Hindawi
Mobile Information Systems
Volume 2019, Article ID 6536925, 8 pages
https://doi.org/10.1155/2019/6536925

Research Article

TANet: A Tiny Plankton Classification Network for Mobile Devices

Xiu Li[1], **Rujiao Long**[1], **Jiangpeng Yan**[1], **Kun Jin**[1] and **Jihae Lee**[2]

[1]*Department of Automation, Graduate School at Shenzhen, Tsinghua University, Shenzhen, China*
[2]*Institute for Interdisciplinary Information Sciences, Tsinghua University, Beijing, China*

Correspondence should be addressed to Xiu Li; li.xiu@sz.tsinghua.edu.cn and Rujiao Long; longrj16@mails.tsinghua.edu.cn

Received 15 November 2018; Revised 1 February 2019; Accepted 6 March 2019; Published 3 April 2019

Academic Editor: Raul Montoliu

This paper is devoted to a lightweight convolutional neural network based on the attention mechanism called the tiny attention network (TANet). The TANet consists of three main parts termed as a reduction module, self-attention operation, and group convolution. The reduction module alleviates information loss caused by the pooling operation. The new parameter-free self-attention operation makes the model to focus on learning important parts of images. The group convolution achieves model compression and multibranch fusion. Using the main parts, the proposed network enables efficient plankton classification on mobile devices. The performance of the proposed network is evaluated on the Plankton dataset collected by Oregon State University's Hatfield Marine Science Center. The results show that TANet outperforms other deep models in speed (31.8 ms per image), size (648 kB, the size of the hard disk space occupied by the model), and accuracy (Top-1 76.5%, Top-5 96.3%).

1. Introduction

Plankton is an essential component of marine life. Phytoplankton (various drifting plants and bacteria) and zooplankton (animal plankton) are well-known types of plankton. Studying the distribution of plankton has important scientific and ecologic value. In this direction, the first attempt is to collect, identify, and record plankton samples manually. However, the high cost of time and human labors are drawbacks of this method. To alleviate the drawbacks, image-capturing tools, e.g., underwater sensor and cameras, have been utilized by which a large number of plankton videos and images are obtained easily, and the related computer vision technology that helps these multimedia materials to be analyzed has been greatly demanded.

Many traditional computer vision algorithms have been proposed in the plankton classification. Tang et al. [1] used several new shape descriptors and a normalized multilevel dominant eigenvector estimation method to select the best feature set from the binary plankton image, which achieved a 91% accuracy. Li et al. [2] proposed a pairwise nonparametric discriminant analysis for binary plankton image recognition, adding discriminant information to the classifier to achieve a 95.06% accuracy. Although these traditional algorithms can achieve high accuracy in the small dataset (3000 images including 7 different species), they are limited in time and accuracy on a large-scale dataset.

Since 2012, CNN [3] has achieved great success in a variety of visual tasks [4–7], which was also introduced in large-scale plankton recognition tasks. Li et al. [8] applied the deep residual network on the Plankton dataset, which is only a simple application. Ouyang et al. [9] proposed a particularly complex CNN framework with pyramid features architecture and compared entropy loss with other baseline models. However, the final accuracy of the proposed method has not been mentioned. On the one hand, for the case of underwater plankton observation, a tiny model is essential for the usage of mobile devices with limited chip storage space and computing power. But, on the other hand, deep models mentioned above are memory intensive and time-consuming. In addition, the deep convolutional neural network has not been well utilized in the field of underwater plankton recognition which motivated us to revisit plankton classification problem.

In this paper, we propose a new network based on the attention mechanism to address the plankton classification

problem for submarine observation. Group convolution was applied into our network in order to tradeoff between representation capability and computational cost. The number of convolution groups not only affects the size and speed of the model but also affects the accuracy of the result. To meet the needs of processing the large-scale plankton dataset, our experiments are based on the Plankton dataset [10] used by the National Data Science Bowl, a data science competition hosted on the Kaggle platform. This dataset consists of 30336 images, including 121 species of plankton. We find the optimal number of groups by a sequence of experiments. Using the attention mechanism, the network feature extraction capabilities are enhanced. Finally, our model has achieved a top-1 accuracy of 76.5333% and top-5 accuracy of 96.2666% in a small mode size of 648 kB and an inference time of 31.8 ms, which satisfy the need of low storage space and real-time classification on mobile devices.

