**HW2 313553041 盧育霆**

**Introduction:**

In this homework, we were asked to perform house number recognition. First, we need to locate the numbers in the image and find appropriate bounding boxes to enclose them. Next, we need to recognize the numbers in the image. For the model requirements, we are only allowed to use the Faster R-CNN architecture. We can only modify the backbone network, region proposal network, and head. In this homework, I made adjustments to the region proposal network and do some data augmentation. Expert to improve the performance.

Github:

**Method:**

**ResNext:**

The main design of ResNeXt is to group high-dimensional convolutional layers into multiple identical convolutional layers for convolution operations. As shown in the figure below, ResNeXt splits the convolutional layer with a dimension of 64 into 32 convolutional layers, each with a dimension of 4. The paper calls these split groups as "cardinality." In their result, the results indicate that increasing cardinality is more beneficial for improving model accuracy than increasing width and depth. So, I use ResNext50 as my backbone model.

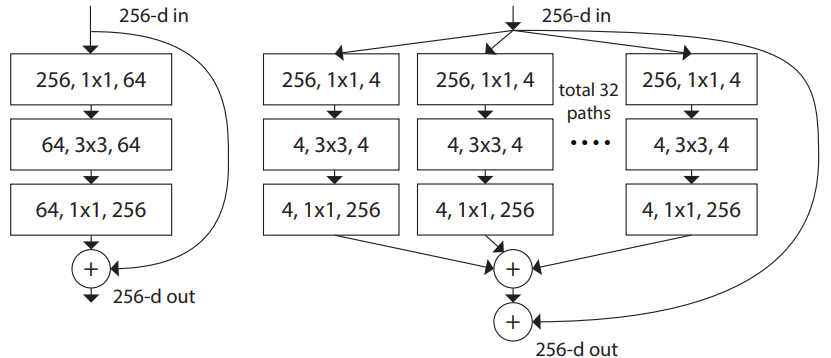


Figure 1. ResNext cardinality.

**CBAM:**

In CNNs, we have many feature maps, but not all of them contain useful information. Therefore, CBAM uses an attention mechanism to filter feature maps. As shown in the figure below, CBAM consists of two attention mechanisms: channel attention and spatial attention. Channel attention emphasize important channels, suppress unimportant channels. And, spatial attention highlights significant spatial regions, and ignore irrelevant background information.

