Research on Lightweight Improvement of Sonar Image Classification Network

Yan Zhou\textsuperscript{1, a}, Shaochang Chen\textsuperscript{1, b}

\textsuperscript{1}School of Electronic Engineering, Naval University of Engineering, Wuhan, Hubei, 430033, China
\textsuperscript{a}email: zhangzs18@mail.tsinghua.edu.cn, \textsuperscript{b}email: 893209671@qq.com

Abstract. According to the constructed sonar common target detection data set derived classification dataset, the convolutional neural network is used to classify the target. And in order to be able to better apply actual engineering, the lightweight of the network has been studied in depth. First, the performance of the ordinary convolutional neural network VGG-16 and the lightweight convolutional neural network MobileNet on the derived classification dataset is compared. The results show that the lightweight network achieves better results at a smaller cost. Then use the dilated convolution and bottleneck structure to improve the MobileNet network to obtain the IMDNet network and the IMBNet network. Through experimental comparison and analysis, the two improved networks have a significant reduction in the amount of parameters and calculations compared to the MobileNet network. The performance of the IMBNet network is basically the same as that of the MobileNet network, and the accuracy of the IMDNet network is 2.02\% higher than that of the MobileNet network. This shows that in the field of sonar image classification research, lightweight convolutional neural networks have good performance, and have certain practical application prospects and values.

1. Introduction

With the national marine strategy proposed, activities in the underwater field have become increasingly frequent, such as submarine salvage \cite{1}, underwater pipeline rupture detection \cite{2}, underwater object tracking, recognition and positioning \cite{3}, etc. The development of artificial intelligence has shown great advantages in the application of deep learning in various fields, which has also brought new ideas and new directions to the research in the underwater field. Underwater targets are mainly obtained by sonar. The signals obtained by collecting sonar are processed into images \cite{4}, and the shape of underwater targets can be observed intuitively. Underwater sonar image target classification is to identify the category of the target through an image \cite{5}. At present, image target classification is mostly for optical images, and the application of convolutional neural networks in the field of optical images has been very successful. However, research on acoustic images is still lacking, and public datasets of underwater acoustic images are also scarce. To solve the problem of the lack of underwater acoustic image dataset, the Sonar Common Target Detection Dataset (SCTD) \cite{6} was constructed, and the derived classification dataset (SCTD-C) was extracted from this. Based on this classification dataset, carry out related research on classification networks.

In the SCTD-C dataset, the advantages of convolutional neural networks are used to extract the deep features of the image, and the relevant knowledge of transfer learning is used to apply the convolutional neural network applied in the field of optical images to the field of sonar images. By
simply adjusting the network structure parameters, it is suitable for the classification of sonar images. This article starts from the classification of sonar images by the classic convolutional neural network VGG-16 [7] and the lightweight convolutional neural network MobileNet [8], compares the performance of the two networks on the SCTD-C dataset. The MobileNet network was also improved and optimized to further enhance the network performance.

2. Research on Sonar Image Classification Based on VGG-16 Network and MobileNet Network

2.1. Convolutional Neural Network VGG-16
VGGNet is a deep convolutional neural network developed by researchers from the Computer Vision Group of Oxford University and Google DeepMind. It explores the relationship between the depth of a convolutional neural network and its performance. A 16-19 layer convolutional neural network is constructed by repeatedly stacking a convolutional layer with a convolution kernel of $3 \times 3$ and a maximum pooling layer with a step size of $2 \times 2$. The structure of VGG-16 is shown in Figure 1.

VGG-16 contains a total of 13 convolutional layers, 5 pooling layers, and 3 fully connected layers. The convolutional layer and fully connected layer have weight coefficients, while the pooling layer does not involve weights, so VGG-16 has 16 weight layers. Its convolutional layer and pooling layer can be divided into different blocks. Each block contains several convolutional layers and a pooling layer. The structure of VGG-16 according to the block division is shown in Figure 2.

2.2. Lightweight Convolutional Neural Network MobileNet
Starting from the application of convolutional neural networks, in order to pursue higher accuracy, neural networks are more inclined to deeper and more complex structural designs, which directly leads to a great dependence on computing power, and the demand for GPUs is also increasing. But this is often difficult to achieve in real life. In real life, recognition tasks are calculated in real time under a limited computing power environment. Because these calculations are basically performed on the mobile terminal. Therefore, a lighter network model is required to be deployed on mobile terminals.