The main contributions of this paper are as follows:

(i) We introduce a reduction module into our network to decrease information loss caused by pooling operation so as to improve the accuracy of the plankton classification task.

(ii) The group convolution is considered in the network to achieve model compression and multibranch fusion showing 4.0% improvement of top-1 accuracy when model size is 648 kB. In addition, the process of obtaining the optimal groups of group convolution for the accuracy/size tradeoff is explained in detail.

(iii) A parameter-free attention operation is proposed and 1.4% improvement of top-1 accuracy for the 648 kB model size is achieved.

The rest of the paper is organized as follows. Section 2 presents works related to efficient networks and attention mechanism. Section 3 gives the whole network architectures and describes the three main parts of the model: reduction module, group convolution operation, and self-attention module. Section 4 shows the evaluation on plankton dataset and the details of the parameter design exploration. Finally, the paper closes with conclusions in Section 5.

2. Related Works

2.1. Development of Efficient Networks. To improve the recognition accuracy, deep models become deeper and wider, which requires large chip storage space and computing cost. However, for mobile applications, under the premise that the accuracy is satisfied, a deep model with small size and low running time is necessary. Therefore, many researchers have shifted their attention to reduce the model size and improve the model efficiency.

Model compression is one of the methods to realize efficient networks. The singular value decomposition algorithm is used for pretrained models to speed up the test-time evaluation of large convolutional networks in [11]. Network pruning [12] replaces the parameters under a certain threshold with 0 and then tunes them within a few iterations. The channels of CNN models can also be pruned to reduce computation [13]. The deep compression [14] combines network pruning with quantization and Huffman encoding which can compress the model significantly.

In addition to apply the model compression to the pretrained models, we can also design a tiny and efficient network from the primary stage. Researchers [15–18] use 1×1 kernel to limit the input channels of large kernels which reduces model parameters and computational cost. This approach has been widely used in the recent literatures [19, 20]. SqueezeNet [21] reduces parameters and computation significantly while maintaining AlexNet-level accuracy. SENet [22] introduces an architectural unit that boosts performance at slight computation cost. Group convolution can prune the redundant connections by dividing channels into groups. In this direction, some research work has been done, e.g., see [23, 24]. The so-called MobileNet approach proposed in [25] utilizes the depthwise separable convolutions. This approach gains state-of-the-art results among lightweight models.

2.2. Attention Mechanism. The human perception process [26] shows the significance of the attention mechanism. The mechanism gets the focus area by fast scanning the global picture. The Residual Attention Network introduced in [27] is a convolutional network that adopts the mixed attention mechanism in very deep structure achieving very high classification accuracy in the image classification task. The nonlocal neural network method presented in [28] shows a high performance in the video classification task by the use of the self-attention mechanism. Global context attention is widely used in the semantic segmentation area. DFN [29] embeds the global average pooling branch in the top to extend the U-shape architecture. EncNet [30] introduces an encoding layer with a SENet-like module to capture encoded semantics and predict scaling factors that are conditional on these encoded semantics. Inspired by abovementioned work, we applied the attention mechanism in our work.

3. Proposed Approach

In this section, we introduce the reduction module and group convolution operation which is meant to reduce the model parameters and improve the efficiency. Then, the self-attention module is given to improve the feature learning ability. The entire architectures along with the effectiveness proof of network architecture design come in the end of this section.

3.1. Reduction Module. Pooling operation is indispensable in the image classification task. It makes the network invariant for translation but leads to a serious information loss. Inspired by Inception V3 [17], we apply the reduction module to avoid the representational bottleneck.

We can concatenate the outputs of two parallel stride 2 blocks (pooling and convolution layer) to replace a simple pooling operation. This process enables down sampling with alleviating information loss and maintaining translation invariance of the network. To keep the number of channels

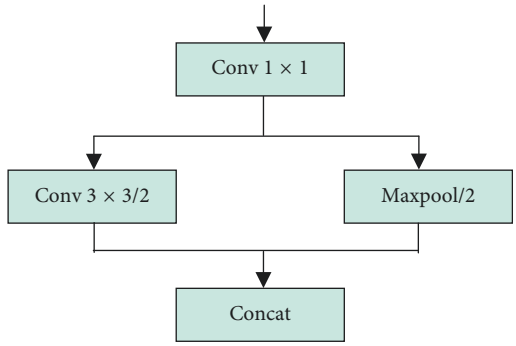

Figure 1: Reduction module. The "Concat" operation merges two feature maps with dimensions of $[w, h, c_1]$ and $[w, h, c_2]$ into one feature map, whose dimension is $[w, h, c_1 + c_2]$. w and h represent feature map sizes, and c is the number of feature map channels. "/2" indicates that the convolution or pooling operation has a stride of 2.

unchanged, we first use a 1×1 convolution to halve the number of channels as shown in Figure 1.

