Assignment-2

Deep Learning

Wen-Hai Tseng
RE6124027
Nation Cheng Kung University
w13281328@gmail.com

Abstract*—This assignment has two main objectives: The first is to design a special convolutional module that is spatial size invariant and can handle an arbitrary number of input channels, focusing solely on this special module rather than every layer of the CNN. The second objective is to design a (2-4)-layer CNN, Transformer, or RNN network that can achieve 90% of the performance of ResNet34 on ImageNet-mini.

I. Introduction

This assignment requires designing a special convolutional module that is spatial size invariant and can handle an arbitrary number of input channels, allowing for input channels of different sizes. We need to compare our method to see if it can improve computational speed or accuracy. Additionally, we need to configure a 2-4 layer CNN, Transformer, or RNN network, and the performance of this model must reach 90% of the performance of ResNet34.

II. Method

This section will introduce the feature extraction methods used, including BRISK, ORB, ResNet50, and HOG. These feature extraction techniques enable our classification models to achieve better performance during training.

A. DynamicConv2D

The DynamicConv2D class is a custom PyTorch module designed to perform dynamic convolution operations. This module adjusts the convolution kernel weights based on the number of input channels, allowing it to handle inputs with varying channel numbers. In the initialization method, the class defines the weights and biases of the convolution kernels. The shape of the weights is (out_channels, in_channels, kernel_size, kernel_size), meaning each output channel has a corresponding weight tensor. If bias is set to True, a bias vector with a length of out_channels is created. During the forward propagation process, the weights of the convolution kernel are dynamically adjusted according to the current number of channels in the input tensor (current_in_channels). If the current number of channels is less than the specified in_channels during initialization, the corresponding part of the weights is selected; otherwise, the complete weights are used. Finally, the adjusted weights and biases are used to perform the convolution operation and return the result. This dynamic convolution operation is particularly suitable for scenarios that require handling inputs with different channel numbers, such as multi-sensor data processing or processing different feature maps of an image.

After modifying the first convolutional layer of ResNet18 to use `DynamicConv2D`, it is unnecessary to replace all the convolutional layers with `DynamicConv2D` to accommodate inputs with various channel sizes. During training, we need to use images with different channel sizes to ensure that the model maintains performance across all channel sizes. However, the fitting speed is relatively slow.

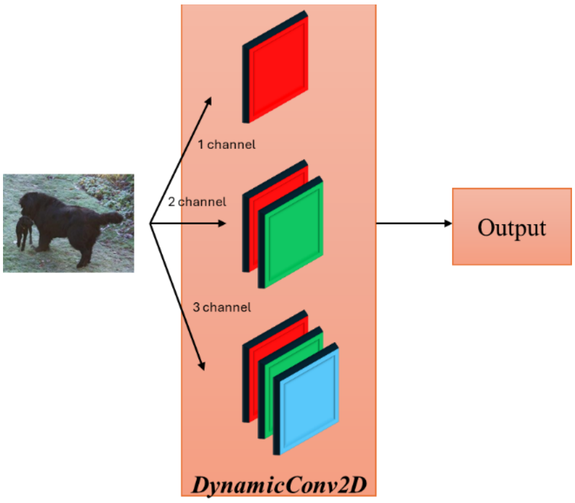


Fig.1 The structure of DynamicConv2D

The structure of DynamicConv2D is shown in Fig.1. We decompose DynamicConv2D into 7 combinations of RGB, forming 3 groups. If the input has only one channel, only one set of weights will be updated. If the input has two channels, then two sets of weights will be updated. If there are three channels, it operates normally.

B. Residual attention network

The Residual Attention Network leverages the attention mechanism, which can be applied to existing end-to-end convolutional networks. The Residual Attention Network changes feature attention by stacking attention structures. As the network deepens, the attention module adapts accordingly. In each attention module, both up-sampling and down-sampling structures are employed.

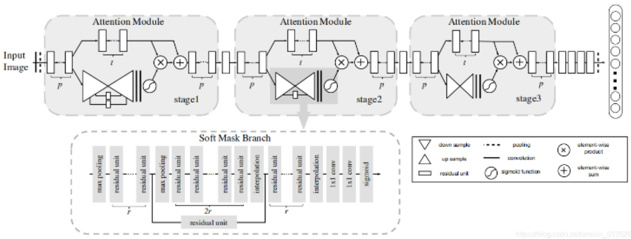


Fig.2 The structure of Residual Attention Network

We adopt the Residual Attention Network proposed by [3] as shown in Fig.2, and it has the following features:

- It is composed of multiple stacked attention modules, each capable of capturing different types of attention.

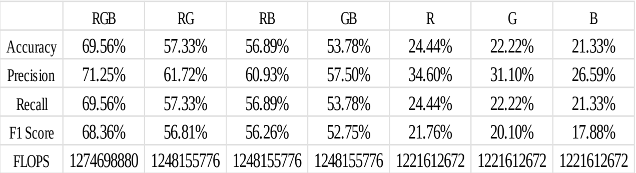
- It employs residual connections to avoid gradient vanishing, allowing it to be extended to great depths and enabling end-to-end training.

