ViT lightweight training at IoT edge based on transfer learning

Zhixin Li[a], Yang Long*[b], Jiahao Miao[b]
[a]School of Computer Technology and Engineering, Changchun Institute of Technology, Changchun 130012, Jilin, China; [b]School of Information and Control Engineering, Jilin Institute of Chemical Technology, Jilin 132022, Jilin, China

ABSTRACT

Recently, the vision transformer (ViT) model of deep learning has achieved surprising performance in the field of computer vision and has been widely used in IoT edge devices. However, the training of ViT models requires a large amount of data and computing resources, which is a challenge for resource-constrained edge IoT devices. To solve the above problems, this paper proposed a lightweight ViT method based on transfer learning. The primary concept of this method is to train large-scale ViT models in the cloud (CloudViT) and deploying small-scale ViT models at the edge (EdgeViT). Firstly, through the method of transfer learning, some underlying parameters of CloudViT were utilized to construct EdgeViT. The purpose is to enable EdgeViT to learn from CloudViT, acquiring knowledge and improving its performance. Secondly, adding a randomly initialized LayerNorm layer before the MLPHead during the training process of EdgeViT, it can improve further model performance. Finally, Experiment results demonstrated that EdgeViT could achieve 91.3% of CloudViT's performance with only half the parameters and floating-point operations (FLOPs). Moreover, finetuning EdgeViT with a 60% reduction in training time still allows it to achieve 81.3% of CloudViT's performance. Relevant conclusions can provide technical support for the proposed method.

Keywords: ViT, deep learning, transfer learning, IoT, edge computing, cloud-assisted

1. INTRODUCTION

In recent years, Vision Transformer (ViT) and its variants[1,2] have demonstrated remarkable performance surpassing convolutional neural networks (CNNs) in various computer vision tasks. The application of ViT models on the Internet of Things (IoT) to provide high-quality services to end-users has garnered significant attention from researchers. However, ViT models entail a large number of parameters and Floating Point Operations per Second (FLOPs), making the training of high-performance ViT models demanding in terms of data and computational resources[3]. Achieving training and deployment of ViT models on resource-constrained edge devices is thus exceedingly challenging. Consequently, the integration of ViT models into edge devices remains a pressing issue to address.

Various neural network compression techniques and model optimization methods, combined with cloud computing for assisted model training, have emerged as primary approaches to address this issue[4]. Researchers have proposed various cloud-edge collaboration frameworks and distributed deep neural network architectures to enhance the performance and efficiency of deep learning models on edge devices. Li et al.[5] introduced the FitCNN framework, which utilizes cloud assistance to gradually collect incremental training data on mobile devices and retrain the model to improve its performance. Ding et al.[6] proposed a cloud-edge collaboration framework for cognitive services, where the cloud is utilized for training deep neural networks, and the edge is used for training shallow neural networks. They leverage the lower layers of the deep model to assist in training the shallow model, thereby enhancing the performance of edge-side neural networks. Hsu et al.[7] proposed a cloud-edge intelligent IoT architecture based on transfer learning techniques to accelerate the deployment of neural network models. Furthermore, significant contributions have been made in the field of model lightweighting and optimization. Zheng et al.[8] introduced a versatile ViT acceleration and pruning framework, SAVIT, which reduces computational costs and accelerates pruning without compromising model performance. Li et al.[9] proposed a fully differentiable quantization method for ViT, Q-ViT, pushing the quantization limit of ViT to 3 bits. Chen et al.[10] presented an early knowledge distillation framework, DearKD, achieving outstanding performance on ImageNet. Chen et al.[11] combined MobileNet with ViT, proposing a lightweight model called Mobile-Former.

