Rethinking ReLU to Train Better CNNs

Gangming Zhao[1,4], Zhaoxiang Zhang[1,2,3,4*], He Guan[1,2,4], Peng Tang[6], Jingdong Wang[5]

[1]Research Center for Brain-inspired Intelligence, CASIA
[2]National Laboratory of Pattern Recognition, CASIA
[3]CAS Center for Excellence in Brain Science and Intelligence Technology
[4]University of Chinese Academy of Sciences
[5]Microsoft Research
[6]School of EIC, Huazhong University of Science and Technology

Abstract—Most of convolutional neural networks share the same characteristic: each convolutional layer is followed by a nonlinear activation layer where Rectified Linear Unit (ReLU) is the most widely used. In this paper, we argue that the designed structure with the equal ratio between these two layers may not be the best choice since it could result in the poor generalization ability. Thus, we try to investigate a more suitable method on using ReLU to explore the better network architectures. Specifically, we propose a proportional module to keep the ratio between convolution and ReLU amount to be N:M (N>M). The proportional module can be applied in almost all networks with no extra computational cost to improve the performance. Comprehensive experimental results indicate that the proposed method achieves better performance on different benchmarks with different network architectures, thus verify the superiority of our work.

I. INTRODUCTION

Nowadays, with the help of large scale image datasets, *e.g.,* ImageNet [1]) and high computing ability of GPU, deep Convolutional Neural Networks (CNNs) [2] have been dominant in many computer vision applications [3], especially in image classification [4–10].

One revolutionary contribution of deep learning system is to utilize Rectified Linear Unit (ReLU) [11] to replace the previous sigmoid and tanh function. Saturated activation functions like sigmoid and tanh lead to the "vanishing gradient", while ReLU prunes the negative values to zero, and retains the positive part that alleviates the vanishing gradient problem. Moreover, Glorot *et al.* [11] holds the view that ReLU boost the sparsity of the output feature which benefits the performance. Recently, to achieve better performance, most works focus on network architectures, such as deeper [6–8] or wider [12, 13] networks. Although there are different architectures, all these networks have the same characteristic: each convolutional layer is followed by an activation function, mostly ReLU. In the previous research, we empirically find that removing some ReLUs in deep CNNs can further improve performance. Thus, we argue that the paired module with the 1:1 ratio between convolution and ReLU may not be the best choice to design network architectures. In this paper, we first propose a proportional module to investigate the more suitable convolution and ReLU ratio, which can assist researchers to design the better network architectures. We call the module with equal ratio between convolution and ReLU module as
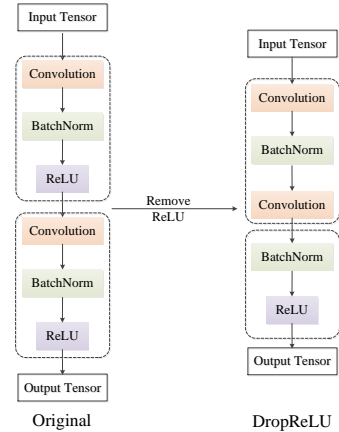

Fig. 1. The left Figure shows a part of plain networks, which is composed of the conventional paired module, the right Figure shows the proportional module, which removes the first ReLU layer.

paired module. Correspondingly, we call the N:M ratio between convolution and ReLU module as proportional module. Paired module is only composed of the non-linear unit. But the linear unit can help different networks expressing the distribution of data better since it use information effectively. For example, the feature tensors after the linear unit can recover the information discarded by ReLUs, which is benefit to the whole network. As shown in Figure 1, our approach removes the first ReLU layer of the basic module. We call this modified module as 2-1 proportional module.

The contribution of this paper includes three parts:

- We first propose the proportional module and apply it to CNN models.
- Numerous experimental results verify the reasonableness of our assumption. The proportional module can further improve the performance of the specific network architectures.
- The proportional module can be applied in almost all networks with no extra computational cost. This simple method can help researchers to design the better network architectures.