Images in the Plankton dataset are grayscale images, which contain less color information compared to RGB images (e.g., ImageNet). Therefore, the information loss caused by the pooling operation might bring serious impact on the model accuracy in the plankton classification task.

3.2. Group Convolution. Although the model size of SqueezeNet [21] is small, it can be compressed further by deep compression [14] from 4.8 M to 0.47 M. This shows the fact that the deep neural network is extremely sparse. The standard convolution is a dense convolution, and each convolution kernel convolutes all channels of the feature map. Due to the sparseness of deep networks, we choose to use group convolution instead of standard convolution.

Zhao et al. [31] show that increasing the fusion number of the deeply fused network can improve network performance. Since the deeply fused network integrates the power of multiple branches, a single network can bring the effect of multimodel fusion. We try to increase the number of branches as shown in Figure 2(b). Each group in the group convolution is a branch, so we can increase the fusion number by increasing the number of groups as shown in Figure 2(c).

Group convolution can also improve the network efficiency significantly. Standard convolutions have the computational cost of

$$M \times K \times K \times S \times S \times N. \tag{1}$$

The computational cost depends on the number of input channels M, the kernel size $K \times K$, the feature map size $S \times S$, and the number of output channels N.

Group convolution whose group number is F has the computational cost:

$$\frac{M}{F} \times K \times K \times S \times S \times N. \tag{2}$$

The computational cost of the standard convolutions is the F times the group convolutions:

$$\frac{M \times K \times K \times S \times S \times N}{(M/F) \times K \times K \times S \times S \times N} = F. \tag{3}$$

The group number F is a parameter that can be adjusted. To tradeoff between accuracy and computational cost, a series of experiments were designed to explore the optimal value of F. We present it in detail in Section 4.3.

Group convolution has been widely used recently, e.g., in [23–25]. But the procedure for designing the group number so as to make the model more accurate has not been mentioned concretely. In our work, the process of obtaining the optimal number of groups is given in detail.

3.3. Self-Attention Module. The attention mechanism does not consider every pixel in an image for the classification task. Some pixels are useless (background) for recognition, and some pixels are crucial (foreground). Therefore, utilizing of the attention mechanism inclines the convolution neural network to note the most critical region of an image.

The general attention mechanism predicts important areas in the feature map to obtain the attention weight by some convolutional layers and then multiply the attention weight by the original feature map. The attention weight has a large value for important parts and small value for other parts so that the network is able to notice the important parts by the attention mechanism operation.

The standard attention mechanism needs to add new convolutional layers to learn the weights, but the feature map itself contains the importance information of each pixel. As a case, network pruning [12] replaces the parameters below a certain threshold with 0 directly for the bigger data is more important. Intuitively, areas in feature map with large activation values impact the output more significantly which means these pixel values are more important. Therefore, we normalize the feature map as the attention weight directly so as to avoid adding parameters which can be considered as self-attention. Thus, output $\widehat{Y}$ of attention module is modified as

$$\begin{aligned} \widehat{Y}(x) &= Y(x) \times M(Y(x)), \\ M(Y(x)) &= \frac{1}{1 + e^{-Y(x)}}, \end{aligned} \tag{4}$$

where x denotes the input, $M(x)$ is the attention weight, and $Y(x)$ is the output of the main branch.