The MobileNet network is a lightweight convolutional neural network proposed by Google for mobile embedded devices such as mobile phones. It does not use model compression technology [9], but redesigns the network structure and proposes a new convolution method-depthwise separable
convolutions [8], through the new convolution method to reduce the number of parameters and increase the speed of operation. Depth separable convolution is two layers, one layer is depthwise convolution [10], the other layer is pointwise convolution (ie 1×1 standard convolution). The depth convolutional layer in the MobileNet network is followed by a BN [11] layer and a ReLU6 [12] layer. The point-by-point convolutional layer is also followed by a BN layer and a ReLU6 layer. The comparison between the depth separable convolution block and the traditional standard convolution block is shown in figure 3. Compared with the traditional standard convolution block, the depth separable convolution block has more ReLU6 activation functions, which increases the nonlinear transformation of the model. The generalization ability is stronger.

![Figure 3. Comparison of standard convolution block and depth separable convolution block (left is standard convolution block, right is depthwise separable convolution block)](image)

The entire MobileNet network is composed of one layer of standard convolution and thirteen sets of depth separable convolutions. Each set of depth separable convolutions can be divided into two layers. Coupled with the last fully connected layer, the MobileNet network has 28 layers in total. Its network structure is shown in table 1, where Conv dw represents depth convolution.

| Type/Stride  | Filter Shape  | Input Size  |
|--------------|---------------|-------------|
| Conv/s2      | 3×3×3×32     | 224×224×3   |
| Conv dw/s1   | 3×3×32 dw    | 112×112×32  |
| Conv/s1      | 1×1×32×64    | 112×112×32  |
| Conv dw/s2   | 3×3×64 dw    | 112×112×64  |
| Conv/s1      | 1×1×64×128   | 56×56×128   |
| Conv dw/s1   | 3×3×128 dw   | 56×56×128   |
| Conv/s1      | 1×1×128×128  | 56×56×128   |
| Conv dw/s2   | 3×3×128 dw   | 56×56×128   |
| Conv/s1      | 1×1×128×256  | 28×28×128   |
| Conv dw/s1   | 3×3×256 dw   | 28×28×256   |
| Conv/s1      | 1×1×256×256  | 28×28×256   |
| Conv dw/s2   | 3×3×256 dw   | 28×28×256   |
| Conv/s1      | 1×1×256×512  | 14×14×256   |
| Conv dw/s1   | 3×3×512 dw   | 14×14×512   |
| Conv/s1      | 1×1×512×512  | 14×14×512   |
| Conv dw/s2   | 3×3×512 dw   | 14×14×512   |
| Conv/s1      | 1×1×512×1024 | 7×7×512     |
| Conv dw/s1   | 3×3×1024 dw  | 7×7×1024    |
| Conv/s1      | 1×1×1024×1024| 7×7×1024    |
| Avg Pool/s1  | Pool 7×7     | 7×7×1024    |
| FC/s1        | 1024×1000    | 1×1×1024    |
| Softmax/s1   | Classifier   | 1×1×1000    |

![Table 1. MobileNet network structure](image)
2.3. Performance of the two networks on the SCTD-C dataset

The operating system of the experimental machine is Window10, the CPU uses an Intel Core i5-8400 six-core processor, the memory is 8GB, and the graphics card is NVIDIA's GTX1050Ti 4G GDDR5 discrete graphics card. The editor is Microsoft's Visual Studio Code. The programming language uses Python, and the deep learning framework is Keras.

The original VGG-16 and MobileNet networks are designed for the ImageNet dataset, so the category parameters of their final fully connected layer output are all 1000. On the SCTD-C dataset, because there are only 4 types of targets, the output category parameter of the fully connected layer should be changed to 4. The original VGG-16 and MobileNet networks both normalized the input image to a size of 224×224. However, there are small-size images in the SCTD-C data set, so the network input image size is normalized to 96×96. Perform model comparison experiments in the same experimental environment, using the same data enhancement strategy, the number of training rounds is 200. Accuracy is selected as the evaluation index. The higher the value, the more accurate the prediction of the category the object belongs to. The optimizers all use Adam optimization algorithm, and the initial learning rate is set to 0.001. The learning rate decay value after each parameter update is $5 \times 10^{-6}$, and the batch_size is 32.