*longyang@jlict.edu.cn

International Conference on Optics, Electronics, and Communication Engineering (OECE 2024),
edited by Yang Yue, Proc. of SPIE Vol. 13395, 133951J · © 2024 SPIE · 0277-786X · Published
under a Creative Commons Attribution CC-BY 3.0 License · doi: 10.1117/12.3048338

Despite the effectiveness of existing methods for lightweighting and optimizing ViT models, when it comes to large-scale deployment in IoT edge environments, the high computational costs and complexity hinder seamless integration with cloud computing. To address this issue, this paper proposes a transfer learning-based approach for ViT model lightweighting. This method utilizes the low-level features of ViT models for transfer learning training, resulting in smaller-scale ViT models that are more suitable for deployment on edge devices. In comparison to the above methods, this approach is more straightforward, does not alter the ViT network structure, and thus can be more easily combined with cloud-assisted techniques. Moreover, it provides a promising starting point and optimization foundation for future research on ViT model lightweighting.

2. METHOD

2.1 Cloud-assisted training framework

The framework comprises three key components: the cloud, the edge, and IoT devices (Figure 1). The cloud, which includes servers from platforms such as Google Cloud and Alibaba Cloud, is known for its powerful computing and storage capabilities. The edge comprises devices like edge gateways and servers, providing fast response services to users. IoT devices comprises various sensors and personal devices like network cameras and personal computers, typically transmitting data to either the cloud or the edge for task processing. Adapting the scale of EdgeViT to match the computational and storage capabilities of edge devices.

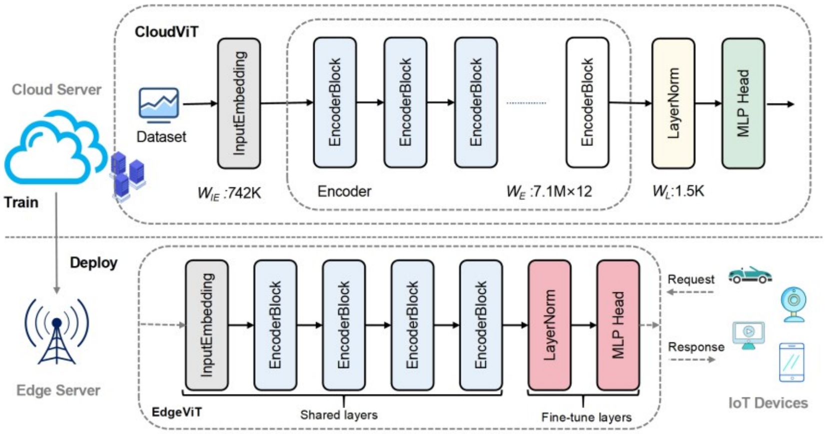

Figure 1. Cloud-assisted framework and its workflow.

2.2 Details

ViTBase contains around 86 million parameters[1], with the majority located in its Encoder (approximately 85.2 million). The Encoder comprises 12 identical EncoderBlocks. According to Jason et al.'s research[12], neural network models tend to learn more general features in layers closer to the input. This characteristic allows us to utilize the low-level features of the ViT model for transfer learning training. By removing some EncoderBlocks from the end of the ViT model, we can shrink its size, reducing both the number of parameters and FLOPs. Leveraging this capability, CloudViT is trained in the cloud, and through transfer learning techniques, a portion of its parameters is shared with EdgeViT to enhance the performance and training efficiency of EdgeViT. As data at the cloud and edge may differ, CloudViT may not directly adapt to specific edge tasks. This aligns with the core principle of transfer learning, which aims to utilize knowledge and experience gained from one task to improve performance on new tasks.

CloudViT developed in the cloud serves as the starting point for EdgeViT at the edge. Using the cloud data domain as the source domain D_{Cloud} and the edge task domain as the target domain D_{Edge}.

$$D_{Cloud} = \{x_i, y_i\}_i^{M_s}; D_{Edge} = \{x_i, y_i\}_i^{M_t} \tag{1}$$

In traditional transfer learning, the standard practice involves transferring all parameters of the source domain model, excluding the classification head, to the target domain. In the target domain, the structure and parameters of the classification head are task-specific and require retraining and adjustment to meet the task requirements of the target domain.

