ON DEEP CONVOLUTIONAL NETWORKS FOR LARGE SCALE IMAGE CLASSIFICATION

Mahesh C., Student Member, IEEE, Madhu S. Nair, Member, IEEE

Department of Computer Science, University of Kerala, Kariavattom
Thiruvananthapuram-695581, Kerala, India.

ABSTRACT

Image classification is an essential to the field of computer vision based systems and recent research in this area explores better feature extraction, feature coding, and classification. The purpose of this paper is to review the application of supervised deep convolution networks in this field. Numerous techniques have been reported to improve the performance of the convolution networks. Experimental analysis shows that with a large amount of labeled data, convolution networks can learn very complex functions such as image classification.

***Index Terms*—** convolution networks, deep learning, neural networks.

1. INTRODUCTION

The performance of an image classification system mainly depends on the extraction and representation of features. Feature representation methods like Haralick texture features [1] got attention from the research community from the earlier days of image classification. However, to develop features that are invariant to position, rotation, scaling, and distortion, researchers had to explore the visual perception of primates. This research led to the development of many models such as convolutional neural networks[2] and Kohonen map[3]. As a result, many successful image classification systems are implemented with better accuracy [4].

Early in that stage, method like the convolutional network is limited by the availability of labeled data and computing infrastructure. Researchers are trying to overcome this problem by many other non-parametric models such as SVM and KNN. But these methods couldn't give a high accuracy result on any of the large-scale classification problem and limited by the preprocessing technique. However, in the recent years, the development of High Performance Computing (HPC) architecture such as General Purpose Graphical Processing Units (GPGPU) accelerated the research in this field. Large scale image dataset such as ImageNet [5] with millions of labeled samples is also accessible to the research community. These changes in data and computing, put back the convolution network with millions of parameters in track.

2. MULTI-STAGE HUBEL-WIESEL ARCHITECTURE

In 1962, Hubel DH and Wiesel TN [6], [7] studied visual cortex of anesthetized cats with spots of white light of various shapes. Cells in the visual system are classified into simple, complex and hypercomplex. Simple cells are influenced by the arrangement of excitatory and inhibitory regions of the receptive field as well as the position of the stimulus. This cell receives input from cells of the lateral geniculate nucleus (LGN), which is connected to the retina. However, the complex cells will respond to an appropriately oriented stimulus regardless of the cell position in the receptive field. Complex cells are activated by edge, dark bar, slit and mixed stimuli. Hypercomplex cells are activated by edge, single-stopped (corner), double-stopped (tongue), slit (double-stopped) and dark bar (double-stopped).

In the visual cortex, perception cells are in the order, simple⟹complex⟹lower-order-hypercomplex⟹higher-order-hypercomplex. Activation of a lower stage is influenced by the position of the input patterns, and higher stages are position-invariant. There are several contradictory to this structure, but no one completely deny this hierarchical model.

Inspired by this work, Fukushima, K [8] proposed a neural network model for pattern recognition called neocognitron. In neocognitron, cells are arranged in a number of cascaded structure. Each structure U include a simple cell layer U_s and a complex cell layer U_c. This network is not affected by change in position or small distortion in the shape of patterns. It is also capable of doing self-organization based on an unsupervised competitive learning algorithm [9] in the first two layers and classification based on supervised learning in the output layer.

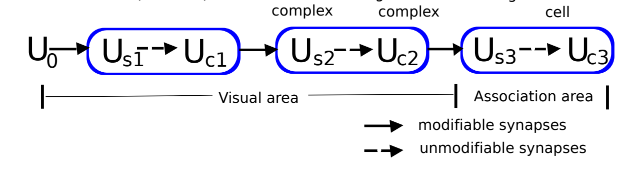

Fig. 1: Neocognitron

3. CONVOLUTIONAL NETWORKS

Neocognitron was improved by Yann LeCun [10], [4], using backpropagation algorithm to train the entire system. It uses local receptive fields, share weights and sub-sampling to achieve shift, position and distortion invariance. A typical Convolutional Network called LeNet-5 was proposed by Yann LeCun et al. [2] for document recognition. Using local receptive fields, network can extract elementary visual features such as edges, end points and corners. These features will be combined to obtain high order features in the following layers. Elementary feature detectors with identical weights can be useful in different parts of the image. So the units with the same set of weights are arranged in plane, and output from the units of a plane is called a feature map. Units in a feature map perform the same operation on different parts of the same image. A convolution layer is composed of the set of feature maps with differently weighed units. In the implementation, a unit in the feature map scans the image and store the states in the feature map. This operation is equivalent to convolution with a kernel composed of a set of weights and image.