During the experiment, it was found that during the training process of the VGG-16 network, the accuracy of the training dataset was increasing, while the accuracy of the verification dataset was always maintained at a constant value. After consulting the data, it may be that the sample of the dataset is relatively small, and the network has over-fitting. Therefore, the original VGG-16 network structure is modified and the regularization item is added. The specific operation is to add a batch normalization (BN) [11] layer after the activation function of each convolutional layer, which can speed up the training and convergence of the network, prevent overfitting.

Since the accuracy of the prediction results is different after each training. In order to ensure the reliability of the prediction results, take the average of the accuracy of 5 prediction results as the value of the accuracy of the final prediction result. Let $A_i$ be the value of the accuracy of the $i$-th prediction result, $m_A$ is the average value of the accuracy of the prediction result, then:

$$m_A = \frac{A_1 + A_2 + \cdots + A_i}{i}, (i = 1, 2, 3, \cdots)$$

The performance of these two networks after training is shown in table 2.

| Classification experiment network | Accuracy of prediction result $(A_i)$ | Average prediction result accuracy rate $(m_A)$ | Parameter | Calculation amount |
|----------------------------------|--------------------------------------|-----------------------------------------------|-----------|-------------------|
| VGG-16                           | 93.06% 91.32%                        | 92.01%                                        | 15,978,308 | 31,993,746        |
| MobileNet                        | 94.10% 88.54%                        | 93.82%                                        | 3,232,964  | 6,586,604         |

It can be seen from table 2 that the accuracy of the prediction results of the MobileNet network is about 1% higher than that of the VGG-16 network, but the amount of parameters and calculations of the MobileNet network is only 1/5 of that of the VGG-16 network. The lightweight convolutional
neural network MobileNet spends less storage and calculation costs without losing network performance. So the classification performance on the SCTD-C dataset, the lightweight convolutional neural network MobileNet is a better choice.

3. Improved algorithm based on MobileNet network structure

3.1. Improve the MobileNet network by using dilated convolution

Dilated convolution [13] is to inject holes into the standard convolution kernel. The adjacent elements of the convolution kernel are applied to pixels separated by a certain distance in the image to increase the receptive field. Another parameter, the expansion rate, is introduced to define the interval between the value and the value of the convolution kernel, as shown in figure 4.

![Figure 4. Dilated convolution](image)

Figure 4(a) shows a 3×3 dilated convolution with an expansion rate of 1, which is equivalent to an ordinary standard convolution. Figure 4(b) is a 3×3 dilated convolution with an expansion rate of 2 based on figure 4(a). Although the convolution kernel only convolves with 9 points in the figure, the receptive field of the convolution increases to 7×7. Figure 4(c) is a 3×3 dilated convolution with an expansion rate of 4 based on figure 4(b), the receptive field of the convolution increases to 15×15. Although their receptive fields are different, the number of parameters associated with each layer is the same, 3×3=9. The receptive field increases exponentially, while the parameter quantity increases linearly.

MobileNet uses a large number of depth separable convolutions, which play an important role in ensuring the performance of the network, reducing the amount of calculation, and reducing the amount of parameters. However, because the MobileNet network uses standard convolution in addition to the first layer, the following 13 groups all use depth separable convolution, which will use a lot of 1×1 convolution, resulting in the amount of calculation and parameter occupied by 1×1 convolution is still very high. And although 1×1 convolution helps to encode information between channels, it is difficult to obtain spatial information. So it is difficult to process and obtain richer feature expressions in high-dimensional space.

Dilated convolution can expand the receptive field without losing feature information by downsampling, so that the output of each convolution contains a larger range of information, and enhance its ability to express features. For the SCTD-C dataset, because the sample size is not very large, too deep networks are prone to overfitting. The MobileNet network has 28 layers in total, which is already a relatively deep network. You can consider reducing the number of network layers when improving it, and using dilated convolution to replace part of the depth separable convolution.

In the original MobileNet structure, there are five layers of depth separable convolutions with a step size of 2 and eight layers of depth separable convolutions with a step size of 1. The next three layers of depth separable convolutions with a step size of 1 are deleted. The first two are retained. The next five layers of depth separable convolutions with a step size of 1 are deleted, and replace the first three layers with dilated convolutions with an expansion rate of 2. The modified network is named
IMDNet, and its network structure is shown in table 3, where Conv D2 represents a dilated convolution with an expansion rate of 2, and Conv dw represents a deep convolution.

