Improving Backbones Performance by Complex Architectures

Jinxin Shao[1], Yutao Hu[3], Zhen Liu[1], Teli Ma[1], and Baochang Zhang[1,2](✉)

[1] School of Automation Science and Electrical Engineering, Beihang University,
Beijing 100191, People's Republic of China
{shaojinxin,liuzhenbuaa,mtl9868,bczhang}@buaa.edu.cn
[2] Shenzhen Academy of Aerospace Technology, Shenzhen 518057, Guangdong,
People's Republic of China
[3] School of Electronic and Information Engineering, Beihang University, Beijing 100191,
People's Republic of China
huyutao@buaa.edu.cn

Abstract. Recently, Convolution Neural Networks (CNNs) have achieved great success in computer vision. To further boost the performance, the depth of the backbone network is continuously increased, which improves the capacity of feature learning but also brings the heavy burden in computation. To address the issues, this paper introduces a complex convolution method to systematically improve the performance of the backbone network. Our contributions are three-fold: 1) the complex architecture backbone network can improve the classification performance without increasing or even reducing the number of parameters; 2) for the detection task, the complex architecture backbone network can improve the ability of feature map extraction, at the same time our joint bounding box generation method using both real and imaginary parts of complex features can obviously improve the object detection ability. 3) the proposed method has a strong generalization ability for both detection and classification tasks. We have achieved significant performance improvements in both classification and detection tasks, which validate the effectiveness of our methods.

Keywords: Complex architectures · Backbones performance · Complex feature map

1 Introduction

Backbone design is significant in the field of computer vision, especially for classification and detection tasks. In recent years, with the development of machine learning technology, the main methods of classification and object detection has changed from a feature-based method to a convolution neural networks (CNNs) based method [1, 2]. For the detection and classification tasks, both of them need to use a suitable and efficient backbone network to extract feature maps from the input image. Classification tasks often use fully connected layers to deal with the feature map and then calculate loss. For detection tasks, it is necessary to use information of feature maps and labels directly

Y. Peng et al. (Eds.): PRCV 2020, LNCS 12306, pp. 394–406, 2020.
https://doi.org/10.1007/978-3-030-60639-8_33

to calculate loss. For existing backbone networks, the direction of improvement could be concluded in two aspects: improving accuracy and saving parameters. To improve the network accuracy, the depth of the network is continuously increased, but for deep networks with more than 20 layers, there will be obvious degradation [3]. To address this issue, He et al. proposed residual neural network [4]. Inspired by the idea of residual learning, the use of identity mapping not only alleviates the problems of gradient explosion and gradient disappearance caused by the increase in network depth, but also avoids the degradation of the network and enables the network depth to reach thousands layers. The representative networks proposed under this idea are ResNet [5], ResNeXt [6] and Res2Net [7]; To meet the needs of the booming edge computing technology [8], small backbone networks with fewer layers have also been proposed. They can save lot parameters by reducing the number of convolution layers. The lightweight networks under this idea include MobileNet [9], ShuffleNet [10] and SqueezeNet [11]. Meanwhile, the pre-trained model of the backbone network of the classification task can be used for the detection task to improve the performance of the detection task. Therefore, it is of great importance to design a structure to balance number of parameters and prediction accuracy. In a summary, how to better improve the performance of the backbone network, that is, based on fewer backbone network parameters to obtain better-performing classification and detection results has very important research significance.

In this paper, we utilize complex structure to improve backbone network performance. Based on the existing complex convolution, complex batch normalization, complex ReLU and, complex weight initialization strategy [12], we follow the line of these algorithms and propose complex down sampling, complex dropout, etc. Using these complex architectures, we transform several backbone networks into complex networks, and proposed several methods of combining complex feature maps to evaluate the classification accuracy of the complex backbone networks. For the detection task, we use the YOLOv3 model to test the efficacy of the complex backbone network based on the VOC dataset. The backbone network we tested is not limited to darknet53: to test whether this change is effective for a wide range of backbone networks, we deleted 15 layers of residual blocks of darknet-53, that is, 30 convolution layers, showing that the algorithm has a lifting effect on existing backbone networks.