In the ViT model for the target task, the parameters of the Input Embedding layer are denoted as W_{IE}, the parameters of the Encoder are denoted as W_E, the parameters of the LayerNorm are denoted as W_L, and the classification head is denoted as W_H. Additionally, this study found that the LayerNorm layer, despite adaptation on the original task's large-scale dataset, may have adverse effects when transferred to the target task due to mismatched data distributions. Therefore, this paper considers LayerNorm as part of the classification head, initializing it randomly and retraining it on the edge task during the training process of EdgeViT.

Consequently, on the edge target task, given the target training data $\{x_i, y_i\}_{i=1}^{N}$, fine-tuning is performed on the classification head and LayerNorm, optimizing the following loss function:

$$f_{\varphi}(W_{L+H}) = \frac{1}{N}\sum_{i=1}^{N}\gamma(y_i, \xi(f(x_i, W_{L+H}))) \tag{2}$$

The algorithm proposed in this paper for ViT cloud-assisted edge fine-tuning training based on transfer learning follows the steps below:

Algorithm 1 Workflow

```
Input: CloudViT with paramters: $W_{IE}, W_E$; $D_{Edge}$ Trainning dataset $\{x_i, y_i\}_{i=1}^{M_S}$
Output: EdgeViT
1 Cloud send CloudViT's Parameters $W_{IE}$ and part of $W_E(W_{poE})$ to Edge
2 Edge server use $W_{IE}$, $W_E$, and random Initialization Layernorm, MLPHead form an EdgeViT
3 $\{x_i, y_i\}_{i=1}^{M_S}$ optimal loss function $f_{\varphi}(W_{L+H})$
4 Return EdgeViT with $W_{IE} \cup W_{poE} \cup W_{L+H}$
```

3. EXPERIMENTS

3.1 Experimental setup

The pre-trained model ViT-Base-Patch16-224 (ViTBase)[1] is utilized as CloudViT. This model is pre-trained on the imagenet-21 k dataset. The experimental environment includes Python 3.7, TensorFlow 2.5, and CUDA 11.6. The experimental hardware consists of an NVIDIA GTX1650 GPU with 896 CUDA Cores, a base clock speed of 1395 MHz, and 4 GB GDDR5 128-bit VRAM. The model employs the SGD optimizer with momentum, an initial learning rate of 1×10^{-3}, a batch size of 4, and 10 epochs. The MLPHead is configured as one Dense layer.

Table 1. Dataset.

	Training set	Test set	Laber	Size
Cat & dog	5913	1477	37	224×224×3
Nature scene	13630	3404	6	224×224×3
Animal	4320	1080	90	224×224×3

The experiments utilized three datasets (Table 1): the Cat & dog dataset[13], comprising approximately 7.3 K images covering 37 different breeds of cats and dogs; the Natural Scene dataset initially released by Intel for an image classification challenge, contains around 17 K images distributed across six categories: buildings, forests, glaciers, mountains, seas and streets[14]; and the Animal dataset, which includes 5.4 K animal images encompassing 90 different species[15]. ViTBase achieved Top-1 accuracies of 93.9%, 93.6%, and 91.6% on these three datasets, respectively.

3.2 LayerNorm

To explore the impact of different transfer training methods on EdgeViT's LayerNorm performance, we devised the following experiments. We categorized the training approaches for transfer models into four distinct types (Table 2).

The ViT-f6 refers to loading the weights of the Input Embedding section and the first 6 EncoderBlocks from the pre-trained ViTBase model into this model and freezing them, which is then used as EdgeViT. After ten epochs, the experimental results are shown in Table 2.

Table 2. The top-1 accuracy of models on each dataset (%).