### Table 3. IMDNet network structure

| Type/Stride | Filter Shape | Input Size   |
|-------------|--------------|--------------|
| Conv/s2     | 3×3×3×32     | 96×96×3      |
| Conv D2/s1  | 3×3×32×64    | 48×48×32     |
| Conv dw/s2  | 3×3×32 dw    | 48×48×64     |
| Conv/s1     | 1×1×64×128   | 24×24×64     |
| Conv D2/s1  | 3×3×128×128  | 24×24×128    |
| Conv dw/s2  | 3×3×128 dw   | 24×24×128    |
| Conv/s1     | 1×1×128×256  | 12×12×128    |
| Conv D2/s1  | 3×3×256×256  | 12×12×256    |
| Avg Pool/s1 | Pool 12×12   | 12×12×256    |
| FC/s1       | 256×4        | 1×1×256      |
| Softmax/s1  | Classifier   | 1×1×4        |

The entire IMDNet network includes one layer of standard convolution, three layers of dilated convolution with an expansion rate of 2 and two layers of depth separable convolution, where each layer of depth separable convolution is divided into two layers of convolution, and finally a full connection is added. The IMDNet network has a total of 9 layers, which greatly reduces the number of network layers compared to the MobileNet network. Although in terms of a single convolution, the parameter amount of the dilated convolution is higher than that of the depth separable convolution. However, the receptive field of the dilated convolution is larger, which is more conducive to the extraction and expression of features, and enhances the information representation ability of the network model.

### 3.2. Using Bottleneck Structure to Improve MobileNet Network

The bottleneck structure [14] first uses a 1×1 convolutional layer to perform feature compression on the input, then uses a 3×3 convolutional layer for feature extraction, and then uses a 1×1 convolutional layer for feature expansion. This structure reduces the amount of many parameters compared to directly performing 3×3 convolution on the input. As shown in figure 5, assuming that the dimension of the input feature map is 128 dimensions, and the dimension of the required output feature map is also 128 dimensions. If the operation in figure 5(a) is used, the input 128-dimensional feature map directly passes through a 3×3×128 convolutional layer. The output is a 128-dimensional feature map, so its parameter quantity is: 128×3×3×128=147456; if the operation in figure 5(b) is used, the input 128-dimensional feature map passes through a 1×1×32 convolution layer, then a 3×3×32 convolution layer, and finally a 1×1×128 convolution layer. The output is also a 128-dimensional feature map. The calculation amount is: 128×1×1×32+32×3×3×32+32×1×1×128=17408. After comparison, it can be found that the bottleneck structure can greatly reduce the amount of parameters, and after the first 1×1 convolution is used for dimensionality reduction, data training and feature extraction can be performed more directly and effectively.
As the bottleneck structure also has the effect of reducing the amount of parameters, it can be considered to add the bottleneck structure when improving the MobileNet network structure. According to the improvement method of the IMDNet network in Section 3.1, the dilated convolution is replaced with the bottleneck structure. The improved network is named IMBNet. Its network structure is shown in table 4, where Bottleneck represents the bottleneck structure and Conv dw represents deep convolution.

Table 4. IMBNet network structure

| Type      | Stride | Input Size  |
|-----------|--------|-------------|
| Conv      | 2      | 96×96×3     |
| Bottleneck| 1      | 48×48×32    |
| Conv dw   | 2      | 48×48×32    |
| Conv      | 1      | 24×24×32    |
| Bottleneck| 1      | 24×24×64    |
| Conv dw   | 2      | 24×24×64    |
| Conv      | 1      | 12×12×64    |
| Bottleneck| 1      | 12×12×128   |
| Avg Pool  | 1      | 12×12×128   |
| FC        | 1      | 1×1×128     |
| Softmax   | 1      | 1×1×4       |

The entire IMBNet network includes one layer of standard convolution, three layers of bottleneck structure and two layers of depth separable convolution. Each layer of bottleneck structure can be divided into three layers of convolution, and each layer of depth separable convolution can be divided into two layers of convolution, plus the fully connected layer, the IMBNet network has a total of 15 layers. Because the IMBNet network structure deletes more depth separable convolutional layers on the basis of the original MobileNet network structure, and replaces the depth separable convolution with a step length of 1 with the bottleneck structure. The bottleneck structure uses a 1×1 convolution more than the depth separable convolution, which compresses the parameter amount of the entire network to a greater extent, so the parameter amount of the IMBNet network is greatly reduced.