The main contributions of this work are as follows:

1) We show the employment of complex convolution backbone networks can improve the classification performance without increasing the amount of parameters, based on our effective combination of real and imaginary feature maps.
2) Extensive experiments demonstrate that the use of real and imaginary feature maps in the same framework can improve the detection accuracy.

2 Complex Convolution Neural Networks

Since complex numerical operations are mostly used in the field of signal analysis, most complex neural networks are applied to the speech signals for enhancing the phase information or predict spectrum. Trabelsi [12], which originally integrated a complex neural network, utilized a complex neural network to test the music transcription of the MusicNet dataset and the speech spectrum prediction and achieved good results. Choi [13] proposed the Deep Complex U-Net model for evaluation on a mixture of Voice Bank corpus and DEMAND database, which has been widely used by many deep learning models for speech enhancement. Pfeifenberger [14] estimates the complex weights by using the full potential of complex-valued LSTM, MLP, and directly obtains beamforming weights from complex-valued microphone array data. A complex-valued deep neural network for speech enhancement and source separation is proposed. It can be seen that most of the improvement work of complex neural networks is applied to speech signal processing, and this work attempts to use it in visual tasks. The composition principle of the complex neural network is almost the same, as shown below.

In this network, after the initialization of complex values, real and imaginary parts of the complex numbers are treated as logically different real-valued entities. By this way we can use real-valued algorithms to simulate complex number operations internally.

Note that the real part of the complex convolution kernel matrix is W_{real}, the imaginary part is W_{imag}, the real part of the input image vector is written as x_{real}, and the imaginary part is written as x_{imag}. In particular, the imaginary part here is represented by real numbers. In the convolution operation, the formula is written as follows:

$$\begin{aligned}(W * x)_{real} &= W_{real} * x_{real} - W_{imag} * x_{imag}\\ (W * x)_{imag} &= W_{imag} * x_{real} + W_{real} * x_{imag}\end{aligned} \tag{1}$$

The '*' represents a two-dimensional real convolution operation. Expressed in matrix form as:

$$\begin{bmatrix}(W * x)_{real}\\ (W * x)_{imag}\end{bmatrix} = \begin{bmatrix}W_{real} & -W_{imag}\\ W_{imag} & W_{real}\end{bmatrix} * \begin{bmatrix}x_{real}\\ x_{imag}\end{bmatrix} \tag{2}$$

2.1 Complex Batch Normalization

For batch normalization of real data, only one-dimensional data needs to be converted into a normal distribution [15]. For complex data, real and imaginary part may have different variances, which will bring bias into the data. Therefore, we treat it as the two-dimensional data, and use the covariance matrix V to normalize the eccentricity of it. As shown in the Eq. (3), $x - E[x]$ refers to the deviation of the two-dimensional data from the center.

$$\tilde{x} = (V)^{-\frac{1}{2}}(x - E[x]) \tag{3}$$

Where the covariance matrix V ψ is denoted as:

$$V = \begin{pmatrix}V_{rr} & V_{ri}\\ V_{ir} & V_{ii}\end{pmatrix} = \begin{pmatrix}\mathrm{Cov}(x_{real}, x_{real}) & \mathrm{Cov}(x_{real}, x_{img})\\ \mathrm{Cov}(x_{img}, x_{real}) & \mathrm{Cov}(x_{img}, x_{img})\end{pmatrix} \tag{4}$$

Where V needs to satisfy the condition of positive semi-definite matrix to make the inverse matrix of V in the above formula be solvable. After mathematical derivation, the conditions to be met are $V_{rr} + V_{ii} = 1$, $V_{ri} = V_{ir} = 0$.Similarly, imitating the batch normalization formula of the real-value network, the input complex values are scaled and translated as follow:

$$BN(\tilde{x}) = \gamma \tilde{x} + \beta,\ \gamma = \begin{pmatrix} \gamma_{rr} & \gamma_{ri} \\ \gamma_{ir} & \gamma_{ii} \end{pmatrix} \tag{5}$$

where γ_{rr} and γ_{ii} are initialized to $\frac{1}{\sqrt{2}}$, γ_{ir}, γ_{ri} and β are initialized to 0.

