Improving Backbones Performance by Complex Architectures

Jinxin Shao[1], Yutao Hu[3], Zhen Liu[1], Teli Ma[1], Baochang Zhang[1, 2,*]

1.School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, P.R. China

2. Shenzhen Academy of Aerospace Technology, Shenzhen Guangdong 518057, P.R. China

3. School of Electronic and Information Engineering, Beihang University, Beijing 100191, P.R. China

{shaojinxin,huyutao,liuzhenbuaa,mtl9868,bczhang}@ buaa.edu.cn

—NANJING · 2020—

Introduction

How to improve the feature extraction capability of the backbone network is a basic problem in computer vision research

The main content of this research is to improve the backbone network through the complex structure

Introducing and combining characteristics into classification and detection problems to achieve complex structure improvement is the key technology of research

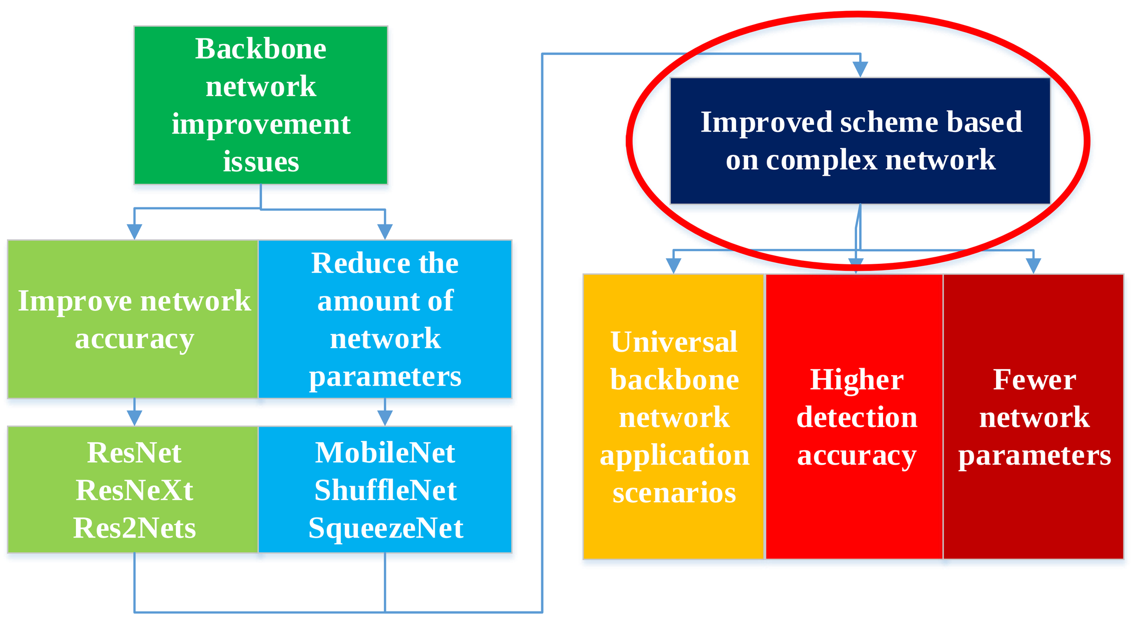

Fig.1 Main Topic of this Paper

Motivation

The realization principle of complex convolutional neural network mainly has 3 parts:

1. Complex Convolution (Conv)
2. Complex number batch normalization (BN)
3. Complex linear rectification function (ReLU) and other functions