4. Comparative analysis of experimental results and performance
The experimental environment platform, experimental parameter settings, and experimental strategies are the same as in section 2.3. IMDNet and IMBNet are experimented on the SCTD-C dataset. The accuracy and loss curves of the training and verification process are shown in figure 6, and the abscissa is the number of training rounds, the ordinate is the accuracy rate and the loss rate. It can be
seen from the curve that the training of the two improved network models has converged, and the convergence speed is faster, reaching the convergence when the number of rounds is 25.

![Figure 6. Improved network training, verification accuracy and loss curve (left picture is IMDNet network, right picture is IMBNet network)](image)

IMDNet network and IMBNet network have been trained and predicted 5 times on the SCTD-C dataset, and the experimental results are shown in table 5.

| Classification experiment network | Accuracy of prediction result ($A_i$) | Average prediction result accuracy rate ($mA$) | Parameter | Calculation amount |
|-----------------------------------|--------------------------------------|-----------------------------------------------|-----------|-------------------|
| IMDNet                            | 94.10%                               | 95.84%                                        | 804,516   | 1,620,731         |
|                                   | 95.14%                               |                                               |           |                   |
|                                   | 96.88%                               |                                               |           |                   |
|                                   | 96.53%                               |                                               |           |                   |
| MobileNet                         | 91.67%                               | 93.06%                                        | 253,668   | 517,053           |
|                                   | 93.75%                               |                                               |           |                   |
|                                   | 92.71%                               |                                               |           |                   |
|                                   | 94.10%                               |                                               |           |                   |
| MobileNet                         | 93.82%                               | 93.06%                                        | 3,232,964 | 6,586,604         |

From the experimental results in the above table, it can be seen that the average accuracy of the prediction results of the IMDNet network is 2.02% higher than the average accuracy of the prediction results of the MobileNet network. At the same time, the parameter amount and calculation amount of the IMDNet network are 1/4 of the MobileNet network. Although the average accuracy of the prediction results of the IMBNet network is 0.76% lower than the average accuracy of the prediction results of the MobileNet network, the performance loss of the network is not much, but the parameter amount and calculation amount of the IMBNet network is 1/13 of that of MobileNet. The amount of parameters and calculations are greatly reduced. Therefore, in the two improved algorithms, if you want the network to get a higher prediction result accuracy rate and cost less computing power, you can choose the IMDNet network; if you don’t lose network performance, use a smaller network parameters and calculations, faster calculation speed to complete training and prediction tasks, you can choose IMBNet network.
The confusion matrix of the prediction results of these two improved networks is shown in figure 7, the ordinate is the real label, and the abscissa is the predicted label. The value in the matrix represents the number of real category labels that are identified as predicted category labels. For example, the first row and the first column of the confusion matrix on the left indicate that the number of real aircraft categories that are identified as predicted aircraft categories is 14, the first row and the second column indicates that the number of real aircraft categories identified as predicted background categories is 0, the first row and the third column indicates that the number of real aircraft categories identified as predicted human categories is 0, and the first row and fourth column indicate the real aircraft category is judged as the number of predicted ship categories is 5. It can be seen from figure 7 that the IMDNet network is better than the IMBNet network in terms of the accuracy of the prediction results.

Randomly select 16 images from the test dataset to make predictions, and visualize the prediction results, as shown in figure 8. It can be seen from the figure that each image has its true category label and its predicted category label. Most images can be correctly predicted as the corresponding category, and there are also prediction errors. For example, in the prediction results of the IMDNet network in the figure, the true category label of the image in the first row and second column is the aircraft category, but it is predicted to be the ship category. In the IMBNet network prediction results in the figure, the image true category label in the fourth row and fourth column is the human category, it is predicted to be the ship category.
Figure 8. Randomly select images of the test dataset for prediction (Figure (a) is the prediction result of the IMDNet network, and figure (b) is the prediction result of the IMBNet network)