2.2 Complex ReLU and Other Functions

The ReLU involved in our proposed module is a complex ReLU, which is also called the CReLU. It is a separate ReLU activation applied to both the real and imaginary parts of the neuron, defined as:

$$\text{CReLU}(x) = \text{ReLU}(x_{real}) + i\text{RELU}(x_{imag}) \tag{6}$$

When the real and the imaginary part are the same sign, that is, when the input complex number is in the first or third quadrant, the formula satisfies the Cauchy-Riemann equation obviously. A series of other complex methods also adopt this idea, first divide the real and imaginary part, then treat them as independent real data, such as complex pooling, complex sigmoid.

3 Methodology for Using Complex Structure in Object Classification and Object Detection

For classification and detection tasks, the final prediction depends on the feature map of backbone network. In the classification tasks, the final feature map is a one-dimensional vector. While in the detection tasks, a high-dimensional tensor is often used. Therefore, it can be said that the network for classification task is composed of the backbone network and classifier. While the network for detection task is composed of the backbone network and the object detection business part [16]. Therefore, for classification tasks, it is only necessary to design some simple rules to combine complex-valued low-dimensional feature maps, while for detection tasks, it is necessary to flexibly design the application of complex feature maps according to the characteristics of object detection business part.

Therefore, to produce one-dimensional feature map, we designed four functions to combine the obtained one-dimensional complex feature map. They are called magnitude, signed-magnitude, summation and absoluted-summation respectively. We also try to directly convert complex feature maps into real feature maps through convolution, and the experimental results show all of them improve the performance of backbone network. In the detection task, we use the complex feature map's real part, the combined feature map of the real and imaginary parts, and the fully complex feature map of both real and imaginary parts combined with non-maximum suppression method to improve the detection accuracy. These methods gain improved detection results by making better use of the information of complex feature maps.

3.1 Complex Feature Map Combination Method

Image classification is a basic problem in image understanding. There are lots of data sets for evaluating image classification effects, such as CIFAR-10/100 [17], Caltech-101/256 [18] and ImageNet [19]. With the great success of the AlexNet [20] deep convolutional neural networks based methods have begun to replace traditional hand-crafted algorithms, and a series of effective backbone networks have been proposed. Based on AlexNet, some improved backbone, such as DenseNet [21], GoogleNet [22], ResNet [4], VGG [23], SENet [24] and ShuffleNet [10] have been proposed and achieved great success. Some classification networks that combine CNN with traditional image processing methods, such as GCN [25], have also achieved good results.

In this work, we use several widely used backbones to verify the improved characteristics of complex convolution for classification in the CIFAR dataset. The basic VGG network structure is shown in Fig. 1; the other two backbone network improvement methods are similar to it. For the CIFAR task, a linearization layer with the width of 4096 is not required. Too many linearization layers is also the reason for the excessive network parameters. We found that removing these fully connected layers does not influence the performance of the network, but on the other hand, can reduce the number of parameters. Therefore, some experiments used this structure which removed a fully connected layer, as shown in the Fig. 1, the 1st chart.

