Re-study on Convolutional vision Transformer(CvT) : Inspired by Bilateral Segmentation Network

Yancheng Cai
School of Information Science and Technology
Fudan University
220 Handan Road
19307140030@fudan.edu.cn

Lesi Chen
School of Data Science
Fudan University
220 Handan Road
19307130195@fudan.edu.cn

Zhiyang Liang
School of Data Science
Fudan University
220 Handan Road
19307130184@fudan.edu.cn

Abstract

In our final project, we try to compare the classification accuracy of the pure CNN network and the CNN network with Transformer on CIFAR-10 under the same parameters or the same FLOPs. Our research starts from Convolutional vision Transformer(CvT), but does not stop at CvT. We use a grid search method similar to CvT to find the best learning rate, and also test the effect of position embedding. But we also try to use a different optimizer.More importantly, inspired by the network used in our mid-term project, we have added a bilateral path to CvT in order to get higher classification accuracy. After a lot of experiments, we find that the combination of convolutional layers and Transformer can achieve better results than pure convolutional structure. At the same time, our approach of adding bilateral paths has also greatly improved the classification performance of the network. The best one of our networks is able to achieve 98.75% accuracy on the CIFAR-10 testset. Our code is available at https://github.com/caiyancheng/Computer-Vision-Final-project.

1 Introduction

As is well-known, the Convolutional vision Transformer (CvT[10]) employs all the benefits of CNNs: local receptive fields, shared weights, and spatial subsampling, while keeping all the advantages of Transformers: dynamic attention, global context fusion, and better generalization. The hybrid model will be a network that can obtain better performance than a pure transformer network or a pure CNN network under the premise of a smaller amount of parameters or FLOPs. More importantly, inspired by the network[12] we used for mid-term project, we add a bilateral path to CvT to improve its classification accuracy. We propose one Prior Path to acquire prior knowledge on ImageNet and migrate to the CIFAR-10 dataset and one Context Path to directly get the information on CIFAR-10. Over the two paths, we use FFM (feature fusion module) to fuse the features, and then use MLP as a classifier to get the final result. Finally, we compare many network structures, and conclude that both introducing convolution to transformer and adding bilateral paths have an effect on improving network performance.

2 Related Work

2.1 Introducing Self-attentions to CNNs

Self-attention mechanisms have been widely applied to CNNs in vision tasks. Among these works, the non-local networks[7] are designed for capturing long range dependencies via global attention. The local relation networks[5] adapts its weight aggregation based on the compositional relations (similarity) between pixels/features within a local window, in contrast to convolution layers which employ fixed aggregation weights over spatially neighboring input feature. Such an adaptive weight aggregation introduces geometric priors into the network which are important for the recognition tasks. Recently, BoTNet[6] proposes a simple yet powerful backbone architecture that just replaces the spatial convolutions with global self-attention in the final three bottleneck blocks of a ResNet and achieves a strong performance in image recognition. Instead, our work performs an opposite research direction: introducing convolutions to Transformers.

2.2 Introducing Convolutions to Transformers

In NLP and speech recognition, convolutions have been used to modify the Transformer block, either by replacing multihead attentions with convolution layers[9], or adding additional convolution layers in parallel[11] or sequentially[3], to capture local relationships. Other prior work[8] proposes to propagate attention maps to succeeding layers via a residual connection, which is first transformed by convolutions. Different from these works, we propose to introduce convolutions to two primary parts of the vision Transformer: first, to replace the existing Position-wise Linear Projection for the attention operation with our Convolutional Projection, and second, to use our hierarchical multi-stage structure to enable varied resolution of 2D reshaped token maps, similar to CNNs. Our unique design affords significant performance and efficiency benefits over prior works.

2.3 Well-designed Models

In addition to Transformer, we also introduce more features adopted from other models to our network.

Deep Residual Learning Deeper neural networks are more difficult to train. ResNet[4] presents a residual learning framework to ease the training of networks that are substantially deeper than those used previously. The model explicitly reformulates the layers as learning residual functions with reference to the layer inputs, instead of learning unreferenced functions. It provides comprehensive empirical evidence showing that these residual networks are easier to optimize, and can gain accuracy from considerably increased depth.