If y_1 and y_2 represent two of the output data $Y(x)$ and $y_1 > y_2$, then we have

$$\begin{aligned} \frac{y_1}{y_2} &= k, \\ \frac{M(y_1)}{M(y_2)} &> 1. \end{aligned} \tag{5}$$

Or, equivalently,

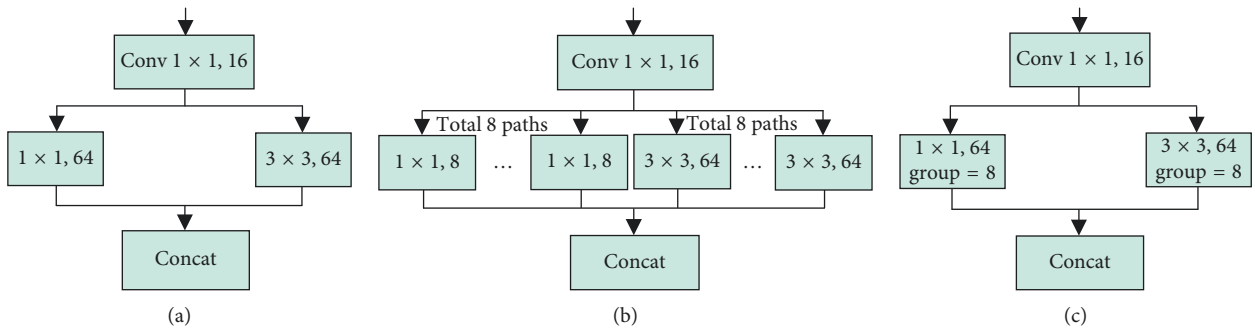

Figure 2: (a) Fire module defined in SqueezeNet. 1×1 and 3×3 represent the convolutional kernel size. 16 and 64 represent the number of filters. (b) Multibranch fire module. While keeping the number of channels unchanged, we split one branch in (a) into eight branches. (c) Group fire module. Group convolution operation has multibranch effect as (b). 8 indicates the number of convolution groups, corresponding to F in formulas (2) and (3).

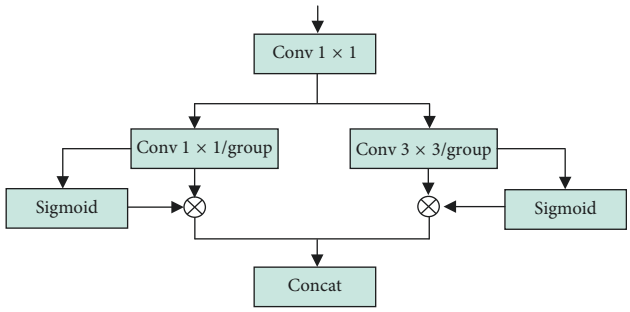

Figure 3: Attention module. "⊗" denotes the element-wise multiplication. "/group" indicates the group convolution. "Sigmoid" indicates the sigmoid normalization.

$$\frac{\widehat{y_1}}{\widehat{y_2}} = \frac{y_1 \times M(y_1)}{y_2 \times M(y_2)} = k\frac{M(y_1)}{M(y_2)} > k. \tag{6}$$

The self-attention module expands the gap between data that make it easier to notice the important parts as shown in Figure 3.

3.4. Assembling Architectures and Evaluation. The final architecture of tiny attention network is shown in Figure 4(c). By adding the group convolution (Figure 4(a)), self-attention module (Figure 4(b)), and reduction module (Figure 4(c)) to the network, the accuracy increases correspondingly on the Plankton dataset which proves the effectiveness and reasonableness of the architecture design as shown in Table 1.

More experiments to prove the effectiveness of network design are presented in Section 4.1.

4. Experiments

4.1. Experiments on Plankton Dataset. The Plankton dataset consists of 30,336 grayscale images including 121 kinds of planktons. This dataset is used for the National Data Science Bowl, a data science competition hosted on the Kaggle platform. We took 3037 images as a test set, 3037 images as a validation set, 24,262 images as a training set, and rescaled them to 256×256. The images are augmented by rotating and flipping operations to avoid the overfitting problem as shown in Figure 5.

Experiments are based on the caffe [32] framework. The model size indicates the size of the hard disk space actually occupied by the caffe model. Inference time is measured on the NVIDIA GeForce Titan X Pascal 12 GB. All models are trained from scratch on the Plankton dataset.

First, SqueezeNet and DarkNet are both designed for a mobile system. DarkNet and SqueezeNet are classic small networks with high accuracy designed specifically for hardware with limited memory (e.g., FPGA and mobile system). SqueezeNet achieves AlexNet-level accuracy on ImageNet with 50x fewer parameters. Second, SqueezeNet and DarkNet are well known to be used as backbone for various tasks (e.g., MobileNets and YOLOv3). As a result, SqueezeNet and DarkNet are candidates as tiny models to be compared with TANet.