A typical convolutional network is composed of multiple stages with a filter bank layer, a non-linearity layer and a feature pooling layer [11] followed by a classification network.

Filter Bank Layer: This layer computes y_j the convolution

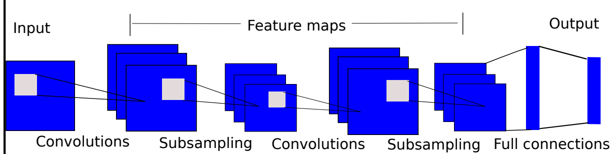

Fig. 2: A typical convolutional network architecture

between an input feature map x_i and trainable filter kernel k_{ij} ie. $y_j = b_j + \sum_i k_{ij} * x_i$, where b_j is a trainable bias, i and j are array indices, and $*$ is the convolution operator.

Non-Linearity Layer: This layer applies a non-linearity function such as $tanh(x)$ or $(1 + e^{-x})^{-1}$ to unit output. But to reduce training time with gradient descent, new implementations uses the function $max(0, x)$. Units with this non-linearity is called Rectified Linear Units (ReLUs) [12].

Feature Pooling Layer: It reduces the dimension of feature map by applying the techniques like averaging or max-pooling.

But this architecture was limited by the availability of computing power and sample data sets.

4. DEEP CONVOLUTIONAL NETWORKS

In the last few years, convolutional networks shows a significant performance improvement in many small scale image classification on data sets such as MNIST [13], CIFAR-10, CIFAR-100, SVHN [14], and STL-10 [15]. Krizhevsky, et

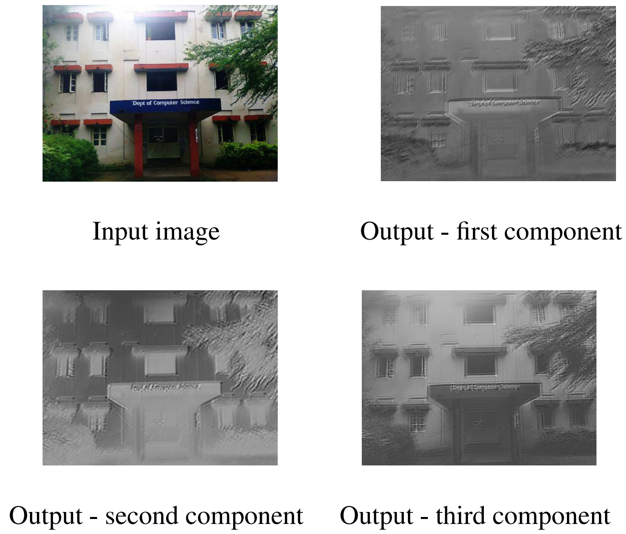

Fig. 3: Output of a randomly initialized convolution filter.

al. [16] proposed a network of 60 million parameters and 650,000 neurons, with five convolutional layers followed by max-pooling layers and three fully-connected layers. Data set used in this experiment was a subset of ImageNet dataset, used in the competition ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) [5] and reported an error rate of 15.3%. ILSVRC data set includes 1.2 million images that contain 1,000 categories. This network uses rectifier as the function in neurons. Even if it shows improvement in speed, this ill-conditioned parameterization must be studied further to understand the effect in very large networks.