More Efficient Residual Network Deep residual networks were shown to be able to scale up to thousands of layers and still have improving performance. However, each fraction of a percent of improved accuracy costs nearly doubling the number of layers, and so training very deep residual networks has a problem of diminishing feature reuse, which makes these networks very slow to train. For the CIFAR10 classification task, the traditional deep resnet may not be the optimal structure, because the deep network will lose a lot of information in the process of continuous downsampling, and this is for the CIFAR10 image with a resolution of only 32x32. Said, it may reduce the performance of the final classification. To tackle these problems, wide residual networks(WRNs[13]) decrease depth and increase width of residual networks, and show that these are far superior over their commonly used thin and very deep counterparts.

Bilateral Paths BiSeNet proposes a Spatial Path with a small stride to preserve the spatial information and generate high-resolution features. Meanwhile, a Context Path with a fast downsampling strategy is employed to obtain sufficient receptive field. On top of the two paths, it introduces a new Feature Fusion Module to combine features efficiently.The ablation study proves that Bilateral paths play a key role in improving the network effect. For our network, we also propose two paths, a Prior Path pretrained on ImageNet and then fine-tune on CIFAR-10, a Context Path directly train on CIFAR-10. Finally, we use feature fusion module(FFM) to fuse the features, and then use MLP as a classifier to output the final result.

3 Convolutional vision Transformer (enhanced by Bilateral Paths)

In order to verify the effectiveness of transformer and bilateral paths in the network, we design a series of networks for comparison. Note that our design(Compare one CNN model and Two Transformer+CNN models) is somewhat different from the requirements of the project(Compare one Transformer+CNN model and two CNN models), but it can also achieve the same goal as we both test networks of the same parameter amount and network of the same FLOPs.

We use the ptflops(available at https://github.com/sovrasov/flops-counter.pytorch) to count the parameter amount and the FLOPs of our models. Sine the attention part of our network is implemented by convolutional layers and linear layers, the FLOPs of them can be rightly calculated by this package. We also calculate the FLOPs of the attention part manually for verification, according to the formulas below :

$2 \times Input \times Output$ (for linear layer with bias)

$(2 \times C_{in} \times K^2) \times H \times W \times C_{out}$ (for convolutional layer with bias)

K denotes the kernel size,H, W denotes the size of the output feature map

And the results are the same. Interestingly, we find that the ptflops may mistake the GFLOPs for GMac.Because it is the GFLOPs that are calculated in the source code, but the unit GMac is mistakenly added in the final output. After solving this important issue, we will continue to introduce the network structure.

3.1 Features of CvT

Convolutional Token Embedding This convolution operation in CvT aims to model local spatial contexts, from low-level edges to higher order semantic primitives, over a multi-stage hierarchy approach, similar to CNNs. The Convolutional Token Embedding layer allows us to adjust the token feature dimension and the number of tokens at each stage by varying parameters of the convolution operation. In this manner, in each stage we progressively decrease the token sequence length, while increasing the token feature dimension. This gives the tokens the ability to represent increasingly complex visual patterns over increasingly larger spatial footprints, similar to feature layers of CNNs.

Convolutional Projection for Attention The goal of the proposed Convolutional Projection layer is to achieve additional modeling of local spatial context, and to provide efficiency benefits by permitting the under- sampling of K and V matrices. Fundamentally, the proposed Transformer block with Convolutional Projection is a generalization of the original Transformer block. While previous works try to add additional convolution modules to the Transformer Block for speech recognition and natural language processing, they result in a more complicated design and additional computational cost. Instead, we propose to replace the original position-wise linear projection for Multi-Head Self-Attention (MHSA) with depth-wise separable convolutions, forming the Convolutional Projection layer.

Figure 1 shows an example stage of CvT with the features mentioned above.