Compared to DarkNet [19] and SqueezeNet [21], the TANet is superior in accuracy and speed. The TANet model size is only 648 kB as shown in Table 2. We deployed TANet on iPhone 6s (32 G), and the actual inference time is only 22.4 ms which fully satisfies the need of low storage space and real-time classification on mobile devices.

Several classic models (VGG-19 and ResNet-18) were also applied to the Plankton dataset. Similarly, the proposed model shows a comparative performance in both accuracy and speed as shown in Table 3.

The performance of the VGG-19 model in comparison with other models is not desirable which might be caused by the special trait of the Plankton dataset. There are two reasons that the Plankton dataset is relatively simple compared to the ImageNet. First, the Plankton dataset contains only 121 classes, which is a small number of categories compared to the 1000 classes of ImageNet. Second, since the Plankton dataset is grayscale images, then it contains less color information in comparison with RGB images. Due to these specialties of the Plankton dataset, the VGG-19 model does not cover very well. Table 2 shows that the number of convolution groups has a significant effect on the model speed, size, and accuracy. By eight groups, the model is relatively balanced in terms of accuracy, speed, and size, so we consider it as the model of TANet.

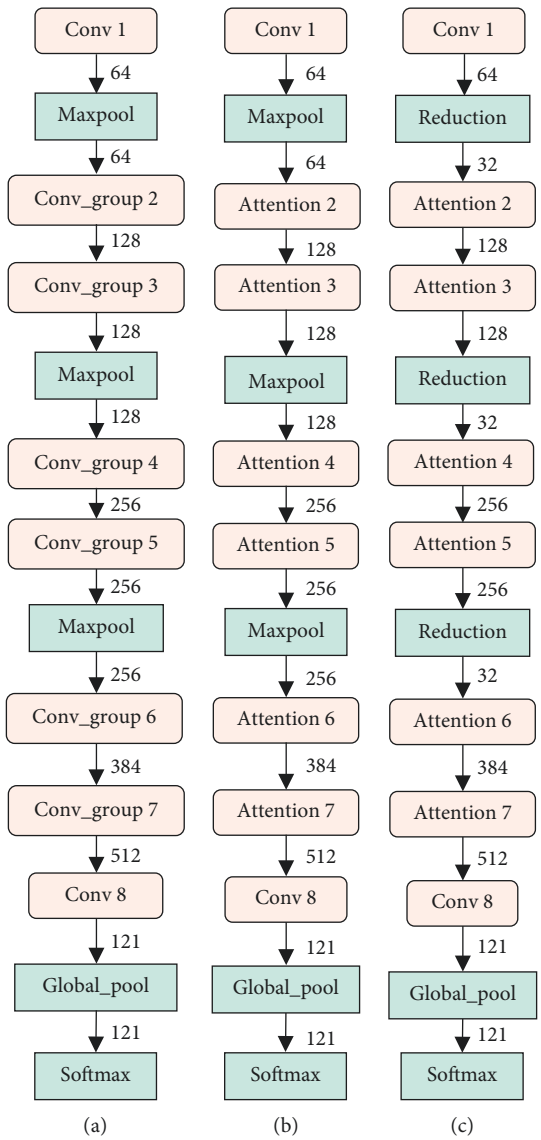

Figure 4: Tiny Attention Network (TANet) architecture. (a) CNN + Group Convolution. (b) (a) + self-attention. (c) (b) + reduction module (TANet).

Table 1: Design architectures and evaluation on plankton dataset. Our experiments are based on the caffe [32] framework, and the model size indicates the size of the hard disk space actually occupied by the caffe model. Inference time is measured on the NVIDIA GeForce Titan X Pascal 12 GB.

Network	Top-1 (%)	Top-5 (%)	Inference time (ms)	Model size (kB)
Figure 4(a)	75.9	95.6	21.6	642
Figure 4(b)	76.1	95.9	27.3	642
Figure 4(c) (TANet)	**76.5**	**96.3**	31.8	648

In addition, the effectiveness of the attention mechanism and multibranch model (group convolution) were demonstrated by additional comparative experiments as shown in Figure 6. Different convolution groups result in different sizes of TANet (model 3). For fairness, we transform model 1, i.e., TANet without the attention mechanism and group convolution, and model 2, i.e., TANet without group convolution, to the same size as model 3 by changing the number of channels in model 1 and model 2.

