A More Efficient CNN Architecture for Plankton Classification

Jiangpeng Yan[1,2](✉), Xiu Li[1,2], and Zuoying Cui[1,2]

[1] Department of Automation, Tsinghua University, Beijing 100084, China
yanjump@outlook.com, czy15@mails.tsinghua.edu.cn
[2] Graduate School at Shenzhen, Tsinghua University, Shenzhen 518055, China
li.xiu@sz.tsinghua.edu.cn

Abstract. With mass data collected by seafloor observation networks, an autonomous system which helps to annotate these pictures are in great demand. In this paper, we study the relationship between the network architecture and the classification accuracy for the Plankton Dataset collected by Oregon State University's Hatfield Marine Science Center. We use multiple classic deep convolutional neural networks (CNN) models to compare the benefit and cost of deeper models which have performed quite well in ImageNet Large Scale Visual Recognition Challenge (ILSVRC) (http://www.image-net.org/challenges/LSVRC) competitions and we discover a hidden degeneration phenomenon. Then we conclude some skills to make CNN smaller and finally propose a more efficient network architecture whose model is much smaller (only 1.5 MB), faster (32.2 fps) and achieve a top-5 accuracy of 96% in the Plankton Dataset. This model can be actually deployed in the seafloor observation network system with its advantages.

Keywords: Plankton classification · Neural networks
Network architecture

1 Introduction

With the rapid development of the seafloor observation networks, lots of visual resources are captured by underwater camera for marine research. Take Oregon State University's Hatfield Marine Science Center for example. Hatfield scientists got 50 million plankton pictures over an 18-day period, which was more than 80 TB. 30,336 images of 121 classes in these pictures are labeled as the Plankton Dataset to hold a Kaggle data science competition[1] in 2015. Faced with such a big number of pictures, an autonomous system which helps to classify and annotate these pictures are in great demand. It is very important for this system to balance classification accuracy and speed properly.

In the competition we mentioned above, classification accuracy matters more than speed. The No.1 team called "Deep Sea", used a parallel CNN model and

[1] https://www.kaggle.com/c/datasciencebowl/data.

J. Yang et al. (Eds.): CCCV 2017, Part III, CCIS 773, pp. 198–208, 2017.
https://doi.org/10.1007/978-981-10-7305-2_18

achieved a top-5 accuracy over 98% and a frame rate of 1.4 fps, which is far from the general video frame rate 25–30 fps. In order to get the high accuracy, this team used one network to learn pixel features and the other to learn additional image features including image size in pixels, Haralick texture features, etc. The final ensemble network model is more than 300M bytes, and quite difficult to train.

In this paper, we try to find out the relationship between the network architecture and the classification accuracy of the Plankton Dataset. We use multiple classic deep CNN models and compare the benefit and cost of deeper models which have performed quite well in ILSVRC competitions and we discover a hid-den degeneration phenomenon for the Plankton Dataset. Then we conclude some skills to make CNN lighter and finally propose a simple but efficient network architecture whose model is much smaller, faster and achieve a top-5 accuracy of 96% in the Plankton Dataset, which can be actually deployed in the seafloor observation network system.

2 Related Work

The deep CNN technology have come into vogue since 2012, when Krizhevsky et al. [10] achieved a record-breaking top-1 and top-5 error rates of 37.5% and 17.0% in ILSVRC competition object classification task [1] using a 7-layer CNN architecture called AlexNet. After that, more and more researchers became interested in CNN method and a series of breakthroughs for object classification was made [5,15–17].

2.1 Going Deeper

As early as 1990s, Lecun et al. [11,12] applied CNN in digital handwritten number and document classification. Due to limitations of computation devices, shallow CNN architecture did not make much progress than other traditional methods. Obviously, the depth of CNN plays an important role in the significant results, but its also harder for networks with tens of layers to converge because of vanishing gradients [2,4]. Many researchers' works about normalization [8,14] help to solve this obstacle. In 2014, researchers from the Visual Geometry Group of Oxford University used VggNet [15] to get better performance in ImageNet. According to [15], the 19-layer VggNet performs best, and we cannot get better accuracy by increasing more layers. Whats worse, stacking more layers may lead to degradation. To solve this problem, He et al. [5] proposed a residual network (ResNet) frame using residual shortcuts, show in Fig. 1. In [5], He compares CNN models with 50, 101, and 152 layers. The l52-layer network gets the state-of-the-art top-5 error accuracy of 3.57% in ImageNet. With the help of residual learning, it seems that we can stack conventional layers as more as we want to get better performance.