II. RELATED WORK

With the rapid development of deep learning, many efforts have been made to improve the learning algorithms for CNN, such as ReLU for improving feature representation of model. Dropout [14] and Batch Normalization [15] are useful for both performance and efficiency improvement. In addition, GoogLeNet [16], ResNet [17] [10], Deep Fusion Network (DFN) [9], Xception [18], ResNeXt [19] and IGCNets [20] have fully verified that it is valuable to effectively train a deeper or wider network with better performance. Furthermore, many researches like [21, 22] explore the inner mechanism and more fundamental theory of deep learning, which effectively improve the interpretability of deep learning.

A. Rectifier Units

Among these advancements, ReLU is one of several factors to the success of deep learning. Despite the great performance improvement, there have still been many recent improvements of activation functions containing leaky rectified linear (LReLU) [23], randomized leaky rectified linear (RReLU) [23] and parametric rectified linear (PReLU) [24], which are useful to the optimization of network so as to improve the performance. In addition, Shang *et al.* [25] proposed concatenated rectified linear (CReLU) to alleviate the problem of negative correction in convolutional layers. Besides, Zhu et al. [26] introduced a compositional activation function for DCNN. Furthermore, exponential linear units (ELUs) [26] were proposed to speed up learning in deep neural networks and acquire higher classification accuracies, but ELUs increased the computational complexity of networks. In contrast with ELUs, our proposed method can effectively improve the performance of various CNN structures with no extra computational cost.

B. Batch Normalization

Batch Normalization (BN) was proposed by Sergey *et al.* [15] to alleviate the internal covariate shift problem. As a regularizer, Dropout is gradually replaced by BN. Currently, BN is widely applied in almost all CNN models.

BN allows researchers to create deeper and wider networks, and it is one of important keys to the great success of ResNet. Generally, networks are composed of the sequence of CONV-BN-ReLU. Also, the sequence of BN-ReLU-CONV can further improve the performance of some networks like ResNet. Proportional module also can be considered as a regularization method to alleviate the overfitting of models. It explores more flexibly combinational methods of CONV, BN and ReLU. For example, the sequence of CONV-BN-CONV-BN-ReLU is a 2:1 proportional module, it effectively improves the performance of networks in our experiments. But whether is the combinational method the best module in CNN models. We need to do more experiments to explore the influence of different proportional module on networks.

In addition, BN have great influence on proportional module. For example, if we remove the first ReLU and BN in building block of ResNet [17], the two convolutional layers
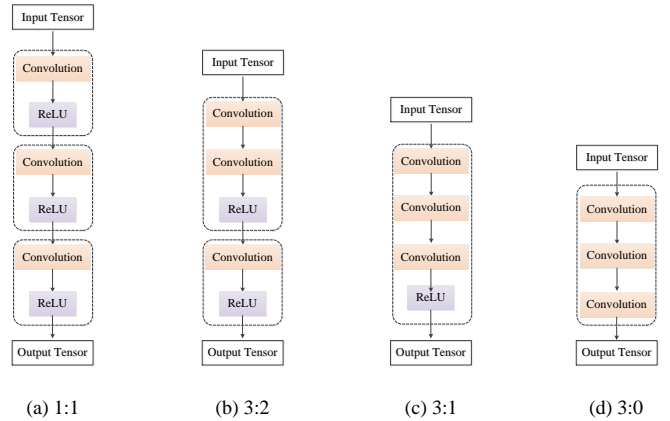

Fig. 2. (a): The paired module. (b), (c): The different proportional modules. (d): If removing all ReLUs, the model becomes a linear module, it is an extreme example and it can not work because of without non-linear function.

become a convolution with larger receptive field. Similarly, if removing the last ReLU and BN, pre-activation building block is also the same. The two convolutional layers do not become one convolution because the existence of the first BN. For another case, if we remove the first ReLU and BN of pre-activation building block or the last ReLU and BN of building block, the phenomenon of the first case will not appear, whether or not is the first BN of pre-activation building block or the last BN of building block redundant. We still need to conduct more experiments and theoretical analysis to explore the influence of this BN.