The accuracy of model 1 and model 2 decreases as the model size decreases. However, the accuracy of model 2 is always higher than that of model 1. The reason is that model 2 uses the attention mechanism. Specifically, the use of attention mechanism achieved a 1.4% improvement when the model size is 648 kB, which is a significant improvement without adding any parameters.

Although the multibranch model is only a single model, it merges multiple branches from the low layer to the high layer multiple times. Thus, it has the effect of multimodel fusion. As shown in Figure 6, by the increase of the number of convolution group, the size of model 3 decreases rapidly. It achieves a 4.0% improvement of top-1 accuracy when model size is 648 kB for the use of group convolution.

(a) (b) (c) (d)

Figure 5: Data augmentation. (a) Original image. (b) Rotated 90° from (a). (c) Flipped from (a). (d) Rotated 90° and flipped from (a).

Table 2: Tiny model comparison on test set.

Network	Top-1 (%)	Top-5 (%)	Inference time (ms)	Model size	Reduction in model size
DarkNet	74.0	94.9	122.4	3.54 M	1x
SqueezeNet	75.8	94.5	112.2	3 M	1.2x
TANet (group = 2)	**77.6**	**96.3**	**28.8**	1.15 M	3.1x
TANet (group = 8)	76.5	**96.3**	31.8	648 kB	5.6x

Table 3: Multiple classic model comparison on test set.

Network	Top-1 (%)	Top-5 (%)	Inference time (ms)	Model size (M)	Reduction in model size
VGG-19	73.2	95.0	148.76	534.4	1x
ResNet-18	77.3	95.7	77.26	43.64	12x
TANet (group = 2)	**77.6**	**96.3**	**31.8**	1.15	465x

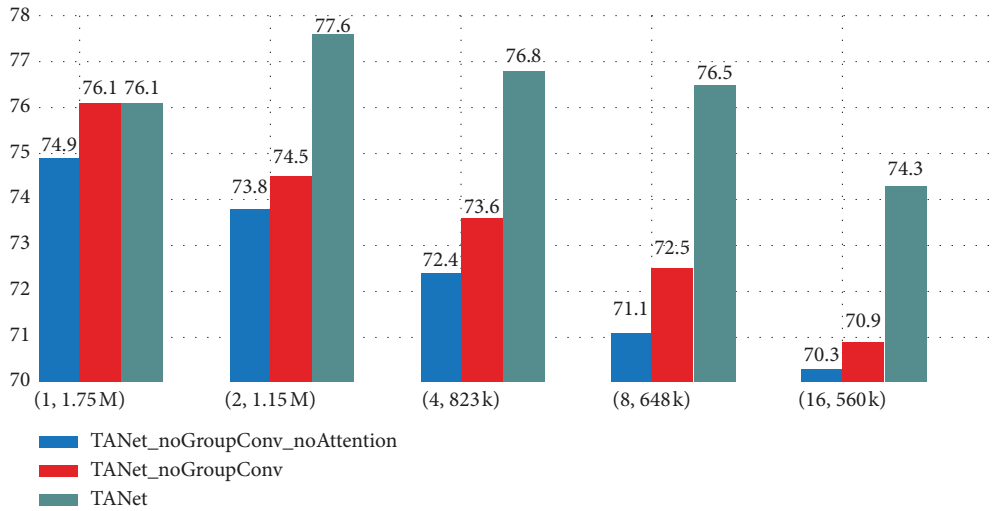

Figure 6: A comparison of top-1 accuracy between TANet without attention mechanism and group convolution (model 1), TANet without group convolution (model 2), and TANet (model 3). Different convolution groups will get different sizes of TANet (model 3). By changing the number of channels in model 1 and model 2, models with the same size as model 3 can be obtained. The numbers within the parentheses in the x-axis indicate (the number of convolution groups, the number of model sizes).

4.2. Model Comparison with Competition Winner. The data used in our experiments are downloaded from the plankton recognition competition on Kaggle, named National Data Science Bowl. The model ensemble is commonly used to improve accuracy in challenges. For instance, in this competition, the third-ranked team named Poisson Process averaged seventeen models in their final results. The second-ranked team, called the Happy Lantern Festival, also merged at least four models. We reproduced the four models of the Happy Lantern Festival team and compared to our single model. The results are shown in Table 4.