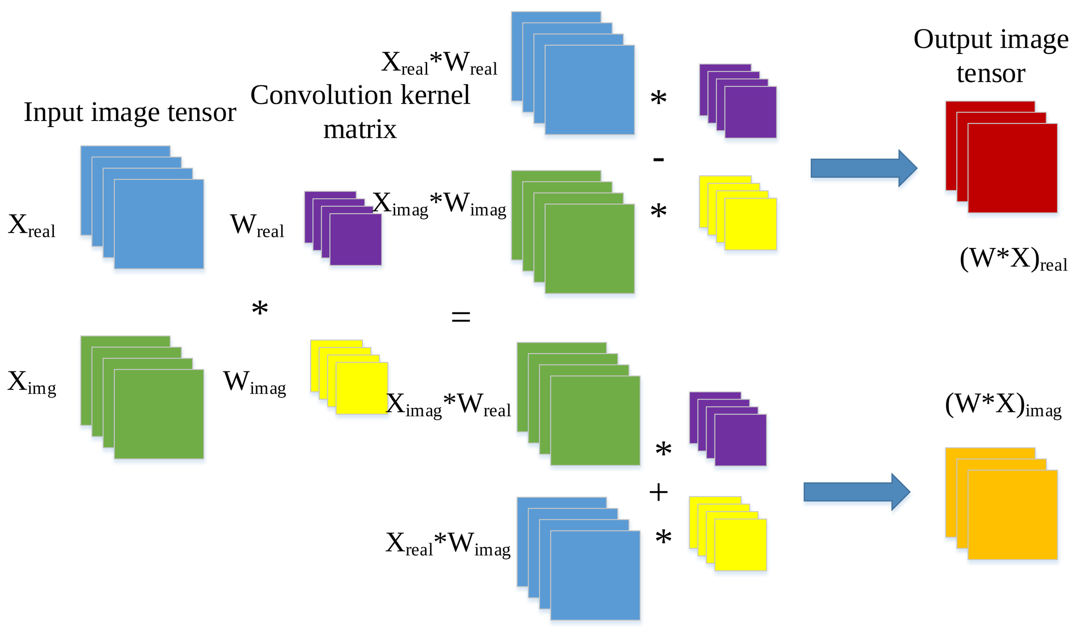

Fig.2 Main idea of Complex Convolution

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

The convolution kernel matrix is W

The input image vector is x

$x-E[x]$ refers to the deviation of the two-dimensional data from the center

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

The covariance matrix is expressed as:

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

The BN formula is:

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

The ReLU formula is:

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

When input is in the 1st or 3rd quadrant, formula satisfies Cauchy-Riemann equation

Method/Model

For classification tasks, you only need to design some simple rules to combine complex low-dimensional feature maps

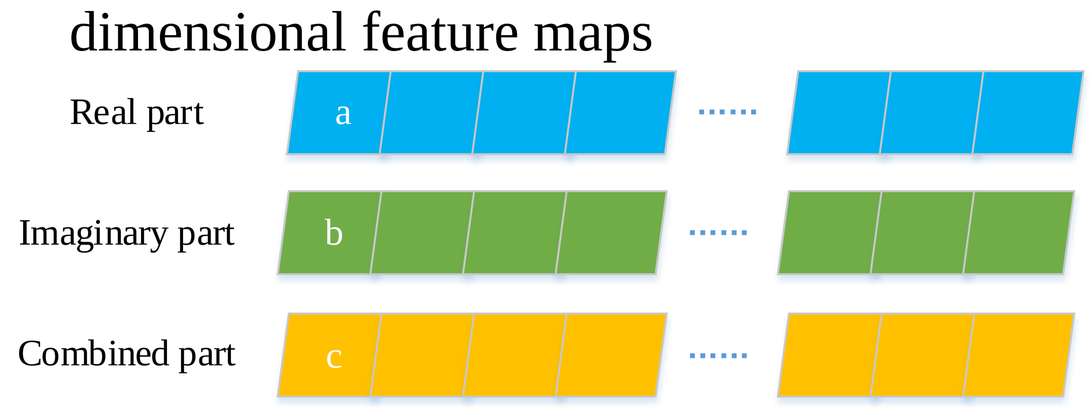

Fig. 3 Combination Method of Complex Vector

$v_{real}, v_{imaginary}$ represents the input 2 vectors

Equivalent to the input object is a mathematical complex number: $v_{real} + v_{imaginary} i$