III. OUR APPROACH

A. Proportional Module

The current popular networks consist of multiple stacked paired modules. Paired module keeps the ratio between convolution and ReLU amount to be 1:1. We find that the conventional module could result in network overfitting and poor generalization ability. We empirically find that reducing the number of ReLUs can be helpful for ameliorating the overfitting in CNN. In our paper, we adopt a fire-new strategy to regularize CNN by utilizing the proposed proportional module. Proportional module integrates the linear and non-linear information into models to form richer representations of data. Furthermore, He *et al.* [17] [10] proposed ResNet which can be considered as another method to combine two types of information. We apply proportional module in ResNet, which can further improve the classification performance. In a word, proportional module can not only act as a regularizer but also as a method which can reduce the information loss in forward and backward prorogation process using ReLU. Figure 2 shows some different proportional modules and paired module.

B. Combinatorial Method for Diverse Modules

In this section, we will introduce the different combinatorial method of diverse modules in networks including plain network in Figure 1, ResNet in Figure 3, Pre-ResNet in Figure 4
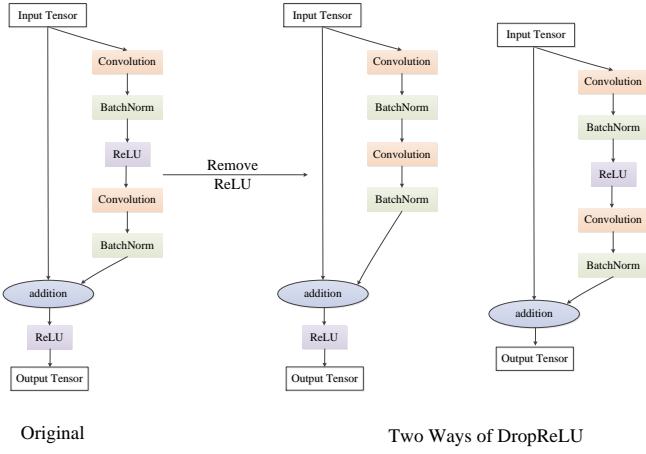
Fig. 3. The building block of ResNet vs. two ways of applying proportional module in building block.

and Deep Fusion Network in Figure 5. Paired module has two forms: (1) a convolutional layer is followed a ReLU that we call this module as Post-Paired Module or Paired Module and (2) a convolutional layer is placed in the after of ReLU that we call this module as Pre-Paired Module. Also, there are two identical forms in proportional module. We apply these four modules in plain network, ResNet and Deep Fusion Network (DFN), respectively.

The building block is the first residual module proposed by He *et al.* [17]. These blocks are shown in Figure 3 and 4. For building block, the method of applying proportional module has two styles that shown in Figure 3.
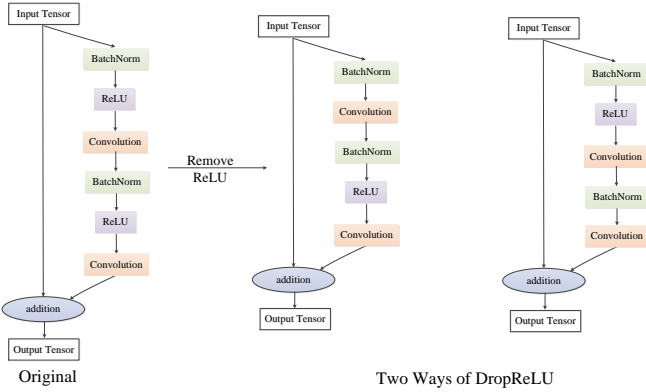

Fig. 4. The pre-activation building block of ResNet vs. two ways of applying proportional module in the pre-activation building block.