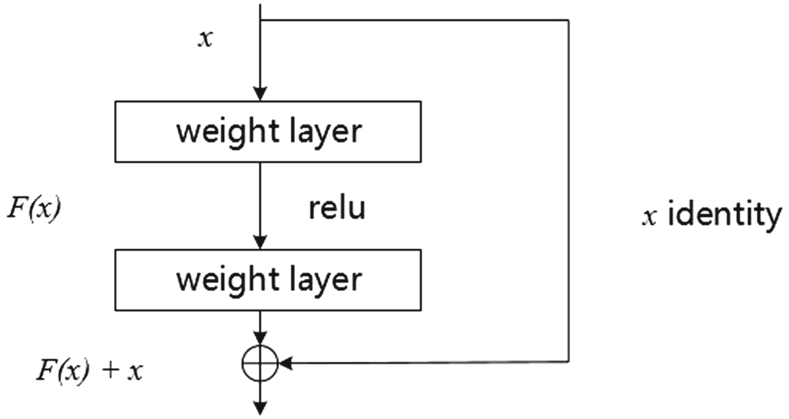

Fig. 1. A basic building block where every few stacked layers learn the residual function with feed-forward shortcut connection.

2.2 Going Smaller

As a result of deeper architecture, the size of CNN grows larger and larger. This means we need not only more expensive computing devices like GPU but also more time to train and deploy the CNN models. There are different ways to make CNN models smaller, such as pruning [3], binarized netural network [6], etc. This paper focuses on how to design the architecture more efficient at beginning. In 2014, Lin, et al. [13] proposed Network-in-Network structure using 1 × 1 filters to reduce computation. Szegedy et al. [17] created GoogLeNet family by using inception structure that contains 1 × 1, 3 × 3 filters layers, show in Fig. 2, to build more efficient and deeper models. Researchers from Berkley proposed SqueezeNet [7] using the cascaded structure that is named with "Fire" layer based on inception. SqueezeNet achieves the same accuracy of AlexNet with the model size squeezed to only 4.8M.

Motived by the development of CNN architecture, we try to find out which architecture can perform best in the Plankton Dataset, so that we can deploy the model in seafloor observation networks to deal with tons of data collected by undersea cameras. We need a model with not only high classification accuracy but also high speed.

3 The Hidden Degeneration

As mentioned above, the past few years have witnessed the break-throughs created by multiple classic CNN architectures with more and more convolutional layers. Before we introduce our final model, we show some experiments we did on the Plankton Dataset. In order to find out how deeper CNN models affect the accuracy in the Plankton Dataset, the models we have tested include CaffeNet[2],

[2] https://github.com/BVLC/caffe/tree/master/models/bvlc_reference_caffenet.