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

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

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

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

Conv1:use 1×1 convolution kernel to connect the real and imaginary part

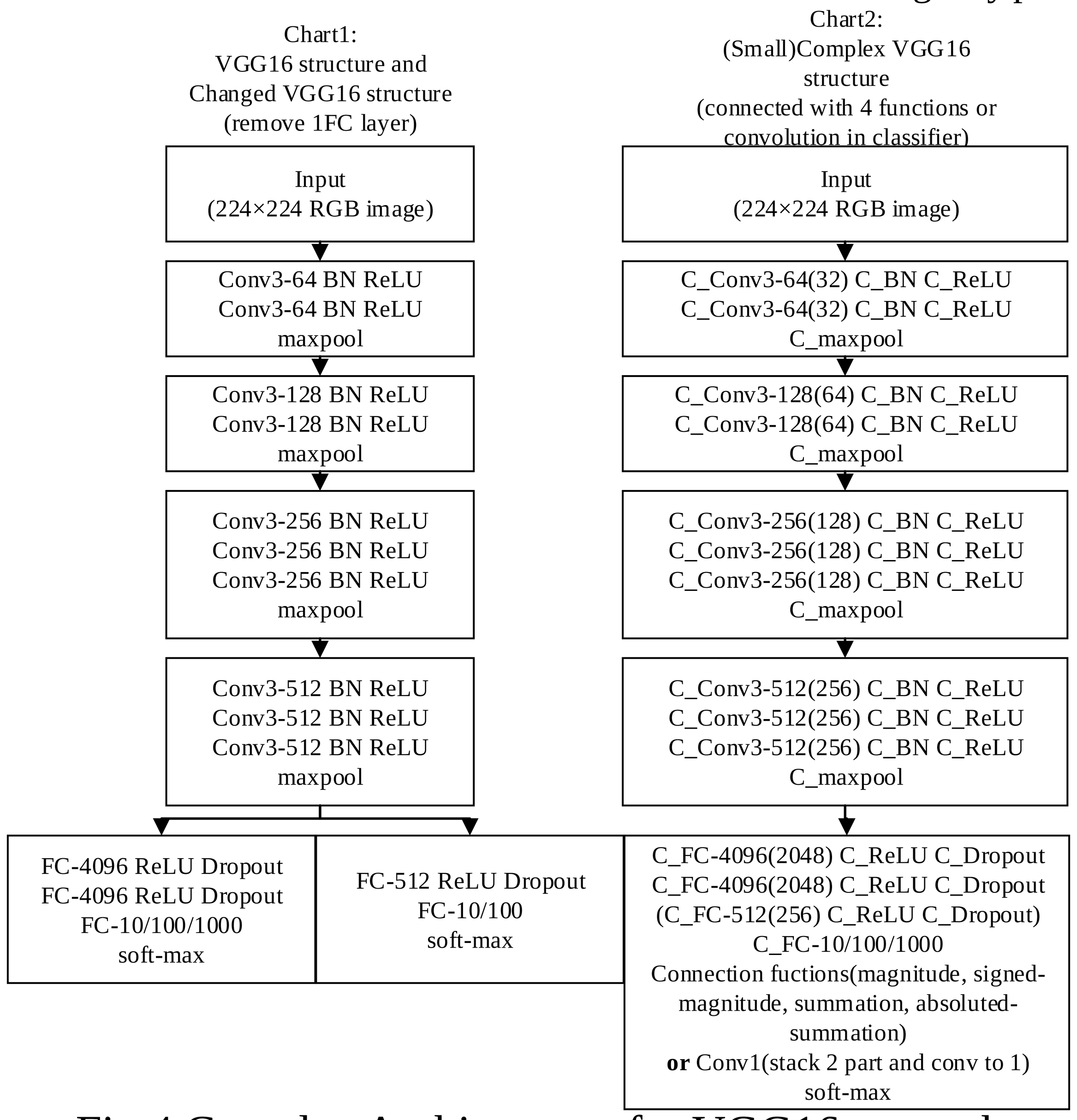

Fig.4 Complex Architectures for VGG16 network

Experiments

For detection task:

1. Only use real part of the output feature map
2. Use complex feature map's Signed-magnitude
3. Use combination of real and imaginary feature maps based on non-maximum suppression method