Also, for the pre-activation building block, there are two styles of applying proportional module. We use the pre-activation building block to conduct experiments on ResNet because we find that removing ReLU in building block has the similar results in pre-activation building block. In experimental section, we only show the results of removing the first ReLU on pre-activation building block since we find that removing

the first ReLU of pre-activation block achieve the higher performance than removing the second one. But why this result occurs, we also find the similar results in the pre-activation bottleneck block. The detailed analysis is shown in experimental section.

Deep Fusion Network (DFN) [9] is proposed by Zhao *et al.*, who introduced a merge and run fusion scheme to combine two sub-networks, which are all called as deeply merge-and-run fused network (DFN-MR). DFN-MR performs superiorly to existing popular networks like ResNet. In our experiments, we utilize proportional module to replace paired module. There are many methods to replace paired module. We list one replaced method shown in Figure 5.
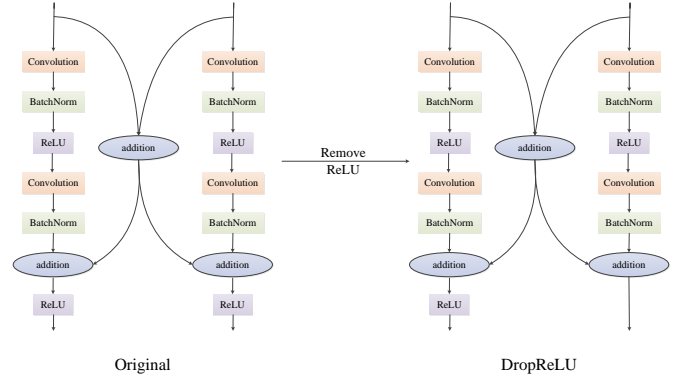

Fig. 5. The merge and run block of DFN vs. two ways of applying proportional module in the merge and run block .

IV. EXPERIMENTS

In this section, we will introduce our experiments on plain network, ResNet [8, 10] and Deep Fusion Network (DFN) [9]. In the first section, we will briefly describe the datasets. Then in next section, we explore the influence of our method by applying proportional module in plain network. In addition, we adopt two popular networks including DFN and ResNet to combine paired module and proportional module for further exploring the better performance. All the tables show the average classification accuracy from 5 runs and the standard deviation (mean $\pm$ std).

A. Datasets

In our experiments, we use two datasets that are commonly used for image recognition: CIFAR-10 and CIFAR-100. All the two datasets are detailed as follows.

CIFAR-10: The CIFAR-10 dataset has 50,000 images for training and 10,000 for testing. All images are tiny RGB images with a size of 32×32.

CIFAR-100: The CIFAR-100 dataset is similar to the CIFAR-10, except that it has 100 classes.

B. The Various Networks

The setting for the different networks are as follows:

Model	Depth	Accuracy
Paired Module	38	64.04 ± 0.14
2-1 Proportional Module	38	64.25 ± 1.44
Paired Module	62	56.88 ± 1.16
2-1 Proportional Module	62	**60.86 ± 1.18**
Paired Module	84	44.16 ± 0.57
2-1 Proportional Module	84	**55.98 ± 2.59**

We modify the plain network, which is introduced in [17], to fit CIFAR-10 and CIFAR-100. The detail about the three networks are introduced in [9]. Like [9], we use the SGD algorithm with the Nesterov momentum to train all the models for 400 epochs on CIFAR-10/CIFAR-100, both with a total mini-batch size 64. The detail of the experimental setting we used is also introduced in [9].

1) The Plain Network: In this section, we utilize plain network to verify that the N:M (N>M) convolution and ReLU ratio is indeed effective. We argue that although ReLUs alleviate the problem of vanishing gradient caused by sigmoid and tanh activation functions, the redundancy of ReLUs is an important reason why the accuracy of deeper network gets saturated. We conduct comparative experiment of the 2:1 convolution and ReLU ratio and 1:1 ratio to explore the effectiveness of proportional module. To show the ratio of convolution and ReLU, we call these modules Paired Module and 2:1 Proportional Module. In addition, 2:1 Proportional Module means we remove the last ReLU of the 1-th, 3-th, 5-th and etc.