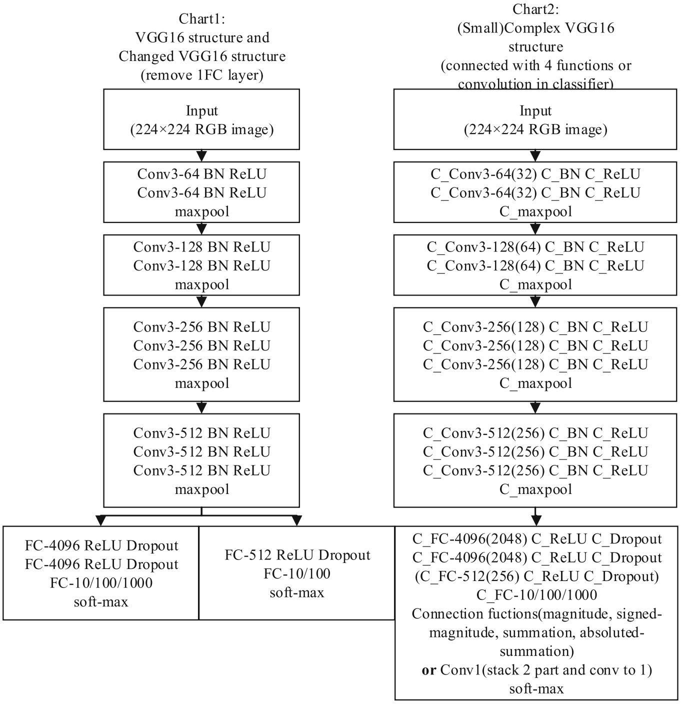
Chart1:
VGG16 structure and
Changed VGG16 structure
(remove 1FC layer)
Input
(224×224 RGB image)
Conv3-64 BN ReLU
Conv3-64 BN ReLU
maxpool
Conv3-128 BN ReLU
Conv3-128 BN ReLU
maxpool
Conv3-256 BN ReLU
Conv3-256 BN ReLU
Conv3-256 BN ReLU
maxpool
Conv3-512 BN ReLU
Conv3-512 BN ReLU
Conv3-512 BN ReLU
maxpool
FC-4096 ReLU Dropout
FC-4096 ReLU Dropout
FC-10/100/1000
soft-max
FC-512 ReLU Dropout
FC-10/100
soft-max
Chart2:
(Small)Complex VGG16
structure
(connected with 4 functions or
convolution in classifier)
Input
(224×224 RGB image)
C_Conv3-64(32) C_BN C_ReLU
C_Conv3-64(32) C_BN C_ReLU
C_maxpool
C_Conv3-128(64) C_BN C_ReLU
C_Conv3-128(64) C_BN C_ReLU
C_maxpool
C_Conv3-256(128) C_BN C_ReLU
C_Conv3-256(128) C_BN C_ReLU
C_Conv3-256(128) C_BN C_ReLU
C_maxpool
C_Conv3-512(256) C_BN C_ReLU
C_Conv3-512(256) C_BN C_ReLU
C_Conv3-512(256) C_BN C_ReLU
C_maxpool
C_FC-4096(2048) C_ReLU C_Dropout
C_FC-4096(2048) C_ReLU C_Dropout
(C_FC-512(256) C_ReLU C_Dropout)
C_FC-10/100/1000
Connection fuctions(magnitude, signed-magnitude, summation, absoluted-summation)
or Conv1(stack 2 part and conv to 1)
soft-max

Fig. 1. The scheme of complex architectures for VGG16 network

Based on this motivation, we use complex architectures on the backbone network, which will double its parameter amount (see related work for the method). For the classifier part, if the fully-connected layer is also complex, some methods need to be designed to combine two one-dimensional vectors. And to verify whether this structure can improve performance when the parameter amount is constant or even reduced, we deleted half the number of convolution kernels to perform the same experiment, as shown in Fig. 1, the 2nd chart.

We design five different methods to combine the complex vectors after the complex classifier, four are mathematical calculation methods, the other one uses 1 × 1 convolution. Magnitude method just treats two parts as mathematical complex numbers, like $v_{real} + v_{imaginary}i$, which $v_{real}, v_{imaginary}$ are 2 vectors on behalf of the input. The combined result is v_{output}. So this method just calculates the magnitude of it, which is written as:

$$\text{Magnitude:} v_{output} = (v_{real}^2 + v_{imaginary}^2)^{\frac{1}{2}} \tag{7}$$

Obviously it is always positive, no negative number will occur. It does not matter for the sigmoid function. But in more fuzzy work, this kind of feature map may cause problem because the feature map should be signed. Therefore, an improved method called signed-magnitude is proposed. The major difference is that the symbolic function is used to make the results keep the same sign with the real part, so it can be written as:

$$\text{Signed - Magnitude:} v_{output} = (v_{real}^2 + v_{imaginary}^2)^{\frac{1}{2}} \times \text{sgn}(v_{real}) \tag{8}$$

So inspired by this idea, we designed 2 more combine function. In these method both the real and imaginary part are treated as 1-dimension data. So if we want a signed output vector, which just calculate the summation, written as:

$$\text{Summation:} v_{output} = v_{real} + v_{imaginary} \tag{9}$$

If we want positive results, just calculate the sum of absolute value, which we called absoluted-summation, written as:

$$\text{Absoluted - Summation:} v_{output} = |v_{real}| + |v_{imaginary}| \tag{10}$$