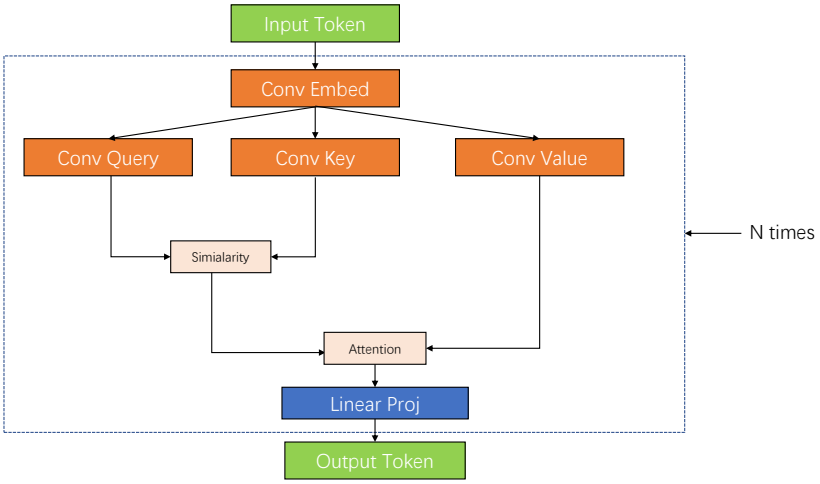

Figure 1: an example stage of CvT

3.2 Bilateral Paths

Inspired by the BiseNet that our team used for the mid-term project, we propose a similar idea for our final-project network. There are one Prior Path to acquire prior knowledge on ImageNet and migrate to the CIFAR-10 dataset and one Context Path to directly get the information on CIFAR-10. Over the two paths, we use FFM (feature fusion module) to fuse the features, and then use MLP as a classifier to get the final result.

Prior Path We use a 3-stage CvT pretrained on ImageNet as one path of our network. We upsample the 32×32 CIFAR-10 picture into 224×224 as the input of this path. We regard this path as a downstream task which means we fine-tune this path on CIFAR-10 in order to unlock the potential of prior knowledge on ImageNet as much as possible.

Context Path The traditional fine-tune method can make good use of the prior knowledge obtained by pre-training, but it lacks direct knowledge of the target dataset. Inspired by BiseNet, we propose a new method to solve such shortcomings.We use common CNN networks (such as Wide ResNet, ResNet) to train directly on the target dataset without pre-training, aiming to obtain more direct information of the target dataset. Figure 2 shows the architecture of one of our networks that uses Wide ResNet as the Context Path.

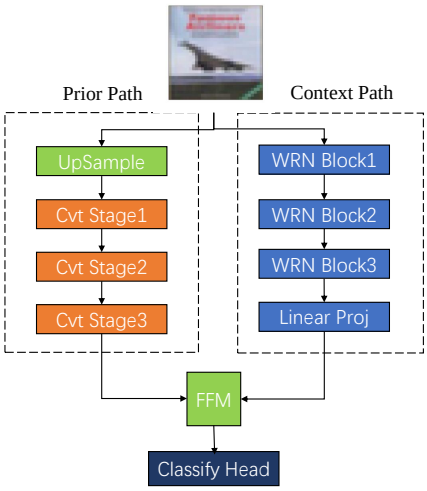

Figure 2: One of our networks

3.3 All the networks

In this project, we aim to compare the performance of convolutional neural networks with the same parameters or the same FLOPs and convolutional neural networks with Transformer. Table 1 shows all the networks we use in this project.

Table 1: All the networks in our project

Architecture	Prior Path	Context Path	Parameters(M)	GFLOPs
CNN	ResNet50	None	23.53	4.11
CNN+Transformer	CvT13	Wide ResNet(8,2)	23.53	4.43
CNN+Transformer	CvT13	Wide ResNet(4,6)	19.91	4.11
CNN+Transformer	CvT13	ResNet18	30.79	4.11

All the Prior Paths are pretrained on ImageNet. Besides the difference of network architecture, we also try different optimization methods or add position embedding.

4 Experiments

4.1 Datasets

ImageNet ImageNet is an image database organized according to the WordNet hierarchy (currently only the nouns), in which each node of the hierarchy is depicted by hundreds and thousands of images. The project has been instrumental in advancing computer vision and deep learning research. The data is available for free to researchers for non-commercial use.The pretrained models that our team use are all on this dataset.

CIFAR-10 The CIFAR-10 dataset consists of 60000 32x32 colour images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images. The dataset is divided into five training batches and one test batch, each with 10000 images. The test batch contains exactly 1000 randomly-selected images from each class. The training batches contain the remaining images in random order, but some training batches may contain more images from one class than another.