Table I shows that our experimental results. He *et al.* [8] had verified that the degradation problem becomes more serious with the network depth increasing. From Table I, we can find that a proportional module with 2:1 convolution and ReLU ratio can alleviate degradation problem effectively. And 2:1 proportional module improves the performance of the conventional paired module by 3.98% and 11.82% in 62 and 84-layer plain networks.

2) ResNet: In this section, we use building block and pre-activation bottleneck block for exploring the influence of different convolution and ReLU ratio. Specifically, we adopt the various combinational method of modules to investigate whether N:M (N>M) convolution and ReLU module can help improving the generalization of networks. We have introduced these models in section *Combinatorial Method for Diverse*

Depth / C-R	1:1 Pre-Paired	2:1 Pre-Proportional
CIFAR-10		
62	93.59 ± 0.20	**93.97 ± 0.23**
CIFAR-100		
62	71.43 ± 0.32	**72.19 ± 0.18**
110	72.46 ± 0.10	**73.41 ± 0.10**
164	73.00 ± 0.31	**74.12 ± 0.22**

Modules. In this section, we will introduce the experimental results of these modules in different residual modules.

Building Block: Table II shows the results of pre-activation building block of ResNet on CIFAR-10 and CIFAR-100. Among of Table II, C-R means the convolution and ReLU ratio. 1:1 Pre-Paired Module and 2:1 Pre-Proportional Module mean two different ratios in pre-activation building block. In addition, for 2-1 Pre-Proportional Module, we remove the first ReLU in pre-activation block. For 62-layer ResNet, 2:1 Pre-Proportional Module achieves 93.97 recognition rate that is higher than the corresponding baseline on CIFAR-10. For CIFAR-100, this strategy also achieves the better performance in 62-layer ResNet. Then we set the depth of network as 110 and 164, respectively. We find that our method improves the recognition rate by 0.95 and 1.12 on CIFAR-100. In addition, Figure 6 shows the test recognition rates of pre-activation building block in 110-layer ResNet with 1:1 and 2:1 convolution and ReLU ratio as a function of number of epochs on CIFAR-10. As shown in Figure 6, we can find that 2:1 Pre-Proportional Module can further improve the performance of traditional 110-layer ResNet.

Pre-activation Bottleneck Block: Figure 7 shows the test recognition rates of pre-activation bottleneck block in 110-layer ResNet with 1:1 and 3:2 convolution and ReLU ratio as a function of number of epochs on CIFAR-100. Among of Figure 7, Bottleneck Pre-Paired Module means the conventional pre-activation bottleneck block and Bottleneck Pre-Proportional Module means we remove one of three ReLUs. So, it has three types, Type 1, 2, 3 correspond to remove the first, second and third ReLUs, respectively. As shown in Figure 7, we can find that Type 1, 2 of Bottleneck Pre-Proportional Module improve the performance of baseline and experimental results show that removing the first ReLU yields the best accuracy. But the performance of Type 3 Bottleneck Pre-Proportional Module is reduced.

In addition, as shown in Figure 8, we can find that although the experiments in 164-layer ResNet have the similar results compared with 110-layer ResNet, removing the third ReLU also improves the accuracy of baseline. Furthermore, we find the improvements of 164-layer ResNet with 3:2 convolution
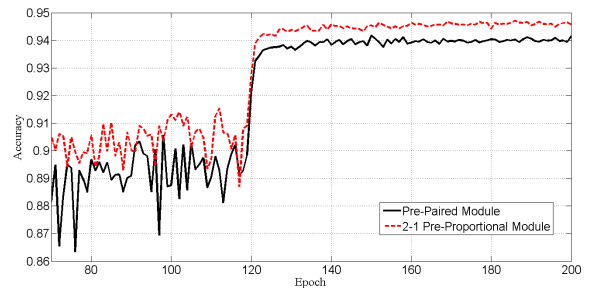

