STM JOURNALS

Journal of Image Processing & Pattern Recognition Progress
ISSN: 2394-1995 (Online)
Volume 5, Issue 3
www.stmjournals.com

Image Classification using Convolutional Deep Neural Networks

Sudhir Vegad[1],*, *Prashant Italiya*[2], *Zankhana Shah*[3]
[1]Information Technology Department, AD Patel Institute of Technology, Anand, Gujarat, India
[2]Computer Software, eClinicalWorks, India
[3]Information Technology Department, BVM Engineering College, Vallabh Vidyanagar, Anand, Gujarat, India

Abstract

Thousands of images are generated every day, which implies the necessity to classify and access them by an easy and faster way. The main objective of classification is to identify the features occurring in the image. Neural networks (NNs), inspired by biological neural system, are a family of supervised machine learning algorithms that allow machine to learn from training instances as mathematical models. NNs have been widely applied in the fields of classification, optimization, and control theory. This work compares the classification of images using Convolutional Deep Neural Network approaches.

Keywords: *Deep Neural Network, Convolutional Neural Networks, image segmentation*

****Author for Correspondence*** *E-mail : svegad@gmail.com*

INTRODUCTION

Image classification is a process to categorize digital images in databases according to the similarity of their features, content, and semantics [1]. As shown in Figure 1, image classification utilizes supervised machine learning algorithms, such as support vector machine (SVM), and neural networks (NNs), to train a model, also known as classifier, with sample images and associated ground truth labels. The trained model can be used to classify unknow images based on the knowledge learn from the examples. Image classification has been widely used in various applications.

For example, Face ID in iPhone X captures phone owner’s facial images from different angles to train a model which is subsequently used to authenticate the user. As aforementioned, various supervised machine learning algorithms can be adopted to build a classification model by associating the extracted visual features and class labels of the training images. In 2007, Lu and Weng reported that feature extraction and selection is one of the key factors to the success of image classification [2]. To reduce the impact of data redundancy, various approaches, such as principal component analysis (PCA), have been commonly used for feature selection. Most recent studies show that the convolution neural networks can learn features, which outperforms other traditional supervised machine learning algorithms that remit on hand crafted visual features [3]. For this reason, this research mainly focus on the comparative studies of two neural network-based approaches.

The perceptron, the simplest neural network design, was introduced by Rosenblatt in 1958 as a two-layer network, i.e., input layer and output layer, to classify any linearly separable data [4]. By inserting one or more intermediate layers, also known as hidden layer, between the input and output layers, a multilayer perceptron was further developed as the typical neural network architecture. NNs are generally constructed as systems of interconnected “neurons”, which models the relations between a set of input parameters and the output labels based on the provided training examples [5].

Although deep architectures can be used to model complex nonlinear relationships, they have not been discussed much until 2006 due to several challenging issues.