In our implementation, the Residual Attention Network uses a 2-stage attention module, followed by a fully connected layer for classification tasks.

III. EXPERMENTS AND RESULT

A. *DynamicConv2D*

We replaced the first convolutional layer of ResNet18 with DynamicConv2D to enable inputs with various numbers of channels. We conducted several experiments to evaluate the accuracy and elapsed time of our model with different input configurations, including RGB, RG, RB, GB, R, G, and B. Additionally, we compared the performance of our model with the original model. All experiments were conducted with a fixed image size of 96x96, and the models were trained for 30 epochs. During training, we applied image augmentations such as horizontal and vertical flipping, rotation, and distortion.



	RGB	RG	RB	GB	R	G	B
Accuracy	69.56%	57.33%	56.89%	53.78%	24.44%	22.22%	21.33%
Precision	71.25%	61.72%	60.93%	57.50%	34.60%	31.10%	26.59%
Recall	69.56%	57.33%	56.89%	53.78%	24.44%	22.22%	21.33%
F1 Score	68.36%	56.81%	56.26%	52.75%	21.76%	20.10%	17.88%
FLOPS	1274698880	1248155776	1248155776	1248155776	1221612672	1221612672	1221612672

Table.1 Accuracy, Precision, Recall, F1 Score and FLOPS for Inputs with Different Numbers of Channels

In the experiments shown in Table 1, we observe that the performance with three channels is better than that with two or one channel, which is not surprising. Additionally, the results for the three combinations with two channels are very close to each other, and the same is true for the single-channel results. FLOP, which stands for Floating Point Operations Per Second, is a common metric for model complexity. From Table 1, it can be observed that the FLOP for RGB 3-channel is greater than that for 2-channel, which in turn is greater than that for 1-channel. This result is expected. However, considering the overall performance, since the accuracy of 3-channel is significantly higher than that of 2-channel, it is still recommended to use 3-channel input.

	Resnet-18(DynamicConv2D)	Resnet-18
Accuracy	69.56%	51.33%
Precision	71.25%	52.68%
Recall	69.56%	51.33%
F1 Score	68.36%	50.48%
FLOPS	1274698880	333585408

Table.2 Accuracy, Precision, Recall, F1 Score and FLOPS for Resnet-18(DynamicConv2D) and Resnet-18.

From Table 2, it can be observed that under the same conditions of training for 30 epochs and identical image processing, the performance of ResNet-18 (DynamicConv2D) with 3 channels on the test set is significantly better than the original ResNet-18. However, its FLOPS is nearly four times higher.

B. *Residual attention network*

In this experiment, we used the Residual Attention Network to build a classification model. We modified the middle Attention Module to have only two stages to meet the requirement of fewer than four layers, and we used a fully connected layer for the output. We aim for our model to achieve an accuracy of at least 90% of that of ResNet34 or better. Therefore, we will compare the accuracy of the two models. Additionally, we will compare the elapsed time of both models to see if our model performs better than ResNet34 in these two metrics. All our experiments use a fixed image size of 224x224, and the models are trained for 30 epochs. During training, we apply image augmentations such as horizontal and vertical flipping, rotation, and distortion.

	Residual Attention Network	ResNet-34
Accuracy	66.44%	69.56%
Precision	69.34%	71.34%
Recall	66.44%	69.56%
F1 Score	66.58%	68.47%
FLOPS	3715245312	3667013632

Table.3 Accuracy and Elapsed Time for Inputs with Different Numbers of Channels

From Table 3, it can be observed that the FLOPS of the Residual Attention Network is higher than that of ResNet-34, but the accuracy is not higher. We believe this is because the number of epochs is not sufficient for either network to fully converge. It can be inferred that ResNet-34 converges faster than the Residual Attention Network, although the difference is not substantial.

IV. CONCLUSION

In conclusion, this assignment has successfully demonstrated the design and implementation of a spatially size-invariant convolutional module, DynamicConv2D, capable of handling varying numbers of input channels. The experimental results indicate that the modified ResNet-18 with DynamicConv2D outperforms the original ResNet-18 in terms of accuracy, precision, recall, and F1 score, albeit with a higher computational cost in terms of FLOPS. The experiments confirm that utilizing three input channels yields the best performance, though the proposed module can flexibly accommodate fewer channels with comparable efficiency.

Furthermore, the Residual Attention Network, despite showing promising accuracy and flexibility in attention mechanisms, did not surpass the performance of ResNet-34 within the 30-epoch training limit. This suggests that longer training durations might be necessary for the Residual

Attention Network to fully converge and potentially outperform traditional models like ResNet-34.

Overall, the designed convolutional module and network configurations provide valuable insights and potential improvements for handling diverse input data in convolutional neural networks, highlighting the importance of adaptive and dynamic approaches in modern deep learning applications. Future work could explore optimization techniques to reduce the computational overhead while maintaining or enhancing the model performance.

References

[1]https://github.com/tengshaofeng/ResidualAttentionNetwork-pytorch/tree/master

[2] https://blog.csdn.net/weixin_51352656/article/details/114231001

[3] https://arxiv.org/abs/1704.06904

[4] Chatgpt