4.1. Network in Network

Inspired by the work of Ian J. Goodfellow et al..[17] on max out networks, Min Lin et al. [18] introduced a micro-network in each convolution layer so that it will compute more abstract features.This network gave a state-of-the-art performance in ILSVRC 2013 competition with an error rate of 12.95%.They used NVIDIA TITAN GPU to train the network. Using multilayer perceptron instead of voting to approximate convex functions of each local patches may result in a good accuracy; but this is equivalent to moving linear separability problem into another hyperspace. So, if the input is in high frequency, this will end up in modeling large number of hidden layers and make the network structure more dependent on the problem. It will diverge the idea of the convolution network from *learn anything* to a *local search.*

4.2. Visualizing Convolutional Networks

Matthew D. Zeiler and Rob Fergus [19] presented a method to visualize the function of intermediate feature layers of convolutional networks and used it as a diagnostic tool to improve the model proposed by Krizhevsky et al. [16]. This method helped them to understand the activation in the feature maps with respect to the input patterns. It shows that Krizhevsky et al.'s architecture does not have enough mid frequency coverage in the first layer filters and causes aliasing artifacts by large stride in the first layer convolutions. Authors solved this problem by decreasing filter size to 7×7 and reducing stride to 2. This implementation won the ILSVRC 2013 competition with an error rate of 11.74%. These techniques might be more useful if it can relate to any of the learning formulations instead of vague approximations of a non-invertible function.

4.3. Spatial Pyramid Pooling in Deep Convolutional Networks

Instead of using fixed input size in convolutional networks, Kaiming He et al. [20] suggested the use of a spooling strategy called Spatial Pyramid Pooling (SPP) [21] [22] to avoid cropping or warping of images. It introduced a new layer on top of the convolution layer and performed aggregation based on Bag-of-Words (BoW) model. However, the classical backpropagation training methods expect layers to have a fixed size. This problem can be solved by using two fixed size networks with shared parameters and switch the network on alternate epochs. Network was trained using a single GeForce GTX Titan GPU with a starting learning rate of 0.01 and achieved a less error rate of 8.06% on ILSVRC 2014 data set.

This implementation improves the performance of baseline architectures including ZF-5 [19], Convnet [16] and Overfeat-5/7. Even if the accuracy of convolutional networks will improve on multi-size training, multi-level pooling, and full-image representations, all these methods will increase both time and space complexity of the system.

4.4. Going deeper with convolutions

Christian Szegedy et al. [23] proposed a network named GoogLeNet with receptive field (input layer) of size 244×244 with the number of layers around 100. Network is trained using asynchronous stochastic gradient descent with 0.9 momentum and fixed learning rate schedule based on the number of epochs. Learning procedure took advantage of model and data-parallelism in a CPU-based cluster environment. This network gave an error rate of 6.67% on ILSVRC 2014 data set. The experimental analysis shows that use of existing dense blocks to build the sparse structure can improve the performance of convolutional networks. Even if higher depth will increase the accuracy, this system will end up with implementing large number of hidden layers to increase accuracy for a high frequency input.

5. VERY DEEP CONVOLUTIONAL NETWORKS

Karen Simonyan and Andrew Zisserman [24] evaluated the effect of network depth in image classification using very small convolution filters. Their deep network architecture comprised of fixed size input layers, a stack of convolution layers, three Fully-Connected (FC) layers and 5 max-pooling layers for spatial pooling over a 2×2 pixel window with stride 2. Hidden layers are modeled using Rectified Linear Units (ReLU) [12]. On the hardware side, it uses a multi-GPU system with NVIDIA Titan Black GPUs. Network is trained using multinomial logistic regression based on back-propagation with momentum of 0.9 and batch size 256.

It has been observed that greater depth with small convolution filters and initialization of certain layers will cause the learning process to converge in less number of epochs and gain significant improvement in accuracy. This model of the convolution network does not differ from the classical architecture proposed by LeCun et al. [2]. This implementation results in a significant improvement in accuracy with an error rate of 6.8% in ILSVRC 2014 of ImageNet.

5.1. Scaling up Image Classification

The latest attempt in image classification with an error rate of 5.98% in ImageNet data set is reported by Ren Wu et al. [25] of Baidu research. They developed an end to end deep learning system named Deep Image. It uses a highly optimized parallel algorithm to implement large deep neural network with augmented input data. The network is trained using Stochastic Gradient Descent algorithms (SGD) on a custom built high performance system comprised of 36 server nodes, each with 2 six-core Intel Xeon E5-2620 processors and 4 NVIDIA Tesla K40m GPUs. System uses an InfiniBand network for interconnections. Parallelism strategies used in this network are model-data parallelism and data parallelism. These methods have been proposed by Alex Krizhevsky [26] and Omry Yadan et al. [27] for training convolutional neural networks with SGD on a multiple GPU systems. However, it is not easy to extend the same strategies to multiple GPU clusters because of the communication overhead. The major objective is to minimize network data transfers and dynamic computation. So it uses butterfly synchronization and lazy update strategies to achieve data parallelism in the gradient computation. These approaches shows that model-data parallelism is better when number of GPUs is less than 16. Implementation of Data parallelism in a large number of GPU cluster is better because of the constant communication requirements.