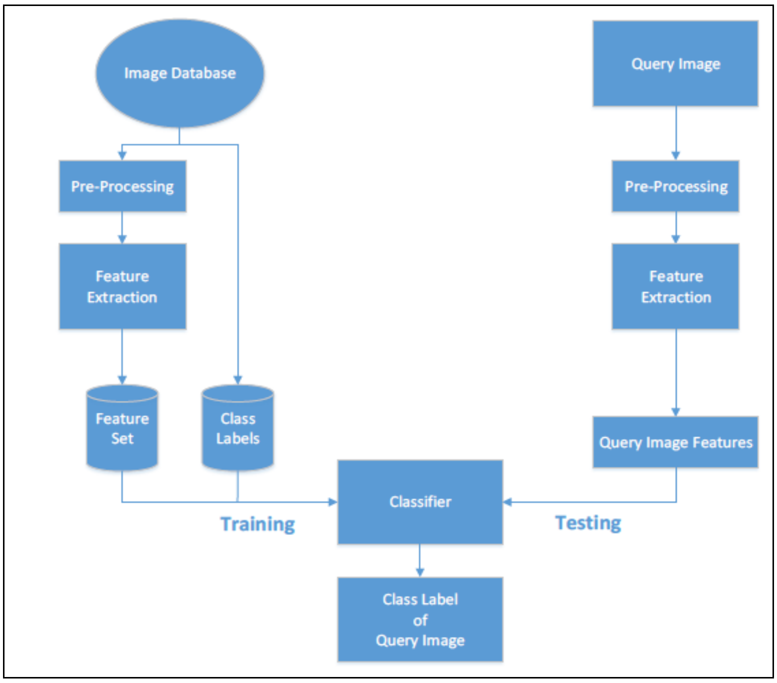

Fig. 1: *A Typical Image Classification Architecture.*

The deep architectures rely on 1) low-cost and high-performance machines to train a model, 2) enormous amount of training data, and 3) efficient and powerful algorithms to process and handle the huge amount of high-dimensional data [6]. In addition, using the standard random initialized parameters commonly leads to poor training and generalization errors [7]. These challenging problems have been resolved in the past few years owning to the development of GPU computing and gigantic hand-annotated dataset such as ImageNet [8, 9]. It is worth to mentioning that convolution neural networks (CNN) is one of the commonly and widely used machine learning algorithms in deep architectures because CNN is capable of extract learned features [3]. In this study, we mainly focus on two well-known deep convolutional neural networks, including AlexNet [10] and VGG16 [11].

RELATED WORK

LeNet-5 [13], convolutional neural networks (CNN) typically had a standard structure stacked convolutional layers are followed by one or more fully-connected layers. Variants of this design are prevalent in the image classification literature and have generated the best results to-date on MNIST (Modified National Institute of Standards and Technology database), CIFAR (Canadian Institute For Advanced Research) and most remarkable on the ImageNet classification challenge [10, 13]. For more massive datasets such as ImageNet, the recent trend has been to increase the number of layers [14], and layer size [13, 15], while using dropout [16], to address the problem of overfitting. Also, max-pooling layers result in loss of accurate spatial information, the same convolutional network architecture as explained by Krizhevsky et al. [10] has also been successfully employed for localization [10, 15], object detection [17, 19] and human pose estimation [20]. Encouraged by a neuroscience model of the primate visual cortex [21], they used different sizes of fixed Gabor filters of to handle various scales. We use a similar technique here. However, contrary to the fixed 2-layer deep model of, all filters in the inception model is learned. Furthermore, inception layers are a repeating number of times, leading to a 22-layer deep model in the case of the GoogLeNet architecture.

Network-in-Network is a method proposed by Lin et al. [14] to enhance the power of neural networks. In Network-in-Network model, additional 1×1 convolutional layers are added to the network model, increasing its depth. We use this method in our model. However, in our setting, 1×1 convolutions have several purposes, most critically; they are used mainly as dimension reduction modules to remove computational tailbacks that would otherwise limit the size of our networks. This allows for not just increasing the depth, but also the width of our networks without significant changes in performance. Finally, the present state of the art for object detection is the Regions with Convolutional Neural Networks (R-CNN) method by Girshick et al. [17]. R-CNN decomposes the overall detection problem into two subproblems: utilizing low-level cues such as color and texture to generate object location proposals in a category-agnostic fashion and using CNN classifiers to identify object categories at those locations. Such a two-stage approach leverages the accuracy of bounding box segmentation with low-level cues, as well as the highly classification power of state-of-the-art CNN. We use a similar pipeline in our detection plans but have explored enhancements in both stages, such as multibox prediction for higher object bounding box recall, and ensemble approaches for better categorization of bounding box proposals [14].

Convolutional Deep Neural Network

Convolutional neural network (CNN) is a category of deep neural networks that have successfully been applied to analyzing visual images. CNN's use a variation of multilayer perceptron designed to require minimal preprocessing. They are also known as shift-invariant artificial neural networks (SIANN), based on their shared-weights architecture and translation invariance characteristics [21, 22].

Convolutional neural networks (CNN) were encouraged by biological processes in which the connection pattern between neurons is inspired by the association of the animal visual cortex. Individual cortical neurons respond to stimuli only in a restricted region of the visual field known as the receptive field. The receptive fields of different neurons partially overlay such that they cover the entire visual field.

A study by Hubel and Wiesel in the 1950s and 1960s presented that cat and monkey visual cortexes contain neurons that independently respond to small regions of the visual field. Provided the eyes are not moving, the region of visual field within which visual stimuli affect the firing of a single neuron is known as its receptive field. Neighboring cells have similar and overlapping receptive fields. Receptive field size and location varies carefully across the cortex to form a complete map of the visual field. The cortex in each hemisphere shows the contralateral visual field. Their paper identified two basic visual cell types in the brain: **Simple cells,** whose output is maximized by straight edges having orientations within their receptive field. **Complex cells** which have more significant receptive fields, and whose output is insensitive to the exact position of the edges in the field.