Input one test image, predict its category and display the probability that it is predicted to be that category. Select 3 images from each category in the test set to predict. The prediction result of the IMDNet network is shown in figure 9. The prediction result of the IMBNet network is shown in figure 10.
From the prediction results of the above two improved networks, it can be intuitively seen that the input image belongs to the category and its prediction probability. Most of the classifications are correct, and there are also prediction errors. For example, in the IMDNet network in figure 9, the input of the human category image, its third image is predicted to be the ship class, and the probability is 95.91%. In the IMBNet network in figure 10, the ship class image is input, and its first image is
predicted to be the background class, and the probability is 91.38%. Judging from the predicted probability value, the probability value of the IMDNet network's prediction result is higher than that of the IMBNet network.

According to the analysis of the above experimental results, if you need higher accuracy for these two networks improved on the MobileNet network, you can choose to use the IMDNet network. If you do not lose the performance of the network and require faster speed, you can choose to use IMBNet network.

5. Summary
For the SCTD-C dataset, through the idea of transfer learning, the convolutional neural network suitable for the optical image field is applied to the sonar image field. And in order to be better deployed to mobile terminals, the application of lightweight convolutional neural networks on the SCTD-C dataset is explored. In order to further improve the network, the lightweight convolutional neural network MobileNet is optimized by using the dilated convolution and bottleneck structure. The corresponding networks are IMDNet and IMBNet respectively. The classification experiment is carried out on the SCTD-C dataset. The IMBNet network is compared with MobileNet. Under the premise of basically not losing network performance, the amount of parameters and calculations are greatly reduced, reaching 1/13 of MobileNet. Compared with MobileNet, the amount of parameters and calculations of the IMDNet network is reduced by 3/4. The accuracy rate has increased by 2.02%. The performance of these two networks on the SCTD-C dataset shows that the lightweight convolutional neural network has application prospects in the field of sonar images, and has good performance, laying the foundation for further deployment to mobile terminals.

References
[1] Xu W, Li H, Zhang J, et al. (2019) Trajectory Tracking for Underwater Rescue Salvage Based on Backstepping Control *. In: Chinese Control Conference. Guangzhou, China. pp. 1956-1961.
[2] Jasper A. Oil. (2012) Gas Pipeline Leak Inspection and Repair in Underwater Poor Visibility Conditions: Challenges and Perspectives. Journal of Environmental Protection, 3(5): 394-399.
[3] Ji-Yong L, Hao Z, Hai H, et al. (2018) Design and Vision Based Autonomous Capture of Sea Organism With Absorptive Type Remotely Operated Vehicle. IEEE Access, 6: 73871-73884.
[4] Guo G, Wang X, Xu H. (2018) Survey of underwater target detection, recognition and tracking based on sonar image. Control and Decision, v.33(05): 141-157.
[5] Fan Z, Li H, Xia W, et al. (2019) Detection and classification of underwater targets in imaging sonar based on deep learning. In: The 2019 Academic Conference of the Hydroacoustics Branch of the Acoustic Society of China. Nanjing, China. pp. 266-268.
[6] Zhou Y, Chen S, Wu K, Ning M, Chen H, Zhang P. (2021) SCTD1.0: Sonar common target detection dataset. Computer Science, Z11
[7] Simonyan K, Zisserman A. (2014) Very Deep Convolutional Networks for Large-Scale Image Recognition. Computer Science, arXiv: 1409.1556.
[8] Howard A G, Zhu M, Chen B, et al. (2017) MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. Computer Science, arXiv: 1704.04861.
[9] Cheng Y, Wang D, Zhou P, et al. (2017) A Survey of Model Compression and Acceleration for Deep Neural Networks. Computer Science, arXiv: 1710.09282.
[10] Sifre L, Mallat, Stéphane. (2014) Rigid-Motion Scattering for Texture Classification. Computer Science, 3559: 501-515.
[11] Ioffe S, Szegedy C. (2015) Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. Computer Science, arXiv: 1502.03167.
[12] Krizhevsky, Alex. (2012) Convolutional Deep Belief Networks on CIFAR-10.
[13] Yu F, Koltun V. (2016) Multi-Scale Context Aggregation by Dilated Convolutions. Computer Science, arXiv: 1511.07122.

[14] He K, Zhang X, Ren S, et al. (2016) Deep Residual Learning for Image Recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas, NV, USA. pp.770-778.