Also 1-by-1 Convolution Layer is used to combine the 2 part into one, it just like another full connection layer. Since the input tensor is a one-dimensional vector, just the kernel size equals 1 may suit for this work. The formula is shown in Eq. (11).

$$\text{Conv1:use } 1 \times 1 \text{ convolution kernel to connect the real and imaginary part} \tag{11}$$

3.2 Joint Bounding Box Generation Method of Complex Feature Map

Detection task is a middle-level problem in the field of computer vision, that is, it needs to understand the foreground and background of the image. So far, object detection methods based on deep learning can be divided into 2 categories: two-stage detection methods and one-stage detection methods. The two-stage detection methods delineate the detection area first, and then determine whether there are targets in the selected area. R-CNN [26], fast R-CNN [27], faster R-CNN [28] and SPP-Net [29] are representative two-stage detection methods.. One-step detection method uses intensive sampling directly from the feature map to obtain the prior frame, and then do classification and regression on the prior frame. Obviously, one-stage method has a faster detection speed. Representatives of such method are YOLO [30], SSD [31], OD-GCN [32], and RON [33]. With the introduction of YOLOv4 [34], this type of method has achieved a balance between detection accuracy and efficiency, becoming the main trend of future research.

Therefore, we take YOLOv3 as an example to study the improvement effect of complex architectures on the backbone network in our work. To judge whether this method will improve performance under the condition of no pre-training model or any backbone network, we deleted the 15-layer residual connection layer of darknet-53, that is, the 30-layer convolution layer. So it can also be called “dark-23” as a comparison in the experiments. In the same way, we also conduct the complex architectures on original darknet-53 backbone network. Because of the lack of pre-trained model about the complex darknet-53 backbone network, we repeatedly assign the original real darknet-53 pre-trained model to the real and imaginary parts of the complex model. This pre-trained model may not be ideal. If there is a better pre-trained model, a better detection effect may be achieved. And the effectiveness of improvement can be judged by the comparison of different feature map using methods.

The output part of the backbone network is shown in Fig. 2. The specific structure of the darknet-53 and “darknet-23” backbone network can also be clearly seen from this figure. In this part, we use complex convolution, complex batch normalization, complex leaky ReLU and others to replace the real-valued function, which is also clear in the diagram. So we just make the input image in which real part equal to its real and imaginary part equal to zero. After this module, we will get 3 complex 3-dimensional tensors. If it is regarded as a real tensor, it can be regarded as 4-dimensional tensor. How to use this tensor for training and testing is shown in Fig. 3.

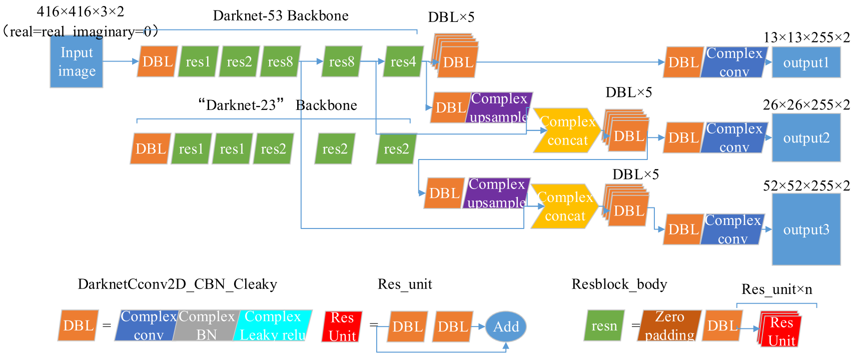

Fig. 2. The complex architectures for darknet-53 and “darknet-23”

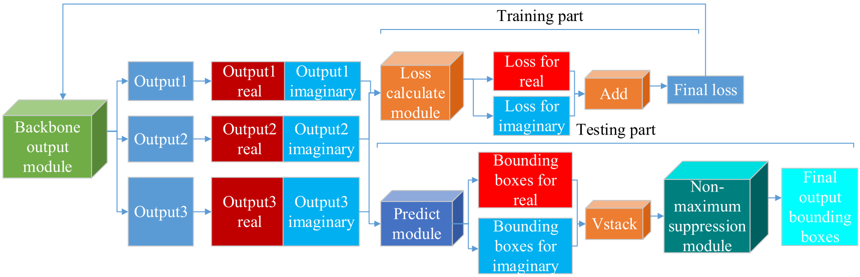

Fig. 3. The principle of training and testing using the output complex-value feature map