LeNet-5, a pioneering 7-level convolutional neural network (CNN) by LeCun et al. that classifies digits, was applied by several banks to recognize hand-written digits on checks (Cheques) digitized in 32×32pixel images [7]. The capacity to process higher resolution images requires more significant and more convolutional layers, so this method is constrained by the availability of computing resources.

Following the work by Steinkraus, Simard, and Buck established the value of GPU for machine learning, several publications described more efficient ways to train convolutional neural networks using GPUs [8]. In 2011, they were refined and executed on a GPU, with notable results. In 2012, Ciresan et al. significantly improved on the best performance in the literature for multiple image databases. It includes the MNIST data-set, the NORB data-set, the HWDB1.0 data-set (Chinese characters), the CIFAR10 data-set (data-set of 60000 32x32 labeled RGB images), and the ImageNet data-set [9].

CNN's use relatively little pre-processing compared to other image classification algorithms. It means that the network learns the filters that in traditional algorithms were hand-engineered. This independence from prior knowledge and human effort in feature design is a significant advantage.

Deep Neural Network

So far, many deep learning networks are developed. Here, some of the known networks are explained.

LeNet: LeCun et al. presents implementation of LeNet primarily for character recognition tasks in documents [7]. The LeNet model is straightforward and small, (regarding memory footprint), making it perfect for training the basics of the convolutional neural network. It can even run on the CPU (if your system does not have a suitable GPU), making it a great "first CNN."

AlexNet: AlexNet [10], CNN to classify the 1.2 million high-resolution images for training, 50,000 validation images, and 150,000 testing images in the LSVRC-2010 ImageNet training set into the 1000 different classes. On the test data, AlexNet achieved top-5 error rates of 17% which is much better than the previous state-of-the-art results. AlexNet achieved top-1 error rates of 37.5% which is much better than the previous state-of-the-art results the neural network, which has 60 million parameters and 500,000 neurons, consists of five convolutional layers, some of which are followed by max-pooling layers, and two globally connected layers with a final 1000-way SoftMax. To make training faster, AlexNet used nonsaturating neurons and a very efficient GPU implementation of convolutional nets [10]. The total number of layers (independent building blocks) used for the construction of the network is about 25.

VGG16: VGG16 is a 25-layer network, which refers to a deep convolutional network for object recognition developed by Oxford's renowned Visual Geometry Group (VGG), which achieved very high performance on the ImageNet dataset [11]. The ILSVRC 2014 classification challenge involves the task of classifying the image into one of 1000 categories in the Imagenet hierarchy. There are about 1.3 million images used for training data-set, 50,000 for validation-set and 100,000 images for testing-set. VGG16 result in the challenge obtains a top-5 error of 7.3% on both the validation and testing data, ranking the second among other participants in ILSVRS 2014 competition for classification task. VGG16 result in the challenge obtains a top-5 error of 24.7 on both the validation and testing data [11].

COMPARISION OVERVIEW

The experimentations have been carried out for VGG16 and AlexNet. Different input parameters were tested on the dataset. Specified numbers of iterations and images are chosen per class to differentiate both models. The evaluation is divided into two different cases, 1) Train both networks on a similar number of images per class with a different number of iterations. We do not find any performance benchmark of these networks on a small dataset, so we choose iterations starting from 0 to 600 in different incremental order. We choose iteration as a key parameter. Iteration is the full pass of the training algorithm over the entire training set. Network learns more after each iteration as same as brain (Hubel and Wiesel, 1968). In this case, we evaluate the influence of changes in number of iterations on the performance behavior of network model. Here, we train 22 different models on the same dataset which contains 100 images per class using different numbers of iteration. 2) Here, we choose size of data-set size as a key parameter for our second experiment. We evaluate the influence of changes dataset on the performance behavior of network model. Here, we train five different models where number of iterations stay same for both models which are 100, but the number of images per class in each dataset is different for both models 100, 200 and 300, 400, 500, respectively. We are increasing the size of data-set means we are providing more learning resources to network, respectively.

Dataset Selection

Dataset used in this study are collected from several datasets and divided into five different categories. We used images from CIFAR-10 dataset and ILSVRC-12 data-set. The CIFAR-10 dataset consists of 60,000 color images in 10 classes, with 6,000 images per class. There are 50,000 training images and 10,000 test images. ILSVRC-12 data-set consists of 1.2 million images from 1,000 different categories. Also include 50,000 validation images, and 150,000 testing images. The five categories in our dataset are dog, cat, airplane, clutter, and Motorcycle. As in normally used convolutional neural network models (AlexNet, VGG16), transformations were applied to images. To fairly test both networks, we make sure to train and test both networks on all images available in the dataset. To achieve this, we used cross-validated dataset. Selection of cross-validated dataset is described below.