Between them, the training batches contain exactly 5000 images from each class. Here are the classes in the dataset, as well as 10 random images from each:

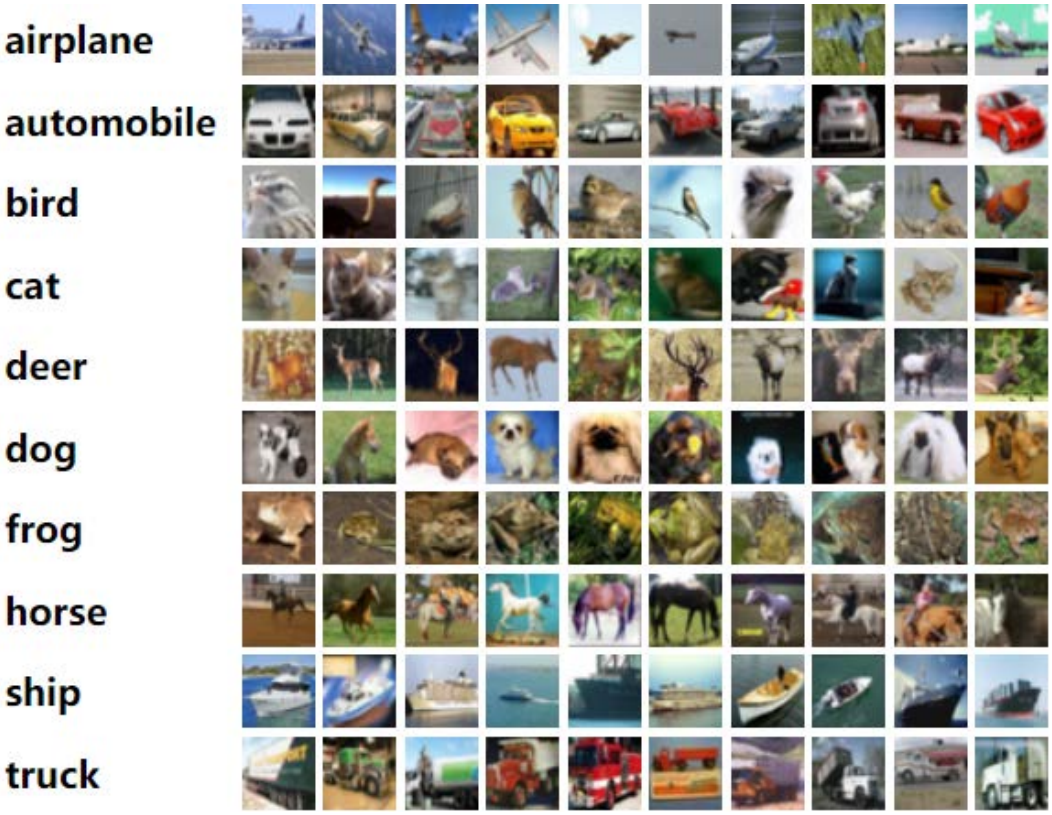

Figure 3: CIFAR-10

The classes are completely mutually exclusive. There is no overlap between automobiles and trucks. "Automobile" includes sedans, SUVs, things of that sort. "Truck" includes only big trucks. Neither includes pickup trucks. Our team use CIFAR-10 for training and evaluation.

4.2 Data augmentation

We apply the random crop(size=(32,32), padding=4), random horizontal flip, label smooth and the famous Cutout[1] method to augment the dataset in training procedures.

4.3 Optimizer

We either use Stochastic Gradient Descent(SGD) or Sharpness-Aware Minimization(SAM[2]) to optimize our models.SGD is already well known, so we will not go into details here in our report, we will focus on SAM.

SAM In today's heavily overparameterized models, the value of the training loss provides few guarantees on model generalization ability. Indeed, optimizing only the training loss value, as is commonly done, can easily lead to suboptimal model quality. Motivated by prior work connecting the geometry of the loss landscape and generalization, a novel, effective procedure for instead simultaneously minimizing loss value and loss sharpness is introduced. In particular, our procedure, Sharpness-Aware Minimization (SAM), seeks parameters that lie in neighborhoods having uniformly low loss; this formulation results in a minmax optimization problem on which gradient descent can be performed efficiently.