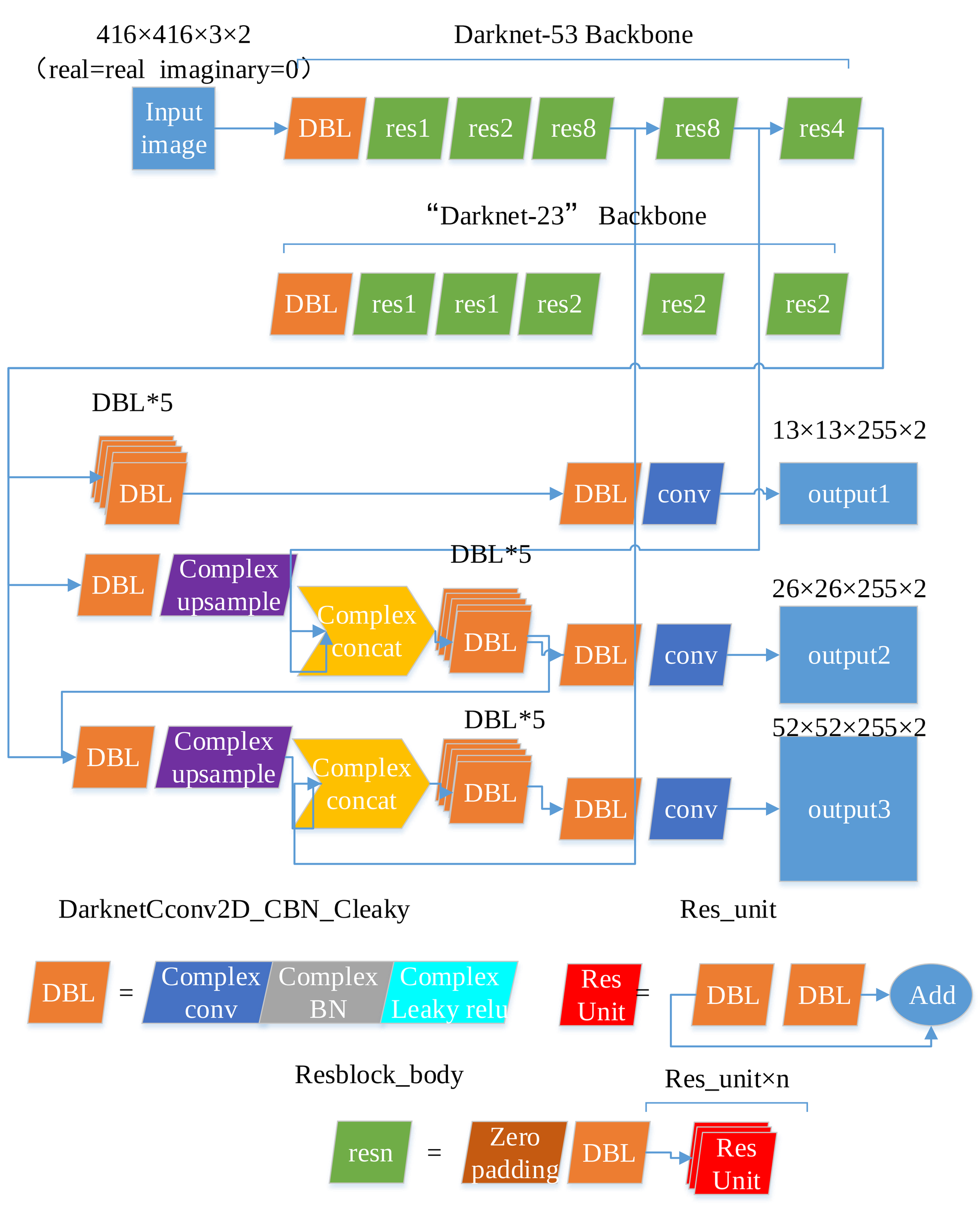

Fig.5 Complex Structure of Darknet53 and Darknet23

Experiments (cont.)