Dataset Cross Validation

For training, we choose 80% of a total number of images in dataset and rest of 20% images for testing. To make sure both networks are trained and tested on each image available in dataset during training and testing, we divided our dataset into five bins, and we used a possible combination of all five bins using (5P1). Using all five possible combinations, we trained both networks five times for each training; we used four unique bins as training and rest bin as a testing dataset. Selection of dataset for each training network is described in Table 1.

Table 1: *Cross Validatoon Bin Selection.*

Network	Testing Set	Training Set			
No.	*Bin*	*Bin*	*Bin*	*Bin*	*Bin*
1	1	2	3	4	5
2	2	3	4	5	1
3	3	4	5	1	2
4	4	5	1	2	3
5	5	1	2	3	4

Performance Evaluation Matrix

Accuracy is calculated by how many images we can correctly identify from all the testing images to total number of available testing images. While training both networks five times with cross-validated dataset, we calculated our average accuracy using following formulas:

$$Accuracy_{(i)} = \frac{Correct\ Identified Images}{Total\ Number of\ Images in Testing\ Set} \quad (1)$$

$$AVG_{Accuracy} = \frac{Accuracy(1)+Accuracy(2)+\cdots+Accuracy(5)}{5} \quad (2)$$

EXPERIMENTAL RESULTS

The experimental work is carried out using the images from CIFAR-10 and ILSVRC-12 data-set. The database is partitioned into five categories of Dog, Cat, Airplane, Motorbike, and Clutter. Images from each category are shown in Figure 2. To realize the proposed experiments MATLAB is used. Image Processing toolbox, Statistics and Machine Learning Toolbox and the Neural Network Toolbox of MATLAB are used to implement both networks. Both networks which are used as a classifier are setup and configured with parameters that are best suitable for image classification task. The configuration includes setting the default Learning rate to 0.001 and selecting the SGDM (Stochastic Gradient Descent with Momentum) as training solver. Then, both networks are trained on cross-validated dataset having a similar dataset for each network.

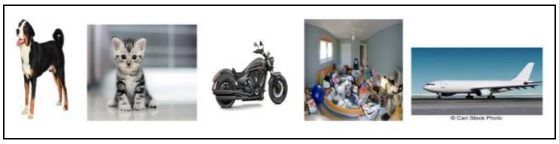

Fig. 1. *Examples of Image Dataset used in Our Experiments.*

Stochastic Gradient Descend with Momentum (SGDM)

The gradient descent algorithm updates the parameters (weights and biases) to minimize the error function by taking small steps in the direction of the negative gradient of the loss function:

$$\theta_{l+1}=\theta_l-\alpha\nabla E(\theta_l) \quad (3)$$

where, l stands for the iteration number, α > 0 is the learning rate, θ is the parameter vector, and E(θ) is the loss function. The gradient of the loss function, ∇E(θ), is evaluated using the entire training set, and the standard gradient descent algorithm uses the entire data set at once. The stochastic gradient descent algorithm evaluates the gradient and updates the parameters using a subset of the training set. This subset is called a mini-batch. Each evaluation of the gradient using the mini-batch is an iteration. At each iteration, the algorithm takes one step towards minimizing the loss function. The full pass of the training algorithm over the entire training set using mini-batches is an epoch. The gradient descent algorithm might oscillate along the steepest descent path to the optimum. Adding a momentum term to the parameter update is one way to prevent this oscillation [1]. The stochastic gradient descent update with momentum is where γ determines the contribution of the previous gradient step to the current iteration.

$$\theta_{l+1}=\theta_{l}-\alpha\nabla E(\theta_{l})+\gamma(\theta_{l}-\theta_{\theta l-1}) \qquad (4)$$

Result: Case-1

In this experiment, we took constant 100 images per each class, and number of iterations is starting from 0 to 600 in different incremental order to determine the influence of both network with a change in the number of iterations, tables given below shows the accuracy of both model for our first case. We choose an iteration as a key parameter because iteration is the full pass of the training algorithm over the entire training set. This experiment helps us to determine how the changes in number of iteration affect the performance of the network. The experimental result for this case shows that VGG16 gave us more accuracy when the dataset is small, and number of iterations is moderately high [12, 18].