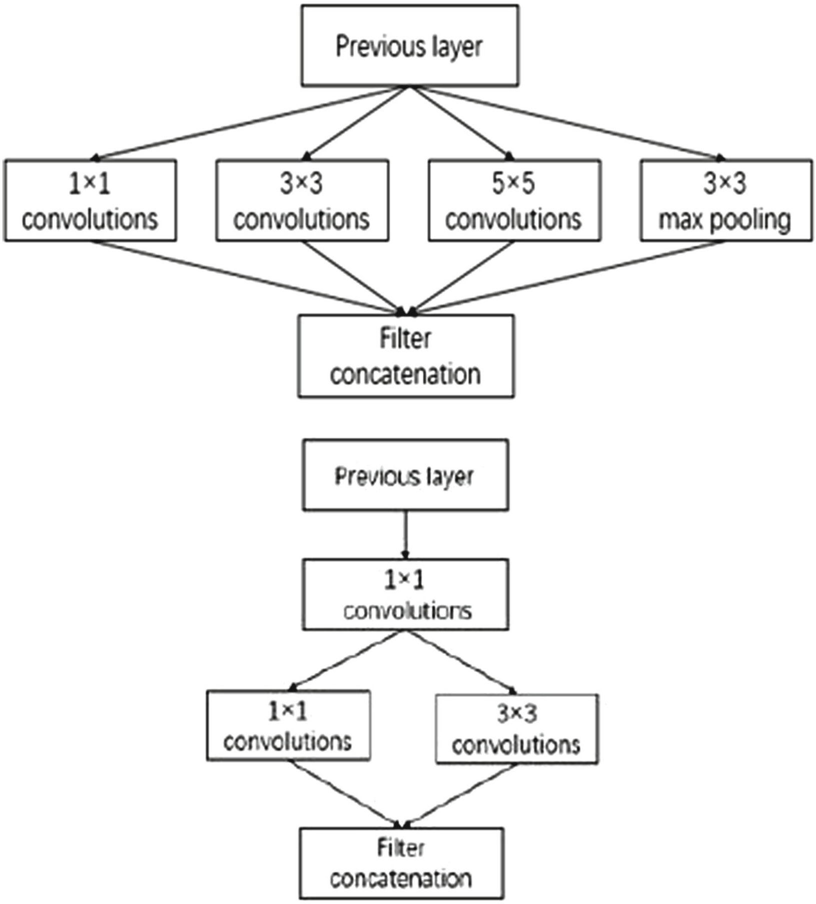

Fig. 2. A typical inception layer structure (up) contains convolutional layers with filters of different size, and the "Fire" layer of SqueezeNet (down) are designed with similar micro structure.

VggNet-19 (Vgg-19) [15], ResNet [5] with different layer numbers of 19, 50, 101. At first we held the view that advanced model with more layers may performed better, but we discovered that the basic CaffeNet has a quite good result and more layers may not only lead to the loss on Top-1 accuracy but also cannot help to improve the Top-5 accuracy. We named this phenomenon as the hidden degeneration, details are described below.

3.1 Data Augmentation and Experiment Environment

The Plankton dataset consists of 30,336 grayscale training images of varying size, divided unevenly into 121 classes which correspond to different species of plankton. This dataset was used for the National Data Science Bowl, a data science competition hosted on the Kaggle platform. We divide the dataset into separate validation, testing and training set of 3,037, 3,037 and 24,262 images respectively and rescaled them to 256 × 256 based on the length of their longest side. We augmented the data to increase the size of the dataset which can be useful for the prevention of overfitting by rotating the original images 0°, 90°, 180°, 270°, show in Fig. 3.

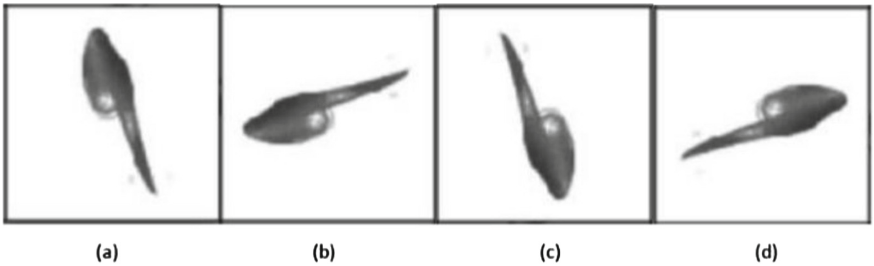

Fig. 3. Data augmentation by rotating target Plankton image 0° (a), 90° (b), 180° (c), and 270° (d).

We use Caffe [9] as experiment environment and implement different CNN models using one piece of GTX1080Ti GPU.

3.2 Multiple Models Comparison

The architecture of these models are shown in Fig. 4. ResNet has too many layers so we only show a single block.

CaffeNet is a replication of AlexNet with pooling layer before normalization by researchers from Berkley. It has 5 convolutional layers and 3 fully-connected layers. 11 × 11 shown in Fig. 4 means the size of filters and 96 means the number of kernels in the specific layer. Vgg-19 can be regarded as a AlexNet stacking more layers. ResNet is much different from other models with residual shortcuts and less fully-connected layers.

All the nets are trained with stochastic gradient descent (SGD) with back-propagation [11], having the same solver parameters with a learning rate of 0.001, a momentum of 0.9, and a weight decay of 0.0005.