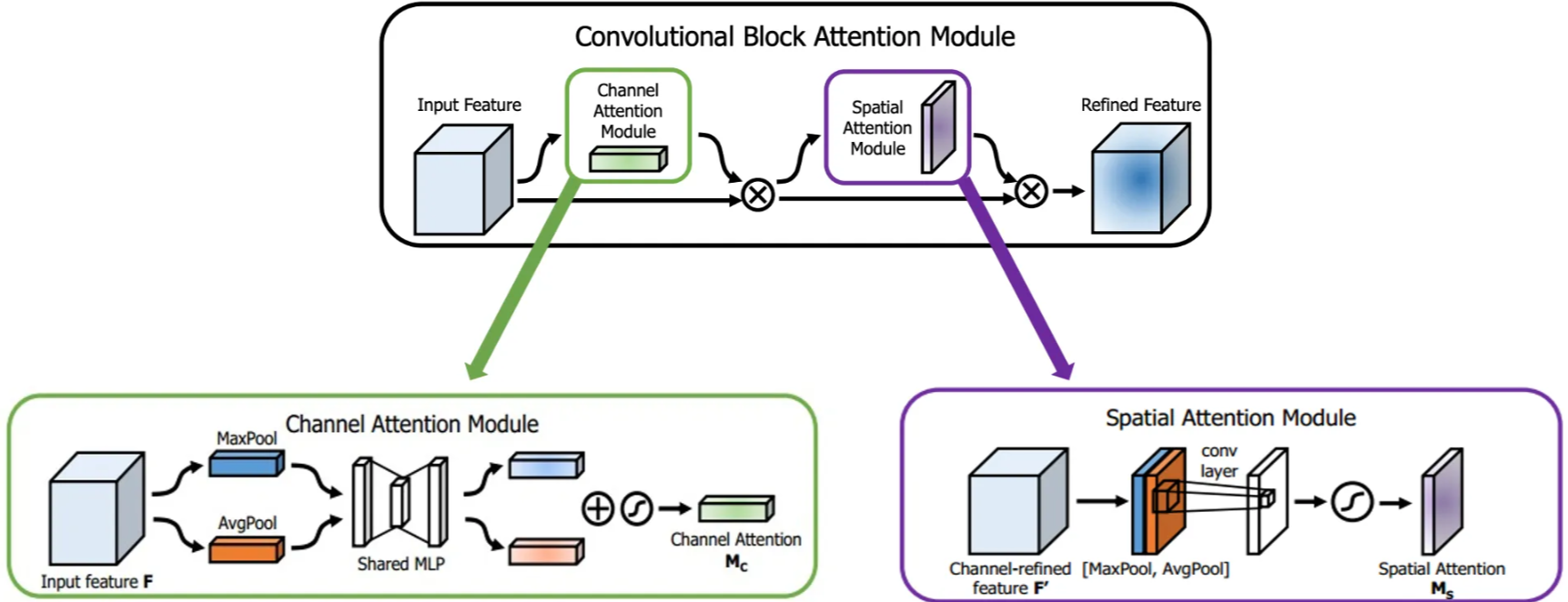


Figure 2. CBAM.

**Contrastive loss:**

Contrastive loss is commonly used to pulls features of the same class closer together and pushes features of different classes farther apart.

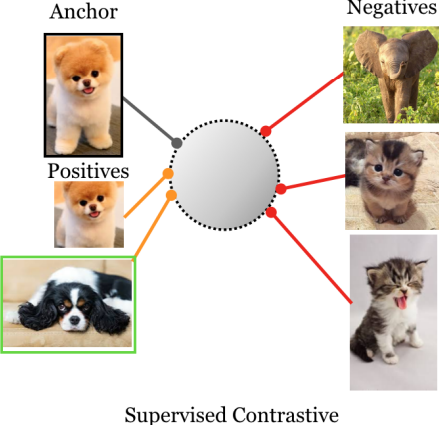


Figure 3. Contrastive learning.

**My model architecture:**

As shown in the figure below, I add CBAM after each ResNext layer.

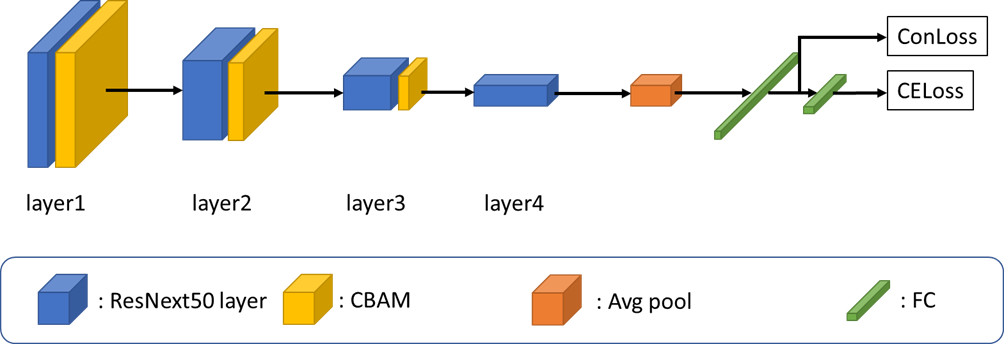


Figure 4. My model structure.

**Pre-Process:**

First, I resize each image size to (600, 600), and do random crop to size (224, 224). then, I do the “AutoAugment” data augmentation in ImageNet policy in transforms module.

AutoAugment: a series of data augmentation operations. Include, ShearX/Y, TranslateX/Y, rotate +-30 degrees, brightness, contrast, sharpness, color, posterize, solarize, equalize, invert, cutout.

The reason I chose the ImageNet policy, after analyzing the dataset, I found that the classification categories are highly diverse, similar to ImageNet.

**Hyperparameters:**

Batch size = 128, learning rate = 0.00005, epoch = 300, optimizer: Adam.

**Results:**

In Figure 5, it can be observed that around epoch 50, the training loss has stabilized. Finally, it converges to around 0.06. In Figure 6, we can observe the accuracy trend. The validation accuracy reaches near-optimal performance around epoch 50. The final training accuracy reaches approximately 99%. So, based on the charts, there may still be some overfitting. Or, it may be necessary to modify the model further to obtain more discriminative features. It may also be because the random crop did not capture the important parts of the image, so a better approach is needed to address this issue.