The performance of these four models mainly depends on the number of parameters because their networks are just a stack of common convolutional layers. In the aspect of accuracy, speed, and model size, the TANet outperforms all of these four networks.

Table 4: Model comparison between TANet and the four models of the second-ranked team [33] in the competition. The filter sizes of the four models are 2×2, 3×3, 4×4, and 5×5.

Network	Top-1 (%)	Inference time (ms)	Model size	Reduction in model size
Model 1 (filter size 2×2)	76.6	153.7	122.4 M	1x
Model 2 (filter size 4×4)	75.2	103.0	74.3 M	1.6x
Model 3 (filter size 3×3)	74.7	105.0	61.5 M	2x
Model 4 (filter size 5×5)	73.6	346.0	30.4 M	4x
TANet (group = 2)	77.6	28.8	1.15 M	106x
TANet (group = 8)	76.5	31.8	648 kB	194x

Table 5: Experiment 1 for the group number design of group convolution on plankton data (test set).

Network (groups)	Top-1 (%)	Top-5 (%)	Model size (kB)
TANet (8x)	**76.7999**	**95.9666**	733
TANet (4x)	76.3667	95.7666	603
TANet (2x)	75.4667	95.6666	538
TANet (1x)	75.6	95.6666	505

TANet (kx) means k channels are convoluted as a group for all layers.

4.3. Parameter Design Exploration. In the model design, the number of convolution groups is a parameter that can be adjusted. The adjustment not only affects the accuracy but also affects the model size and the efficiency. We did some experiments to find the optimal group value. In Experiment 1, the number of channels for each group of all layers was set to be the same, but the number of groups varies for different layers. On the contrary, in the Experiment 2, the number of channels in each group of different layers was set to be different, and the number of groups was the same. The results of Experiments 1 and 2 are shown in Tables 5 and 6, respectively.

Two conclusions can be drawn from the results of experiments. First, as shown in Table 6, when the number of the group is 2, top-1 reaches 77.5666%, which is much higher than the result of without grouping (76.0666%). Group convolution can extract the main features between specific channels and brings the effect of clustering. Therefore, grouping channels for convolution not only reduces parameters but also improves network accuracy.

Second, the result of experiment 2 shows that the settings of the network results in a more efficient output. Therefore, it can be concluded that using the same number of groups for all layers makes the network more effective. To balance the relationship between accuracy and model size, it seems that the model with the eight groups can be considered as the first candidate for TANet as shown in Table 6.

5. Conclusions

In this paper, we proposed a tiny network called TANet based on the self-attention mechanism and group convolution for the plankton classification task. The reduction module was applied to reduce the information loss caused by pooling operation; and the group convolution to compress the model size. Self-attention was utilized to improve the feature learning ability. The proposed model can be applied to real-time submarine observation efficiently.

Table 6: Experiment 2 for the group number design of group convolution on plankton data (test set).

Network (groups)	Top-1 (%)	Top-5 (%)	Model size
TANet (1)	76.0666	95.3666	1.75 M
TANet (2)	**77.5666**	**96.2666**	1.15 M
TANet (4)	76.8333	96.2666	823 kB
TANet (8)	76.5333	96.2666	648 kB
TANet (16)	74.2667	96.2666	560 kB

TANet (k) means that the channel of the feature map is divided into k groups for all layers.

Data Availability

The data used to support the findings of the study are available at https://www.kaggle.com/c/datasciencebowl/data.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research was partly supported by National Natural Science Foundation of China (Grant no. 41876098) and Shenzhen Science and Technology Project (Grant no. JCYJ20151117173236192).

References

[1] X. Tang, F. Lin, S. Samson, and A. Remsen, "Binary plankton image classification," *IEEE Journal of Oceanic Engineering*, vol. 31, no. 3, pp. 728–735, 2006.

[2] Z. Li, F. Zhao, J. Liu, and Y. Qiao, "Pairwise nonparametric discriminant analysis for binary plankton image recognition," *IEEE Journal of Oceanic Engineering*, vol. 39, no. 4, pp. 695–701, 2014.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, Las Vegas, NV, USA, June 2015.

[5] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA, June 2014.

[6] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks,"

IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 6, pp. 1137–1149, 2017.