We use the Plankton Dataset as train source directly at first, the results are shown in Table 1.

According to the results, we can find that unexpectedly although CaffeNet is not as deep as Vgg-19, Res-19, or Res-50, it has higher classification accuracy and speed with less storage space. The deepest model ResNet with 101 layers only makes little progress comparing to CaffeNet with a great setback in classification

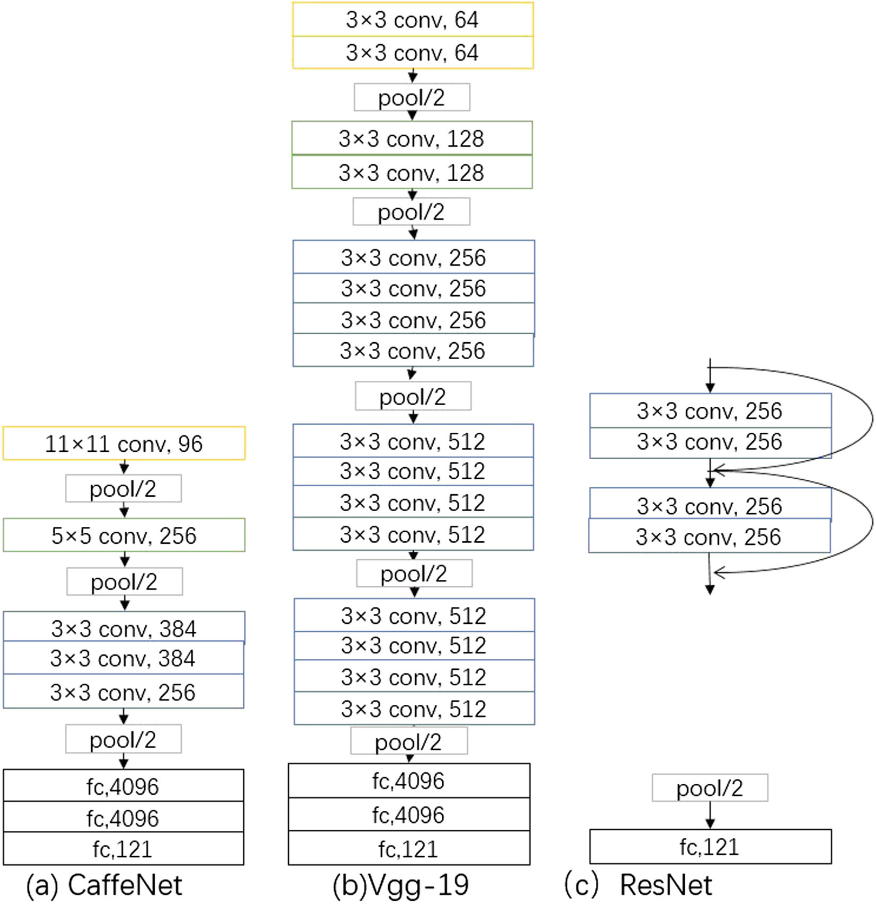

Fig. 4. There are mutiple CNN models used in the Plankton Dataset.

Table 1. Models performance on the Plankton Dataset

Model	CaffeNet	Vgg-19	Res-19	Res-50	Res-101
Top-1 accuracy (%)	74.5	60.6	70.7	72.9	74.7
Top-5 accuracy (%)	94.6	89.7	93.6	95.7	95.9
Model size (MB)	224.0	547.2	44.0	93.0	173.4
Frame rate (fps)	16.9	2.9	12.4	2.7	0.9

rate. As we clarified above, CaffeNet and Vgg-19 have nearly the same micro structure, while ResNet is much more different from them. It is shown that Vgg-19's result is worse than CaffeNet's with more convolutional layers. As for ResNet, the more convolutional layers there are, the higher the accuracy we get.

As a result, we cannot get final conclusion whether more conventional layers help to improve the classification accuracy for the Plankton Dataset by this experiment.

We hold the view that this is because the Plankton Dataset is a small scale dataset. As a result, CaffeNet has less parameters than other nets but can be trained better. To prove our inference, we design the research process in below.