The iteration process is as follows:

$$\epsilon_t = \rho \frac{\nabla L(w_t)}{\|\nabla L(w_t)\|_2}$$

$$w_{t+1} = w_t - \alpha(\nabla L(w_t + \epsilon_t) + \lambda w)$$

where ρ is a hyperparameter called the neighborhood size. We use SAM to reduce the generalization error to alleviate the problem of network overfitting.

And in some of our experiments, we apply learning rate scheduler to optimize our model.The learning rate change is shown in Figure 2.

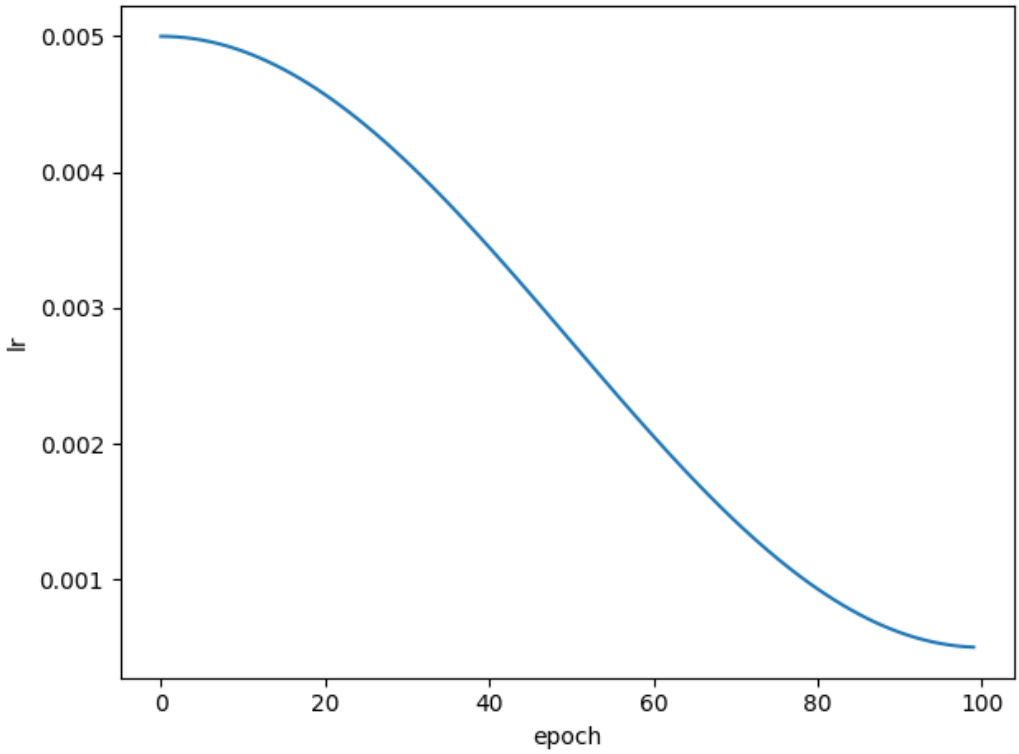

Figure 4: Learning rate Scheduler

The scheduler is adopted from YOLOv5(available at https://github.com/ultralytics/yolov5), actually part of the Cosine Annealing Scheduler.We think there are the following benefits.

1. Since we have already loaded the pre-trained model on ImageNet, the initialization point should not be far from the global minimum we want, and it is unlikely that there will be a local minimum that escapes in the middle, so there is no need for multi-peak Cosine Annealing.
2. We hope to maintain a large learning rate during the initial training period so that the model can quickly approach the global minimum.
3. Then, We hope to keep a small learning rate in the last period of training so that the model can fit more fully.
4. Continuous changes in the learning rate will also prevent instability in training (for example, a drop in the stepped learning rate may cause the model to fail to run stably in certain stages of training)
5. In the middle of the training, we allow the learning rate to drop rapidly, because the previous training has allowed it to break away from most local minimum.

4.4 Other Details

All the CvT parts, ResNet parts and WRN parts are pretrained on ImageNet. Since we use label smooth, the loss function is not the original CrossEntropy Loss, but is not of great difference.We try learning rate in {1e-2,5e-3,3e-3,1e-3} to train our network with batch size 32 and epoch 100. We set the momentum = 0.9 weight decay = 0 for SGD and SAM and the $\rho = 0.5$ for SAM. Figure 3 and Figure 4 show that our training methods work quite well.