Fig. 6. The accuracy of 110-layer ResNet with pre-paired module and pre-proportional module on CIFAR-10. Pre-proportional module means we remove the first ReLU in pre-activation building block.
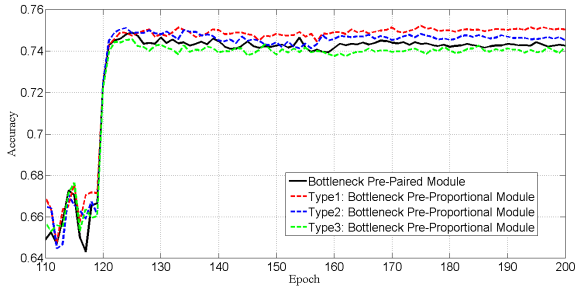
Fig. 7. The accuracy of 110-layer ResNet with bottleneck pre-paired nodule (baseline) and bottleneck pre-proportional module on CIFAR-100. Type 1, 2 , and 3 denote Removing the first, second, and third ReLU in pre-activation bottleneck block, respectively.

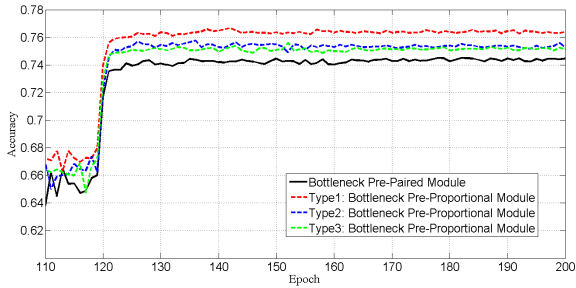Fig. 8. The accuracy of 164-layer ResNet with bottleneck pre-paired module (baseline) and bottleneck pre-proportional module on CIFAR-100. Type 1, 2 , and 3 denote Removing the first, second, and third ReLU in pre-activation bottleneck block, respectively.

and ReLU ratio are higher than the improvements of 110-layer ResNet.

3) Deep Fusion Network: In this section, we utilize the merge and run block to explore the influence of different convolution and ReLU ratio on Deep Fusion Network. The module we used is shown in Figure 5. Among of it, Paired Module means we adopt the 1:1 convolution and ReLU ratio in DFN-MR1 model. Type1 Proportional Module means we remove the ReLU after element-wise in one branch, Type 2 Proportional Module means we remove the ReLU before element wise in one branch. We find that the Type 1 Proportional Module achieves the better performance on CIFAR-10 and CIFAR-100, but the improvement is not higher enough. Type 2 Proportional Module cannot improve the performance compared with the baseline

V. DISCUSSION

In previous research, we empirically find that removing some ReLU layers in CNN models can achieve better performance. Thus, in our work, we investigate the more flexible proportional module to design the better network architectures.

A. The redundancy of ReLUs in Plain Network

First, we think the redundancy of ReLUs is one of important reasons why the accuracy of deeper network gets saturated. In

Model	Depth	Accuracy
CIFAR-100		
Paired Module [9]	110	75.59 ± 0.19
Type1 Proportional Module	110	**75.61 ± 0.38**
Type2 Proportional Module	110	74.53 ± 0.80
CIFAR-10		
Paired Module [9]	110	95.04 ± 0.06
Type1 Proportional Module	110	**95.26 ± 0.14**

plain network, we find that removing the half of ReLUs can great improve the classification performance. It verifies that the N:M (N>M) convolution and ReLU ratio is indeed effective, especially in deeper networks. Therefore, 2:1 convolution and ReLU ratio may be a more suitable proportional module, but it is a question whether the module can improve the performance of other more popular networks like ResNet.