Here we conduct 3 experiments. First, we use the real part of the output complex feature map. Second, we use the signed-magnitude of the output complex feature map (see Eq. (8)). Last, we use both the real and imaginary parts of the output feature map for training and testing. For using real part of complex feature map and using signed-magnitude of complex feature map, these two methods actually convert the 4-dimensional output tensor into 3-dimensional tensor. Therefore, for subsequent processing parts, such as loss function calculation, or prediction box generation, there is no need to transform these parts. However, if we want to use both real and imaginary parts of complex tensor at the same time, it is equivalent to use two 3-dimensional feature maps. Here we treat the real and imaginary parts as two independent feature maps to calculate the loss function and generate the detection frame respectively. Then, the losses are combined together through summation. Therefore, the totally loss can be written as:

$$L_{final} = L_{real} + L_{imaginary} \tag{12}$$

According to the principle of back propagation, it can be derived and the training parameters can feedback on the backbone network automatically. During the test process, the real and imaginary parts will simultaneously generate bounding boxes from the prediction model. At this time, we put the prediction frames generated by the two parts in the same stack, and use the non-maximum suppression method to remove the repeated prediction frames to obtain the most comprehensive result. Therefore, the final output of bounding boxes can be written as:

$$bboxes_{output} = bboxes_{real} \cup bboxes_{imaginary} \tag{13}$$

Employing the designed method, the information of all complex feature maps can be used as much as possible by reducing the missed detection rate of objects.

4 Result Analysis

4.1 Result Analysis for Object Classification

In the experiments, we involve 3 different backbones, VGG16 [23], ResNet18 [4] and SENet18 [24] for comparison. It is clearly that the connection methods designed for complex architectures are obviously robust to classification problems. We designed 5 methods to connect the complex feature map and the final output result. All the classification results have been significantly improved, which shows that the complex backbone network has fundamentally improved the performance of the backbone network. When the number of parameters of the multiple backbone networks is doubled, from the experimental results of the three networks, a performance improvement of 2–4% points can be achieved. Among them, due to the parameter quantity's redundancy of the fully connected layer, the VGG16 network's parameter quantity can be basically unchanged. Similarly, when the number of parameters is reduced to the half by reducing the number of convolution channels, according to the statistical results of the three network experiments, a classification improvement of 1–2.5% points can be achieved (VGG16 network can be reduced to a quarter of the original parameter). The above experiments are sufficient to demonstrate that the improvement of the complex architectures backbone network is owing to the backbone itself (Table 1).

4.2 Result Analysis for Object Detection

In this part, we use the VOC data set to evaluate the improvement of the detection effect by complex architectures network. We employ VOC2007trainval dataset and VOC2012trainval dataset as the training data, while utilize VOC2007test dataset during the test. For the part without pre-trained model, "darknet-23" is used as backbone network. We tested non-complex convolution model and three variants of the complex convolution model (using only real part of feature map, using signed-magnitude of feature map, and using both real and imaginary parts of feature map) on it. Obviously, for any networks, the involvement of complex convolution on the backbone network will improve the feature map extraction ability. For the part with pre-trained model, since it is hard to train the complex darknet-53 pre-trained model on ImageNet dataset due to the hardware limitations, the real-number pre-trained model is loaded twice for both real and imaginary parts. Therefore, in this part, we don't compare the results with original darknet53 detection model. On the other hand, we compare the 3 improved models to evaluate the effectiveness of the improvement. We can see that under the condition of using the same complex convolution backbone network, employing full-feature map information (signed-magnitude) will improve the detection effect by nearly 5% points compared with only using real-part feature map. More than that, using the information of the real and imaginary feature map together by NMS will improve the detection effect by nearly 8% points compared with only using the real part of feature map. This shows that our proposed method, using non-maximum suppression methods to jointly apply real and imaginary feature maps for bounding box prediction, is effective in object detection task. Additionally, it can be applied to various detection models and tasks in the future (Table 2).