The ILSVRC2012 dataset which contains 141 GB images of 1,000 classes is used to train all the nets instead of using the Plankton Dataset directly. After the models hit limits on ILSVRC2012, the weighs of previous layers in these nets that we finally saved are adopted to learn about the features of images in the Plankton Dataset. This method is called "finetune" in common. The results are shown in Table 2.

Table 2. Models performance after finetune

Model	CaffeNet	Vgg-19	Res-19	Res-50	Res-101
Top-1 accuracy (%)	77.7	74.1	78.5	76.2	76.8
Top-5 accuracy (%)	96.1	95.2	95.4	96.1	96.8

After finetune, we can see all the nets make significant progress than training on the Plankton Dataset directly, and the degeneration phenomenon that deeper models get lower accuracy is more obvious. We can see that Vgg-19 has a worse result but more layers than CaffeNet, comparing both Top-1 and Top-5 accuracy. Res-19 has the fewest layers but the highest Top-1 accuracy, while the Top-5 accuracy of different ResNet models are nearly the same.

The comparison experiment shows that the deeper and advanced models do not lead to better results on the Plankton Dataset. Residual connection does not make significant progress than CaffeNet in the experiment. And models with proper layers can help to get a better result.

Comprehensively speaking, the mass data collected by underwater camera calls for more efficient model which takes less storage space, runs more rapidly and keeps a proper accuracy. The new model should take the hidden degeneration into consideration.

4 Propose Framework

4.1 Key Points to Make CNN Smaller

In fact, when researchers try to make CNN architecture deeper with the limited storage of GPU, lots of tricks have been applied in different models to make them smaller. Here we conclude some key points on designing efficent architecture [15,17]:

First, use smaller filter instead of 5 × 5, 11 × 11 convolutional filter. Because a 5 × 5 convolutional layer has the same sense field as two cascaded 3 × 3 convolution-al layers, and the latter method reduce computation because $3 \times 3 + 3 \times 3 < 5 \times 5$. So we can find that Vgg-19 only uses 3 × 3 filter rather than AlexNet with 5 × 5, 11 × 11 filters.

Second, avoid adding fully-connected layers. Fully-connected layer has much more parameters than convolutional layer. One of the alternative methods is to use convolutional layer with global average pooling, but this can lead to loss of accuracy. This trick is also adopted by He in [5], but he adds one fully connected layer at last.

Third, using inception to enrich nonlinear representation of feature maps. The former two ideas are decreasing as more parameters as possible, but this will make accuracy decrease as well. Motived by SqueezeNet [7], we try to use more 1 × 1 inception structure to keep the accuracy with least cost.

Based on key points listed above, we proposed a more efficient framework for the Plankton Dataset.

4.2 Framework Architecture

We now introduce the CNN architecture we proposed in Fig. 5. The Model contains 14 convolutional layers in total. To squeeze the size, we do not use any fully-connected layers. All the convolution layers are designed with filters smaller than 3 × 3.

However, this may increase classification error. So 1 × 1 inception structure are adopted to maintain the accuracy.

According to the hidden degeneration we discover, the depth has an optimal value for the Plankton Dataset. We increase the net layer by layer to and finally choose the 14-layer model. Some of experiments are shown in Table 3.

Table 3. The framework with different layer numbers

Model	12 layers	14 layers	16 layers	18 layers
Top-1 accuracy (%)	74.6	76.4	76.3	76.0
Top-5 accuracy (%)	94.8	96.2	96.0	96.1

As mentioned above, SqueezeNet [7] proposed by Berkley researchers also use the 1 × 1 inception structure and they call two cascaded layers as Fire module, but they also use late downsample and different convolutional filter size compar-ing with our architecture. We conclude the results of different models in Table 4.

Our architecture is only 1.5 MB, and can reach real time operation rate of 32.2 fps with nearly no loss in classification accuracy.