B. Comparison with ResNet and DFN Baseline

Second, to validate the generalization of our method, we utilize ResNet and DFN as baseline to explore the influence of the proportional module. From our experimental results, we find that the proportional module can help the building block, pre-activation bottleneck block and merge and run block achieve the better performance. Experimental results show that the proportional module not only alleviates the degradation problem in plain network but also can further improve the performance of ResNet and DFN. It demonstrates the effectiveness of proportional module in regularizing CNNs.

C. To Further Explore the Influence of ReLU

Third, from coarse to fine, we explore the influence of Re-LUs in different positions on some blocks. For pre-activation bottleneck block, we find some interesting phenomena: With the depth increasing, the influence of single ReLU in one block is reduced. On the other hand, the influence of last ReLU in a pre-activation bottleneck block is the most important. For the first phenomenon, we think that with the depth increasing, the redundancy of ReLU becomes serious so that the influence of single ReLU is reduced. For the second phenomenon, we think that there are two reasons: The one reason is that the first ReLU leads to information loss, which inhibits the learning effect of the first convolutional layer. The other one is that two convolutional layers before the last ReLU can learn a better transformation of data. When this transformation of data was through ReLU, the whole network can acquire the better non-linear information. In addition, from our experimental results shown on Table III, we can find that the first ReLU in the merge and run block is more important than the ReLU after element-wise. We think the first ReLU has an important effect on the representation ability of the block because the first ReLU controls the non-linearity of the whole block.

At last, we think that the reason why the proposed method works is two-fold. First, the redundancy of ReLUs leads to the accuracy of deeper network gets saturated. Second,

the proportional module utilizes information more effectively since it can recover the part of information discarded by ReLU.

VI. CONCLUSION

In this work, we argue that the conventional paired module may not be the best choice to design the network architectures. We found that proportional module can help network achieve better performance. From our experimental results, we found that proportional module could enrich the expressive power of the network. Although the non-linearity preserves the depth property of a network, the linear information cannot still be ignored for learning a better distribution of data since it use information effectively. Experimental results show that such a simple design has superior performance than the corresponding baselines while having the same computation cost compared to the original models. To our best knowledge, this is the first work to introduce the proportional module in CNN models. For future work, we will utilize reinforcement learning to assemble adaptively the convolutional layer and ReLU.

REFERENCES

[1] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *IJCV*, vol. 115, no. 3, pp. 211–252, 2015.

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

[3] G. Zhu, J. Wang, P. Wang, Y. Wu, and H. Lu, "Feature distilled tracking," *IEEE Transactions on Cybernetics*, pp. 1–13, 2017.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012, pp. 1097–1105.

[5] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," in *BMVC*, 2014.

[6] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *CVPR*, 2015, pp. 1–9.

[7] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *ICLR*, 2015.

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.

[9] L. Zhao, J. Wang, X. Li, Z. Tu, and W. Zeng, "On the connection of deep fusion to ensembling," *arXiv preprint arXiv:1611.07718*, 2016.

[10] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *ECCV*, 2016, pp. 630–645.

[11] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks." 2011.

[12] S. Zagoruyko and N. Komodakis, "Wide residual networks," in *BMVC*, 2016.

[13] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *ECCV*, 2014, pp. 818–833.

[14] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[15] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning*, 2015, pp. 448–456.

[16] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[18] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," *arXiv preprint arXiv:1610.02357*, 2016.

[19] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," *arXiv preprint arXiv:1611.05431*, 2016.

[20] T. Zhang, G.-J. Qi, B. Xiao, and J. Wang, "Interleaved group convolutions for deep neural networks," 2017.

[21] R. Stewart and S. Ermon, "Label-free supervision of neural networks with physics and domain knowledge."

[22] P.-L. Bacon, J. Harb, and D. Precup, "The option-critic architecture." 2017.

[23] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models."

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

[25] W. Shang, "Understanding and improving convolutional neural networks via concatenated rectified linear units," 2016.

[26] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (elus)," *arXiv preprint arXiv:1511.07289*, 2015.