Table 1. CIFAR10 result by complex method.

cifar10 (%)	Baseline	Summation	Magnitude	Absoluted-summation	Signed-magnitude	Convolution1 × 1 (after classifier)
VGG16	84.44 (0)	88.25 (3.81)	88.75 (4.31)	88.76 (4.32)	88.65 (4.21)	88.55 (4.11)
Parameter numbers:	33.647 m	67.273 m	67.273 m	29.977 m (remove 1 FC layer)	29.977 m (remove 1 FC layer)	29.977 m (remove 1 FC layer)
ResNet18	88.04 (0)	90.23 (2.19)	90.06 (2.02)	90.21 (2.17)	90.68 (2.64)	90.38 (2.34)
Parameter numbers:	11.174 m	22.353 m	22.353 m	22.353 m	22.353 m	22.353 m
SENet18	87.4 (0)	90.24 (2.84)	90.38 (2.98)	90.11 (2.71)	90.34 (2.94)	90.47 (3.07)
Parameter numbers:	11.390 m	22.771 m	22.771 m	22.771 m	22.771 m	22.771 m
cifar10 (%)	Baseline	Small summation	Small magnitude	Small absoluted-summation	Small signed-magnitude	Small convolution1 × 1 (after classifier)
VGG16	84.44 (0)	87.24 (2.8)	86.79 (2.35)	86.85 (2.41)	87.08 (2.64)	87.76 (3.32)
Parameter numbers:	33.648 m	16.845 m	16.845 m	7.503 m (remove 1 FC layer)	7.503 m (remove 1 FC layer)	7.503 m (remove 1 FC layer)
ResNet18	88.04 (0)	89.14 (1.1)	89.05 (1.01)	89.04 (1)	89.85 (1.81)	89.16 (1.12)
Parameter numbers:	11.174 m	5.598 m	5.598 m	5.598 m	5.598 m	5.598 m
SENet18	87.4 (0)	88.98 (1.58)	89.14 (1.74)	89.47 (2.07)	89.76 (2.36)	89.08 (1.68)
Parameter numbers:	11.390 m	5.641 m	5.641 m	5.641 m	5.641 m	5.641 m

Table 2. VOC result by complex method.

VOC(mAP)	Baseline	Real	Signed-magnitude	Real and imaginary with NMS
"Darknet-23" (without pretrain model)	0.608413 (0)	0.617585 (0.009172)	0.622672 (0.014259)	0.631942 (0.023529)
Darknet-53 (load pretrain model both real and imaginary)	–	0.685472 (0)	0.731432 (0.04596)	0.759334 (0.073862)

5 Conclusion and Future Works

In this work, we use CIFAR and VOC datasets to verify the effectiveness of the complex architecture on the backbone network. Through theoretical analysis and experimental verification, the following conclusions can be obtained. The complex convolution architectures can improve the performance of feature extraction of backbone network and improve performance in either classification or object detection. Moreover, for classification tasks, classification performance can be improved without increasing or even reducing the number of parameters. In the object detection task, when the prediction frames jointly listed by the real and imaginary feature map are combined by using the non-maximum suppression, it will significantly improve the detection performance. Moreover, this method could be generalized for any detection task.

We plan to employ ImageNet and COCO data sets for more scientifically verifying the ability of complex architectures on improving the performance of the backbone network. Meanwhile, we expect to explore whether the joint detection method of real and imaginary parts based on non-maximum suppression has certain improvement capabilities on some special tasks, such as small target detection tasks or unclear target detection tasks, by reducing the missed detection rate for difficult-to-identify targets.

Acknowledgement. Baochang Zhang is also with Shenzhen Academy of Aerospace Technology, Shenzhen, China, and he is the corresponding author. He is in part Supported by Shenzhen Science and Technology Program (No. KQTD2016112515134654).

