29] A. Khan, A. Sohail, U. Zahoora, A. S. Qureshi, “A Survey of the Recent Architectures of Deep Convolutional Neural Networks”, Artificial Intelligence Review, 2020, vol. 53, pp. 5455–5516. DOI: https://doi.org/10.1007/s10462-020-09825-6.

[30] D. Han, and J. Kim, “Deep pyramidal residual networks”, In: 2017 IEEE conference on computer vision and pattern recognition (CVPR), pp. 6307–6315.

[31] S. Zagoruyko and N. Komodakis, “Wide Residual Networks”, In Richard C. Wilson, Edwin R. Hancock and William A. P. Smith, editors, Proceedings of the British Machine Vision Conference (BMVC), pp.1-87.12. BMVA Press, September 2016.

[32] C. Szegedy, W. Liu, Y. Jia, “Going deeper with convolutions”. In: 2015 IEEE conference on computer vision and pattern recognition (CVPR). Pp. 1–9.

[33] R. Szeliski, R., Computer Vision: Algorithms and Applications. Springler, 2021.

[34] O. M. Parkhi, A. Vedaldi, A. Zisserman, C. V. Jawahar, “Cats and Dogs”. IEEE Conference on Computer Vision and Pattern Recognition, 2012.

[35] Kaggle: https://www.kaggle.com/datasets/alxmamaev/flowers-recognition