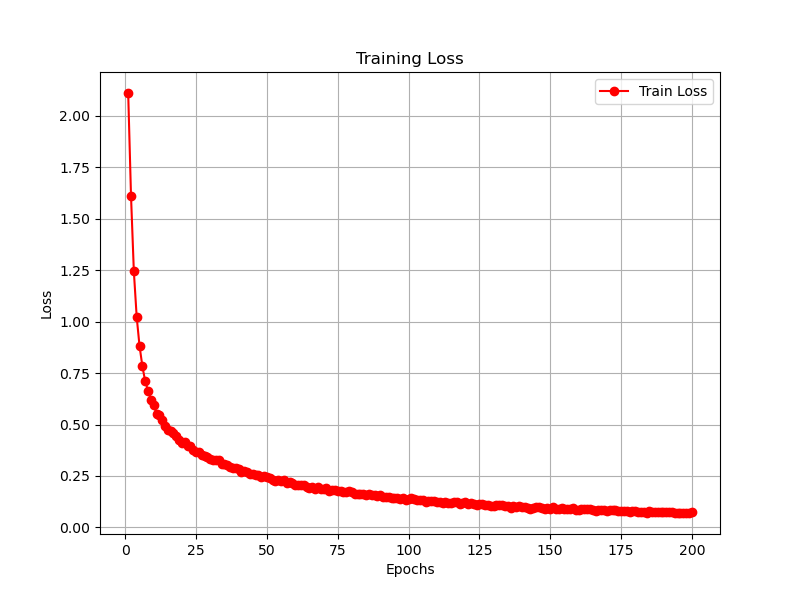


Figure 5 Loss plot.



Figure 6. Training and Validation accuracy

At the beginning, I experimented with the basic ResNet-50 and ResNeXt-50. I only applied data augmentation techniques including RandomCrop, HorizontalFlip, and Rotation. The results are shown in Table I. Thus, it can be observer that splitting channels into cardinality can get better performance. Therefore, I chose ResNeXt-50 as my backbone.

**Table I**

|  |  |
| --- | --- |
| Model | Accuracy |
| Resnet50 | 80.0000 |
| Resnext50 | 84.0000 |

To improve performance, I analyzed the dataset. I observe that the dataset has a very diverse range of categories, so it is similar to ImageNet. First, I applied AutoAugment to the dataset. I can use some of the ImageNet data augmentation strategies in AutoAugment.

Next, I attached CBAM after each ResNeXt layer. It is expected to further amplify the features of important channels. Because CBAM includes both channel attention and spatial attention. So, I tested the performance with and without spatial attention. It can be observed that after applying spatial attention, the spatial features are enhanced, leading to a slight increase in accuracy.

In Table II, it can be observed that AutoAugment achieves the best performance. Based on the experimental results, adding CBAM may not be the best approach. Or, it might be worth trying to add CBAM to other layers of the network. Although AutoAugment achieved the best performance, the performance on the test dataset is lower compared to the model with CBAM.

In the loss function, in addition to the basic cross entropy, I also added contrastive loss. It was originally expected to pull similar features closer together and push dissimilar features farther apart. However, based on the experimental results, the performance did not improve.

**Table II**

|  |  |
| --- | --- |
| Model | Accuracy |
| ResNext50 + Simple augmentation | 84.0000 |
| ResNext50 + AutoAugment | 91.6667 |
| ResNext50 + CBAM w/o spatial | 88.6667 |
| ResNext50 + AutoAug + CBAM w/o spatial | 90.0000 |
| ResNet50 + AutoAug + CBAM w/o spatial | 89.3333 |
| ResNext50 + AutoAug + CBAM with spatial | 90.6667 |
| ResNext50 + AutoAug + CBAM with spatial + contrastive loss | 90.6667 |

**References:**

**[1]** S. Xie, R. Girshick, P. Dollár, Z. Tu and K. He, "Aggregated Residual Transformations for Deep Neural Networks," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 2017, pp. 5987-5995

**[2]** Woo, S., Park, J., Lee, JY., Kweon, I.S. (2018). “CBAM: Convolutional Block Attention Module”. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds) Computer Vision – ECCV 2018. ECCV 2018. Lecture Notes in Computer Science(), vol 11211. Springer, Cham. <https://doi.org/10.1007/978-3-030-01234-2_1>

**[3]** Prannay Khosla and Piotr Teterwak and Chen Wang and Aaron Sarna and Yonglong Tian and Phillip Isola and Aaron Maschinot and Ce Liu and Dilip Krishnan. “Supervised Contrastive Learning”, 2020 arXiv preprint arXiv:2004.11362

**[4]** ResNext:

<https://github.com/miraclewkf/ResNeXt-PyTorch/blob/master/resnext.py>

**[5]** CBMA:

<https://github.com/Jongchan/attention-module/blob/master/MODELS/cbam.py>