References

1. Jiao, L., Zhang, F., Liu, F., et al.: A survey of deep learning-based object detection. IEEE Access **7**, 128837–128868 (2019)
2. Hu, Y., Yang, Y., Zhang, J., et al.: Attentional kernel encoding networks for fine-grained visual categorization. IEEE Trans. Circ. Syst. Video Technol. (2020)
3. He, K., Zhang, X., Ren, S., Sun, J.: Identity mappings in deep residual networks. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9908, pp. 630–645. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46493-0_38

4. He, K., Zhang, X., Ren, S., et al.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
5. He, K., Gkioxari, G., Dollár, P., et al.: Mask R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2961–2969 (2017)
6. Xie, S., Girshick, R., Dollár, P., et al.: Aggregated residual transformations for deep neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1492–1500 (2017)
7. Gao, S., Cheng, M.M., Zhao, K., et al.: Res2net: a new multi-scale backbone architecture. IEEE Trans. Pattern Anal. Mach. Intell. (2019)
8. Chen, J., Ran, X.: Deep learning with edge computing: a review. Proc. IEEE **107**(8), 1655–1674 (2019)
9. Howard, A.G., Zhu, M., Chen, B., et al.: Mobilenets: efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 (2017)
10. Zhang, X., Zhou, X., Lin, M., et al.: Shufflenet: an extremely efficient convolutional neural network for mobile devices. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6848–6856 (2018)
11. Iandola, F.N., Han, S., Moskewicz, M.W., et al.: SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and $<$ 0.5 MB model size. arXiv preprint arXiv:1602.07360 (2016)
12. Trabelsi, C., Bilaniuk, O., Zhang, Y., et al.: Deep complex networks. arXiv preprint arXiv:1705.09792 (2017)
13. Choi, H.S., Kim, J.H., Huh, J., et al.: Phase-aware speech enhancement with deep complex U-Net. arXiv preprint arXiv:1903.03107 (2019)
14. Pfeifenberger, L., Zöhrer, M., Pernkopf, F.: Deep complex-valued neural beamformers. In: ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 2902–2906. IEEE (2019)
15. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167 (2015)
16. Li, Z., Peng, C., Yu, G., et al.: Detnet: a backbone network for object detection. arXiv preprint arXiv:1804.06215 (2018)
17. Krizhevsky, A., Hinton, G.: Learning multiple layers of features from tiny images. In: Handbook of Systemic Autoimmune Diseases, vol. 1, no. 4 (2009)
18. Fei-Fei, L., Fergus, R., Perona, P.: Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories. In: 2004 Conference on Computer Vision and Pattern Recognition Workshop, p. 178. IEEE (2004)
19. Deng, J., Dong, W., Socher, R., et al.: Imagenet: a large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255. IEEE (2009)
20. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105 (2012)
21. Huang, G., Liu, Z., Van Der Maaten, L., et al.: Densely connected convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4700–4708 (2017)
22. Szegedy, C., Liu, W., Jia, Y., et al.: Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–9 (2015)
23. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
24. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7132–7141 (2018)
25. Luan, S., Chen, C., Zhang, B., et al.: Gabor convolutional networks. IEEE Trans. Image Process. **27**(9), 4357–4366 (2018)

26. Girshick, R., Donahue, J., Darrell, T., et al.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 580–587 (2014)
27. Girshick, R.: Fast R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1440–1448 (2015)
28. Ren, S., He, K., Girshick, R., et al.: Faster R-CNN: towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems, pp. 91–99 (2015)
29. He, K., Zhang, X., Ren, S., et al.: Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Trans. Pattern Anal. Mach. Intell. **37**(9), 1904–1916 (2015)
30. Redmon, J., Divvala, S., Girshick, R., et al.: You only look once: unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 779–788 (2016)
31. Liu, W., et al.: SSD: Single Shot MultiBox Detector. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9905, pp. 21–37. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46448-0_2
32. Liu, Z., Jiang, Z., Wei, F.: OD-GCN object detection by knowledge graph with GCN. arXiv preprint arXiv:1908.04385 (2019)
33. Kong, T., Sun, F., Yao, A., et al.: Ron: reverse connection with objectness prior networks for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5936–5944 (2017)
34. Bochkovskiy, A., Wang, C.Y., Liao, H.Y.M.: YOLOv4: Optimal Speed and Accuracy of Object Detection. arXiv preprint arXiv:2004.10934 (2020)