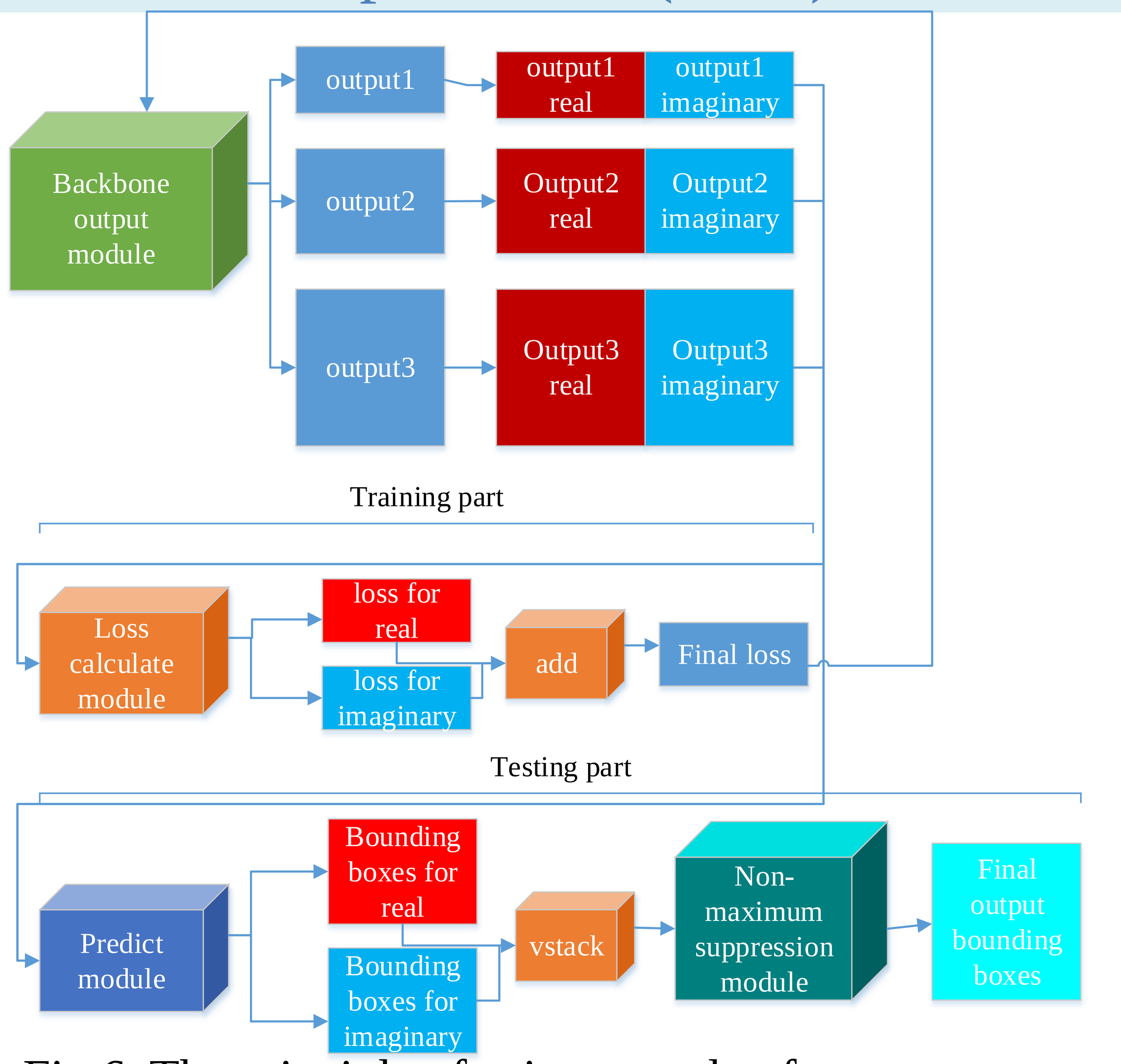

Fig.6 The principle of using complex feature maps for training and testing

The entire loss function can be written as:

$$L_{final} = L_{real} + L_{imaginary}$$

The final output of the bounding box can be written as:

$$bboxes_{output} = bboxes_{real} \bigcup bboxes_{imaginary}$$

Table 1. CIFAR10 results of complex number

cifar10(%)	Baseline	Summation	Magnitude	Absoluted-summation	Signed-magnitude	Convolution1×1 (after classifier)
VGG16	84.44	88.25	88.75	88.76	88.65	88.55
Parameter numbers:	33.647m	67.273m	67.273m	29.977m (remove 1 FC layer)	29.977m (remove 1 FC layer)	29.977m (remove 1 FC layer)
ResNet18	88.04	90.23	90.06	90.21	90.68	90.38
Parameter numbers:	11.174m	22.353m	22.353m	22.353m	22.353m	22.353m
SENet18	87.4	90.24	90.38	90.11	90.34	90.47
Parameter numbers:	11.390m	22.771m	22.771m	22.771m	22.771m	22.771m
cifar10(%)	Baseline	Small summation	Small magnitude	Small absoluted-summation	Small signed-magnitude	Small convolution1×1 (after classifier)
VGG16	84.44	87.24	86.79	86.85	87.08	87.76
Parameter numbers:	33.648m	16.845m	16.845m	7.503m (remove 1 FC layer)	7.503m (remove 1 FC layer)	7.503m (remove 1 FC layer)
ResNet18	88.04	89.14	89.05	89.04	89.85	89.16
Parameter numbers:	11.174m	5.598m	5.598m	5.598m	5.598m	5.598m
SENet18	87.4	88.98	89.14	89.47	89.76	89.08
Parameter numbers:	11.390m	5.641m	5.641m	5.641m	5.641m	5.641m

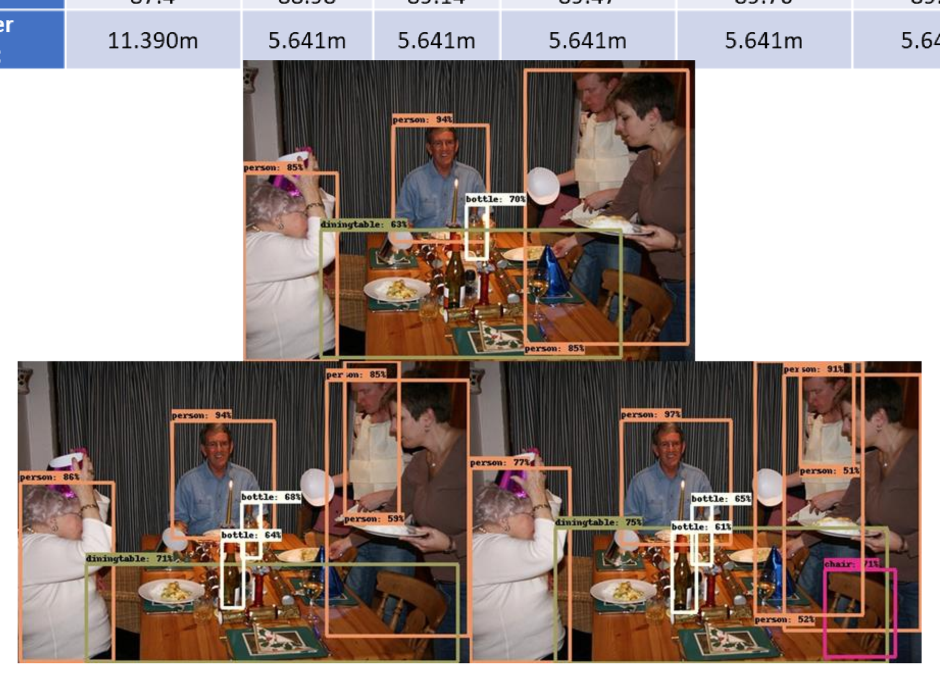

Fig.7 Results of 3 methods, VOC2007test No. 72

Table 2. VOC detection Results of Complex Structure

VOC(mAP)	Baseline	Real	Signed-magnitude	Real and imaginary
Darknet-23 (without pretrain model)	0.608413	0.617585	0.622672	0.631942
Darknet-53 (without pretrain model)	--	0.649837	0.651723	0.685857
Darknet-53 (load pretrain model both real and imaginary)	--	0.685472	0.731432	0.759334

Conclusion

1. This work introduces the complex number network and we employ it in the classification and detection tasks to achieve the better performance.
2. The proposed method can generalize to different computer vision tasks and achieve great performance, especially for the detection of small targets.