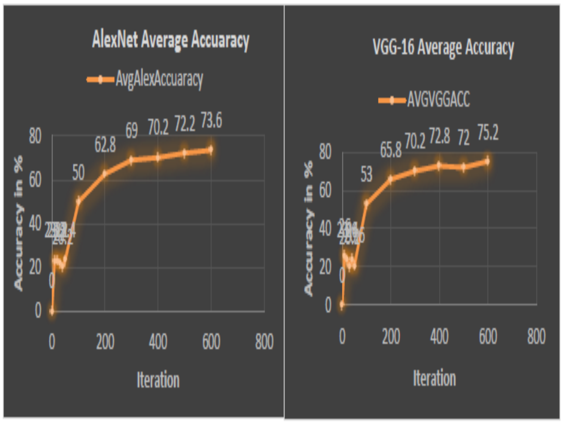

Fig. 2: *Average Accuracy Comparison of AlexNet and VGG16 for Case-1.*

Table 2: *Alexnet Results For Case-1.*

Iterations	**AlexNet Accuracy – 100 Images / Class**					
	Fold-1	***Fold-2***	***Fold-3***	***Fold-4***	***Fold-5***	***Average***
50	23	20	19	28	27	24.4
100	52	60	43	51	44	50
200	69	64	59	75	47	62.8
300	68	66	70	68	73	69
400	76	75	69	69	62	70.2
500	70	65	74	78	74	72.2
600	77	78	71	74	68	73.6

Table 3: *VGG16 Results for Case-1.*

Iterations	**VGG16 Accuracy – 100 Images / Class**					
	Fold-1	***Fold-2***	***Fold-3***	***Fold-4***	***Fold-5***	***Average***
50	20	18	19	20	26	20.6
100	51	58	59	44	53	53
200	68	56	68	64	73	65.8
300	67	74	71	73	69	70.2
400	76	76	72	73	67	72.8
500	76	69	69	78	75	73.4
600	80	70	80	67	79	75.2

Figure 3 shows the average accuracy of both models versus iteration using the data available in Tables 2 and 3 for case-1.

Result: Case-2

In this experiment, we took constant 100 iterations, and number of images per class is starting from 100 to 500 in incremental order of 100 to determine the influence of both network with a change in the number of images per class, tables given below shows the accuracy of both model for our second case. We are increasing the size of data-set means we are providing more learning resources to a network, respectively. This experiment helps us to determine how the changes in the size of data-set affect the performance of the network. The experimental result for this case shows that AlexNet gave us more accuracy when there will be strict computation resource available (less number of iterations to train network), and moderately high number of images per class.

Table 4: *Alexnet Results For Case-2.*

Images per Class	**AlexNet Accuracy – 100 Iteration1**					
	Fold-1	***Fold-2***	***Fold-3***	***Fold-4***	***Fold-5***	***Average***
0	0	0	0	0	0	0
100	52	60	43	51	44	50
200	58	56.5	30.5	61.5	59.5	53.2
300	59.3	56	52	55	66	57.7
400	54	58.5	48.5	65.2	56	56.4
500	50.4	22	66	49.8	58.8	49.4

Table 5: *Vgg16 Results For Case-2.*

Images per Class	**VGG16 Accuracy – 100 Iteration**					
	Fold-1	***Fold-2***	***Fold-3***	***Fold-4***	***Fold-5***	***Average***
0	0	0	0	0	0	0
100	51	58	59	44	53	53
200	57	50.5	49	39.5	51.5	49.6
300	52	55	47	24.7	54.7	46.8
400	39.8	56.8	44.8	49.2	58.5	49.8
500	56.6	41.6	60.2	44.8	4.6	49.6

Figure 4 shows the average accuracy of both models versus iteration using the data available in Tables 4 and 5 for case-2.

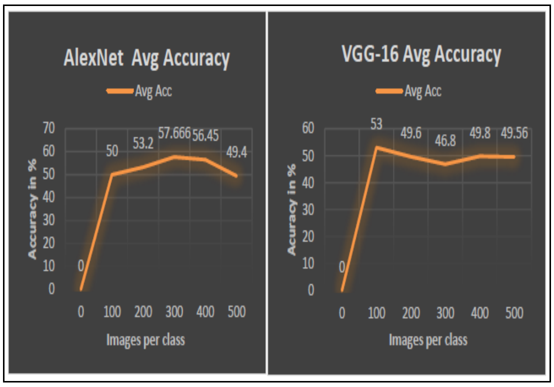

Fig. 2. *Average Accuracy Comparison of AlexNet and VGG16 for Case-2.*

CONCLUSION

The most important aspect of this study was to find out and compare performance of two of the most known convolution neural network on small computational resources and small dataset. For our dataset, when we consider iteration as key parameter for experiment, VGG16 is more suitable then AlexNet. The experimental results demonstrate that VGG16 give us more accuracy when dataset is small, and number of iterations is moderately high. When, we consider data-set size as key parameter for experiment, AlexNet is more suitable then VGG16. Our results show that AlexNet give us more accuracy when there will be strict computation resource available (less number of iterations to train network), and moderately high number of images per class.