Data augmentation techniques are used to increase the number of labeled images in the training set. This includes color casting, vignetting, lens distortion, rotation, flipping and cropping. But this data augmentation techniques doesn't solve the major problems such as occlusion, presence or

Table 1: Comparative analysis of state-of-the-art deep convolutional network based image classification algorithms using ILSVRC dataset.

Team	Year	Data Augmentation	Scalable over network	Time taken	Hardware	Error rate	Observations
Ren Wu et al. [25]	2015	Aggressive	Yes	8.8 hours	Multi-GPU Cluster	5.98%	Aggressive augmentation will make the problem dependent on data set.
Karen Simonyan et al. [24]	2014	Minimum	No	3 weeks	Multi-GPU	6.80%	Effect of small convolution filter in low frequency domain need to be studied.
Christian Szegedy et al. [23]	2014	Minimum	Yes	1 week	CPU Cluster	6.67%	Inserting more layers will make system depend on the data set.
Kaiming He et al. [20]	2014	Minimum	No	4 weeks	Single GPU	8.06%	Good method, but higher time complexity.
Matthew D. Zeiler et al. [19]	2013	Minimum	No	12 days	Single GPU	11.74%	Visualizations are not possible in very deep network.
Min Lin et al. [18]	2013	Minimum	No	12 days	Single GPU	12.95%	Inserting more layers will over-fit the system.
Krizhevsky, et al.[16]	2012	Minimum	No	6 days	Multi-GPU	15.30%	Effect of rectifier must be studied further.

absence of structural components, and lighting conditions. Major contribution of this work is the demonstration of tremendous computational power required to achieve high accuracy in image classification. It also shows that augmented multi-scale images can be combined to achieve less error rate in convolutional network in the context of the image classification.

6. OBSERVATIONS

In recent years, the research community has reported several image classification algorithms based on deep convolutional network which gives less rate. A comparative analysis of recent futuristic deep convolutional networks is shown in Table 1. Ren Wu et al. [25] uses aggressive data argumentation and high computational power to reduce the computational time and error rate. Karen Simonyan et al. [24] work shows, error rate can be reduced using small convolution filters. Christian Szegedy et al. [23] and Min Lin et al. [18] introduces a set of hidden layers to model local patches. Kaiming He et al. [20] uses a different pooling technique to reduce the error rate. Krizhevsky, et al. [16] introduces rectifier as the neuron function. Although significant progress has been made in the last few years, there is still room for further research and improvement. Major observations from our study are listed below.

- Wisely chosen data augmentation techniques can increase the performance of the network. But need to be tested with multiple data set.
- Data parallelism and model parallelism can increase the speed of the training process. This area can be exploited further to increase the speed of the training process.
- Greater depth with small convolution filters will improve the accuracy. But, response to different frequency input must be studied.
- Performance of the network will get improved if images at difference scales are used for training.
- Use of different pooling technique such as spatial pyramid pooling in sub-architectural level may reduce the error rate.
- Rectified Linear Units can increase the speed of SGD.

7. CONCLUSION

This paper attempts to provide a review of research on deep convolutional networks and provide an overview of its architecture and performance. Majority of the reviewed works are reported from ImageNet Large-Scale Visual Recognition Challenge. These networks can only be trained using very expensive computing resources such as multiGPU cluster to achieve more accuracy on large data sets. So this research heavily depends on other research domains such as parallel algorithms, computer networks and multicore architecture. Because of the heavy computational requirements, it is not easy to apply this method directly to the small level computing platforms such as embedded systems and application level processors. On the other side, these methods can easily bring live with the help of cloud computing infrastructure, so to the mobility solutions such as mobile phones and the web.

8. REFERENCES

[1] Robert M. Haralick, K. Shanmugam, and Dinstein, "Textural features for image classification," 1973.