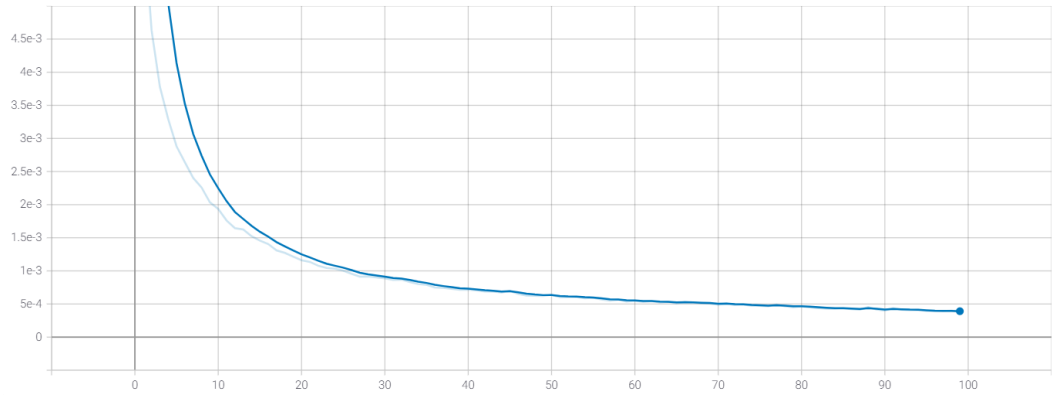

Figure 5: Training Curve

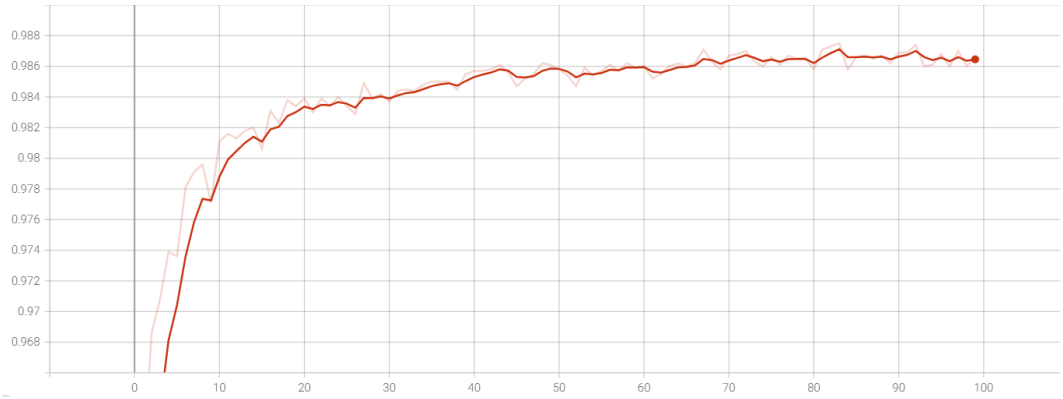

Figure 6: Test Accuracy Curve

4.5 Comparison Results

Same FLOPs We try SGD without any scheduler of different learning rate to train the networks with the same FLOPs. The training parameters are given in the format: (learning rate,epoch). And the Table shows the accuracy on the test set.

Table 2: Results of Networks with Same FLOPs

Arch | Param	1e-2,20	1e-2,100	3e-3,20	3e-3,100	1e-3,20	1e-3,100
CNN	96.14	96.96	96.93	97.41	97.11	97.80
CNN+Trans	97.04	97.69	97.88	98.37	97.83	98.28

Same Parameters We try SGD without any scheduler of different learning rate to train the networks with the same parameters. The training parameters are given in the format: (learning rate,epoch). And the Table shows the accuracy on the test set.

Table 3: Results of Networks with Same Parameters

Arch | Param	1e-2,20	1e-2,100	3e-3,20	3e-3,100	1e-3,20	1e-3,100
CNN	96.14	96.96	96.93	97.41	97.11	97.80
CNN+Trans	96.87	97.44	97.82	98.46	97.93	98.33

Synthesizing the contents of Table 2 and Table 3, we can see that whether it is a network with the same FLOPs or the same parameters, after adding the Transformer, the results are better than the original structure, which shows the effectiveness of the Transformer.