REFERENCES

1. Vegad SP, Italiya PK. Image classification using neural network for efficient image retrieval. In *Electrical, Electronics, Signals, Communication and Optimization (EESCO), 2015 International Conference on*, IEEE, 2015, 1–6p.
2. Lu D, Weng Q. A survey of image classification methods and techniques for improving classification performance. *Int J Remote Sens*. 2007; 28(5): 823–870p.
3. Antipov G, Berrani SA, Ruchaud N, Dugelay JL. Learned vs. hand-crafted features for pedestrian gender recognition. In *Proceedings of the 23rd ACM international conference on Multimedia*, ACM, 2015, 1263–1266p.
4. Rosenblatt F. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychol Review*. 1958; 65(6): 386p.
5. Keijsers NL. *Neural Networks*. (K. K. L., Ed.) Cambridge, MA: Academic Press, 2010.
6. Bengio Y. Learning deep architectures for AI. *Found trends® Mach Learn*. 2009; 2(1): 1–127p.
7. LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, 1998; 86(11): 2278–2324p.
8. Steinkraus D, BuckI, Simard PY. Using GPUs for machine learning algorithms. In *Document Analysis and Recognition, 2005. Proceedings. Eighth International Conference on*, IEEE, 2005, 1115–1120p.
9. Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Z, Karpathy A, Khosla A, Bernstein M, Berg AC, Li F-F. ImageNet large scale visual recognition challenge. *Int J Comp Vis*. 2015; 115(3): 211–252p.
10. Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. In *Adv Neur Info Process Syst*. 2012, 1097–1105p.
11. Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. In *Intelligent Control and Automation*. 2014, 7(4), arXiv preprint arXiv:1409.1556.
12. LeCun Y, Boser B, Denker JS, Henderson D, Howard RE, Hubbard W, Jackel LD. Back propagation applied to handwritten zip code recognition. *Neural Comput*. 1989; 1(4): 541–551p.
13. Zeiler MD, Fergus R. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision*, Springer, Cham, 2014, 818–833p.
14. Lin M, Chen Q, Yan S. Network in network. 2013, arXiv preprint arXiv:1312.4400.
15. Sermanet P, Eigen D, Zhang X, Mathieu M, Fergus R, LeCun Y. Overfeat: Integrated recognition, localization and detection using convolutional networks. 2013, arXiv preprint arXiv:1312.6229.
16. Hinton GE, Srivastava N, Krizhevsky A, Sutskever I, Salakhutdinov RR. Improving neural networks by preventing co-adaptation of feature detectors, 2012, arXiv preprint arXiv:1207.0580.

17. Girshick R, Donahue J, Darrell T, Malik J. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, 580–587p.
18. Szegedy C, Toshev A, Erhan D. Deep neural networks for object detection. In *Adv Neur Info Process Syst*. 2013, 2553–2561p.
19. Toshev A, Szegedy C. DeepPose: Human pose estimation via deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, 1653–1660p.
20. Serre T, Wolf L, Bileschi S, Riesenhuber M, Poggio T. Robust object recognition with cortex-like mechanisms. *IEEE T Pattern Anal*. 2007, 29(3): 411–426p.
21. Zhang W, Giger ML, Nishikawa RM, Schmidt RA. An improved shift-invariant artificial neural network for computerized detection of clustered microcalcifications in digital mammograms. *Med Phys*. 1996; 23(4): 595–601p.
22. Zhang W, Itoh K, Tanida J, Ichioka Y. Parallel distributed processing model with local space-invariant interconnections and its optical architecture. *Appl Optics*. 1990; 29(32): 4790–4797p.

Cite this Article

Sudhir Vegad, Prashant Italiya, Zankhana Shah, Image Classification using Convolutional Deep Neural Networks. *Journal of Image Processing & Pattern Recognition Progress*. 2018; 5(3): 7–14p.