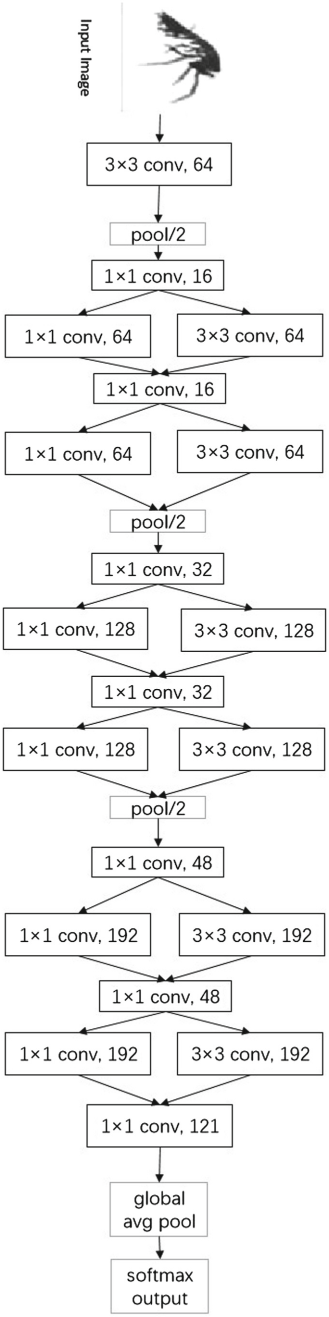

Fig. 5. There are mutiple CNN models used in the Plankton Dataset.

Table 4. Plankton classification comparision

Model	Ours	CaffeNet	Res-19	Res-50	Res-101	SqueezeNet
Top-1 accuracy (%)	76.4	77.7	78.5	76.2	76.8	74.9
Top-5 accuracy (%)	96.1	96.1	95.4	96.1	96.8	95.8
Model size (MB)	1.5	224.0	44.0	93.0	173.4	3.1
Frame rate (fps)	32.2	16.9	12.4	2.7	0.9	18.4

5 Conclusion

The development of CNN has been quite rapidly recently, however, for the specific problem such the Plankton Dataset, advanced models generlize not so well. Lots of work remains to be done for researchers.

In this paper, we compare multiple models to classify planktons. We find that the deeper and advanced models do not lead to better results on the Plankton Dataset. What's worse, deeper nets take up more storage space and run more slowly, so that cannot be actually deployed in the seafloor observation network. To solve this problem, we proposed a framework inspired by concluding the common skills to make CNN smaller, our architecture archive a real time process speed and nearly no loss of classification accuracy.

References

1. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: a large-scale hierarchical image database. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255 (2009)
2. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, pp. 249–256 (2010)
3. Han, S., Mao, H., Dally, W.J.: Deep compression: compressing deep neural networks with pruning, trained quantization and Huffman coding. In: International Conference on Learning Representation (2016)
4. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: surpassing human-level performance on imagenet classification. In: IEEE International Conference on Computer Vision, pp. 1026–1034 (2015)
5. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
6. Hubara, I., Courbariaux, M., Soudry, D., El-Yaniv, R., Bengio, Y.: Binarized neural networks. In: International Conference on Neural Information Processing Systems, pp. 4107–4115 (2016)
7. Iandola, F.N., Han, S., Moskewicz, M.W., Ashraf, K., Dally, W.J., Keutzer, K.: Squeezenet: alexnet-level accuracy with 50x fewer parameters and <0.5 MB model size. arXiv preprint arXiv:1602.07360 (2016)

8. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. In: International Conference on Machine Learning, pp. 448–456 (2015)
9. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: convolutional architecture for fast feature embedding. In: ACM International Conference on Multimedia, pp. 675–678 (2014)
10. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: International Conference on Neural Information Processing Systems, pp. 1097–1105 (2012)
11. LeCun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W., Jackel, L.D.: Backpropagation applied to handwritten zip code recognition. Neural Comput. **1**(4), 541–551 (1989)
12. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proc. IEEE **86**(11), 2278–2324 (1998)
13. Lin, M., Chen, Q., Yan, S.: Network in network. arXiv preprint arXiv:1312.4400 (2013)
14. Saxe, A.M., McClelland, J.L., Ganguli, S.: Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. arXiv preprint arXiv:1312.6120 (2013)
15. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: International Conference on Learning Representation (2015)
16. Srivastava, R.K., Greff, K., Schmidhuber, J.: Training very deep networks. In: International Conference on Neural Information Processing Systems (2015)
17. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–9 (2015)