SAM We no longer use SGD, but use SAM to optimize our training process. The training parameters are given in the format: (learning rate,epoch). And the Table shows the accuracy on the test set.

Table 4: Results of CNN+Transformer with 4.11 GFLOPs

Opt | Param	3e-3,20	3e-3,100	1e-3,20	1e-3,100
SGD	97.88	98.37	97.83	98.28
SAM	97.92	98.60	98.21	**98.75**

Table 5: Results of CNN+Transformer with 23.53M Parameters

Opt | Param	3e-3,20	3e-3,100	1e-3,20	1e-3,100
SGD	97.82	98.46	97.93	98.33
SAM	98.13	98.71	98.01	98.63

Combining the results of Table 4 and Table 5, we can see that the use of SAM always improves the final result of the network to a certain extent, which is consistent with the SAM theory proving that it can reduce the degree of network overfitting.

Bilateral Paths To illustrate that our bilateral paths works, we compare it with original CvT that does not use bilateral paths. Still, we try SGD without any scheduler of different learning rate to train the networks with the same parameters. The training parameters are given in the format: (learning rate,epoch). And the Table shows the accuracy on the test set.

Table 6: Results of CNN+Transformer with 23.53M Parameters

Arch | Param	3e-3,20	3e-3,100	1e-3,20	1e-3,100
CvT	97.77	98.39	97.63	98.27
CvT+CNN	97.82	98.46	97.93	98.33

As is shown in Table 6, our bilateral paths do provide more information and make the results better.

Other Experiments We try to use YOLOv5 scheduler or add position embedding to Transformer. Unfortunately, these do not help improve the final classification accuracy(98.60% and 98.53% respectively). We speculate that the reason for the former is that the scheduler may be specially designed for YOLOv5 and is not suitable for our network structure and CIFAR-10 dataset. The latter is because the introduction of Convolutional Projections for every Transformer block, combined with the Convolutional Token Embedding, has already givenus the ability to model local spatial relationships through the network.

5 Conclusion

In the final project, we have presented a detailed study of introducing convolutions into the Vision Transformer architecture to merge the benefits of Transformers with the benefits of CNNs for image recognition tasks. Extensive experiments demonstrate that due to the built-in local context structure introduced by convolutions, CvT no longer requires a position embedding. What's more, our experiments re-prove the advantage of SAM in preventing overfitting of small datasets like CIFAR-10. Last but not the least, the introduced bilateral paths can make our CvT architecture achieve superior performance while maintaining computational efficiency, which may be an idea of designing and optimizing neural networks with strong generalization.

References

[1] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.

[2] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. *arXiv preprint arXiv:2010.01412*, 2020.

[3] Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, et al. Conformer: Convolution-augmented transformer for speech recognition. *arXiv preprint arXiv:2005.08100*, 2020.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[5] Han Hu, Zheng Zhang, Zhenda Xie, and Stephen Lin. Local relation networks for image recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3464–3473, 2019.

[6] Aravind Srinivas, Tsung-Yi Lin, Niki Parmar, Jonathon Shlens, Pieter Abbeel, and Ashish Vaswani. Bottleneck transformers for visual recognition. *arXiv preprint arXiv:2101.11605*, 2021.

[7] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018.

[8] Yujing Wang, Yaming Yang, Jiangang Bai, Mingliang Zhang, Jing Bai, Jing Yu, Ce Zhang, Gao Huang, and Yunhai Tong. Evolving attention with residual convolutions. *arXiv preprint arXiv:2102.12895*, 2021.

[9] Felix Wu, Angela Fan, Alexei Baevski, Yann N Dauphin, and Michael Auli. Pay less attention with lightweight and dynamic convolutions. *arXiv preprint arXiv:1901.10430*, 2019.

[10] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. *arXiv preprint arXiv:2103.15808*, 2021.

[11] Zhanghao Wu, Zhijian Liu, Ji Lin, Yujun Lin, and Song Han. Lite transformer with long-short range attention. *arXiv preprint arXiv:2004.11886*, 2020.

[12] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Bisenet: Bilateral segmentation network for real-time semantic segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 325–341, 2018.

[13] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.