[2] LeCun et al., "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, pp. 2278–2323, 1998.

[3] Teuvo Kohonen, "Self-organized formation of topologically correct feature maps," *Biological cybernetics*, vol. 43, no. 1, pp. 59–69, 1982.

[4] Y. LeCun et al., "Backpropagation applied to handwritten zip code recognition," *Neural Computation*, vol. 1, no. 4, pp. 541–551, Winter 1989.

[5] Olga Russakovsky et al., "Imagenet large scale visual recognition challenge," *CoRR*, vol. abs/1409.0575, 2014.

[6] Wiesel TN Hubel DH, "Receptive fields, binocular interaction and functional architecture in the cats visual cortex," *The Journal of Physiology*, vol. 160, no. 1, pp. 106–154, 1962.

[7] D H Hubel and T N Wiesel, "Receptive Fields and Functional Architecture in Two Nonstriate Visual Areas (18 and 19) of the Cat.," *Journal of neurophysiology*, vol. 28, pp. 229–289, 1965.

[8] K Fukushima, "Neocognitron: a self organizing neural network model for a mechanism of pattern recognition unaffected by shift in position.," *Biological cybernetics*, vol. 36, pp. 193–202, 1980.

[9] Kunihiko Fukushima and Sei Miyake, "Neocognitron: A new algorithm for pattern recognition tolerant of deformations and shifts in position," *Pattern Recognition*, vol. 15, no. 6, pp. 455–469, 1982.

[10] Y. LeCun, "Learning processes in an asymmetric threshold network," in *Disordered systems and biological organization*, Les Houches, France, 1986, pp. 233–240, Springer-Verlag.

[11] Yann LeCun, Koray Kavukcuoglu, and Clément Farabet, "Convolutional networks and applications in vision," in *Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on*. IEEE, 2010, pp. 253–256.

[12] Vinod Nair and Geoffrey E Hinton, "Rectified Linear Units Improve Restricted Boltzmann Machines," *Proc. 27th Int. Conf. Mach. Learn.*, , no. 3, pp. 807–814, 2010.

[13] Dan Claudiu Ciresan et al., "Deep big multilayer perceptrons for digit recognition," *Neural Networks Tricks of the Trade*, vol. 1, pp. 581–598, 2012.

[14] Chen-Yu Lee, Saining Xie, Patrick Gallagher, Zhengyou Zhang, and Zhuowen Tu, "Deeply-supervised nets," *arXiv preprint arXiv:1409.5185*, 2014.

[15] Bogdan Miclut et al., "Committees of deep feedforward networks trained with few data," *CoRR*, vol. abs/1406.5947, 2014.

[16] Alex Krizhevsky and et al., "ImageNet Classification with Deep Convolutional Neural Networks," *Advances In Neural Information Processing Systems*, pp. 1–9, 2012.

[17] Ian J. Goodfellow and et al., "Maxout Networks," *arXiv preprint*, pp. 1319–1327, 2013.

[18] Min Lin, Qiang Chen, and Shuicheng Yan, "Network In Network," *arXiv preprint*, p. 10, 2013.

[19] Matthew D Zeiler and Rob Fergus, "Visualizing and Understanding Convolutional Networks," *arXiv preprint arXiv:1311.2901*, pp. 818–833, 2013.

[20] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. PP, no. 99, pp. 1–1, 2015.

[21] Kristen Grauman and Trevor Darrell, "The pyramid match kernel: Discriminative classification with sets of image features," *Proceedings of the IEEE International Conference on Computer Vision*, vol. II, no. October, pp. 1458–1465, 2005.

[22] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, 2006, vol. 2, pp. 2169–2178.

[23] Christian Szegedy et al., "Going deeper with convolutions," *CoRR*, vol. abs/1409.4842, 2014.

[24] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.

[25] Ren Wu, Shengen Yan, Yi Shan, Qingqing Dang, and Gang Sun, "Deep Image: Scaling up Image Recognition," *arXiv Prepr. arXiv1501.02876*, 2015.

[26] Alex Krizhevsky, "One weird trick for parallelizing convolutional neural networks," pp. 1–7, 2014.

[27] O Yadan and Keith Adams, "Multi-GPU Training of ConvNets," *arXiv*, pp. 10–13, 2013.