[7] T. Alexander and C. Szegedy, "Deeppose: human pose estimation via deep neural networks," in *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1653–1660, Columbus, OH, USA, June 2014.

[8] X. Li and Z. Cui, "Deep residual networks for plankton classification," in *Proceedings of the Oceans 2016 MTS/IEEE Monterey*, pp. 1–4, Monterey, CA, USA, September 2016.

[9] P. Ouyang, H. Hu, and Z. Shi, "Plankton classification with deep convolutional neural networks," in *Proceedings of the 2016 IEEE Information Technology, Networking, Electronic and Automation Control Conference*, pp. 132–136, IEEE, Chongqing, China, May 2016.

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

[11] E. Denton, W. Zaremba, J. Bruna, Y. Lecun, and R. Fergus, "Exploiting linear structure within convolutional networks for efficient evaluation,"pp. 1269–1277, Columbus, OH, USA, June 2014.

[12] J. Tran, J. Tran, J. Tran, and W. J. Dally, "Learning both weights and connections for efficient neural networks," in *Proceedings of the International Conference on Neural Information Processing Systems*, pp. 1135–1143, Montreal, Canada, December 2015.

[13] Y. He, X. Zhang, and J. Sun, "Channel pruning for accelerating very deep neural networks," in *Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy, October 2017.

[14] S. Han, H. Mao, and W. J. Dally, "Deep compression: compressing deep neural networks with pruning, trained quantization and huffman coding," *Fiber*, vol. 56, no. 4, pp. 3–7, 2015.

[15] C. Szegedy, W. Liu, Y. Jia et al., "Going deeper with convolutions," in *Proceedings of the Computer Vision and Pattern Recognition*, pp. 1–9, Boston, MA, USA, June 2015.

[16] I. Sergey and C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift," in *Proceedings of the International Conference on International Conference on Machine Learning*, pp. 448–456, Lille, France, July 2015.

[17] C. Szegedy, V. Vincent, Sergey Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the Computer Vision and Pattern Recognition*, pp. 2818–2826, Las Vegas, NV, USA, June 2016.

[18] C. Szegedy, Sergey Ioffe, V. Vincent, and A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Proceedings of the Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, June 2016.

[19] R. Joseph, "Darknet: open source neural networks in c," 2016, http://pjreddie.com/darknet/.

[20] S. Xie, R. Girshick, P. Dollar, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5987–5995, Honolulu, HI, USA, July 2017.

[21] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "SqueezeNet: alexnet-level accuracy with 50x fewer parameters and <0.5 mb model size," 2016, https://arxiv.org/abs/1602.07360.

[22] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, June 2018.

[23] F. Chollet, "Xception: deep learning with depthwise separable convolutions," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1800–1807, Honolulu, HI, USA, July 2017.

[24] X. Zhang, X. Zhou, M. Lin, and J. Sun, "Shufflenet: an extremely efficient convolutional neural network for mobile devices," in *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, June 2018.

[25] A. G. Howard, M. Zhu, B. Chen et al., "Mobilenets: efficient convolutional neural networks for mobile vision applications," in *Proceedings of the Computer Vision and Pattern Recognition*, Honolulu, HI, USA, July 2017.

[26] V. Mnih, N. Heess, A. Graves, and K. Kavukcuoglu, "Recurrent models of visual attention," in *Proceedings of the Machine Learning*, vol. 3, pp. 2204–2212, Bejing, China, June 2014.

[27] F. Wang, M. Jiang, C. Qian et al., "Residual attention network for image classification," in *Proceedings of the Computer Vision and Pattern Recognition*, pp. 6450–6458, Honolulu, HI, USA, July 2017.

[28] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proceedings of the Computer Vision and Pattern Recognition*, Honolulu, HI, USA, July 2017.

[29] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, "Learning a discriminative feature network for semantic segmentation," in *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, June 2018.

[30] H. Zhang, K. Dana, J. Shi et al., "Context encoding for semantic segmentation," in *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, June 2018.

[31] L. Zhao, J. Wang, X. Li, Z. Tu, and W. Zeng, "On the connection of deep fusion to ensembling," in *Proceedings of the Computer Vision and Pattern Recognition*, Honolulu, HI, USA, July 2017.

[32] https://github.com/BVLC/caffe.

[33] X. Cao, "A practical theory for designing very deep convolutional neural networks," Tech. Rep., 2015.