Model	Training methods	Cat & dog	Nature scene	Animal
ViT-f6-1	Transfer LayerNorm with fine-tuning	49.9	89.2	39.4
ViT-f6-2	Transfer LayerNorm, non-fine-tuning	54.7	90.1	39.8
ViT-f6-3	No Transfer of LayerNorm, no fine-tuning	71.6	92.2	58.5
ViT-f6-4	No Transfer of LayerNorm with fine-tuning	73.6	92.6	61.0

Based on the data in Table 2, training methods three and four outperform methods one and two by approximately 20%, 19%, and 2.6% in average accuracy, respectively. This indicates that not transferring the LayerNorm layer and fine-tuning it can achieve the optimal model performance. Additionally, the LayerNorm layer only consists of 1.5 k parameters, so training this part hardly incurs any additional computational cost.

3.3 Performance of EdgeViT

This experiment aims to compare the top-1 accuracy of EdgeViT with randomly initialized ViT on the three datasets mentioned above. The performance of the models trained using the three methods from Table 3 is compared. After training for 10 epochs, the experimental results are shown in Table 4.

Table 3. Explanation of different training methods.

Model	Initialization and training methods
ViT-n*	Randomly initialized ViT model with * EncoderBlocks trained from scratch.
ViT-f*	EdgeViT, initialized with CloudViT assistance, undergoes fine-tuning training
ViT-f*-T	EdgeViT initialized with CloudViT assistance, trained from scratch.

Table 4. Model accuracy on the mentioned datasets. (%)

	Cat & dog	Natural scenes	Animal
ViT-n*$^{(4/6/8)}$	27.7/24.1/24.7	78.5/80.3/78.6	19.8/19.2/19.5
ViT-f*$^{(4/6/8)}$	62.1/73.6/79.1	90.5/92.4/93.4	45.0/61.0/68.8
ViT-f*$^{(4/6/8)}$-T	81.4/88.0/91.9	92.4/93.9/94.5	59.4/73.1/81.9

Note: * denotes 4/6/8, representing ViT models with 4, 6, or 8 Encoder blocks respectively. Accuracy of each model is listed from left to right.

Table 4 displays the Top-1 accuracy comparison of various compact ViT models after fine-tuning and retraining for 10 epochs on the three datasets. For ViT-f4, fine-tuning achieves an average accuracy of 70.8% of ViTBase; if fully trained, it reaches 83.5% of ViTBase's accuracy. ViT-f6 attains 81.3% of ViTBase in fine-tuning training; if fully trained, it reaches 91.3% of ViTBase's accuracy. When fine-tuning ViT-f8, it achieves 86.4% of ViTBase's accuracy; if fully trained, it reaches 96.1% of ViTBase's accuracy. It is evident that, whether through fine-tuning or retraining, EdgeViT utilizing shared parameters from CloudViT demonstrates performance closer to CloudViT and significantly outperforms traditional ViT models. Additionally, during fine-tuning training, compared to training models of the same scale from scratch, each epoch can save approximately 60% of the training time (Figure 2).

Table 5 shows that ViT-f4 has only one-third of the parameters and FLOPs of ViTBase, ViT-f6 has half, and ViT-f8 has two-thirds. The EdgeViT model has a significantly smaller scale. Combining the experimental data above, it can be clearly observed that EdgeViT achieves higher performance in a shorter training time.

Table 5. Model scales.

	ViTBase	**ViT-f4**	**ViT-f6**	**ViT-f8**
FLOPs	16.86 G	5.7 G	8.49 G	11.28 G
#Params	86 M	29 M	43 M	57 M

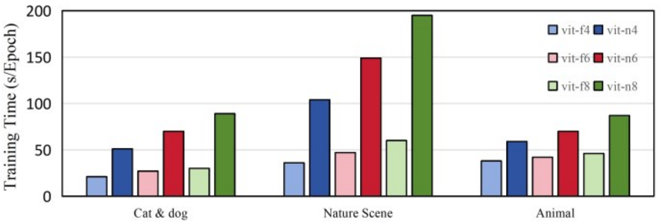

Figure 2. Training duration of the model on the dataset.

Taking ViT-f6 as an example, this model achieves a comprehensive performance of 90% compared to ViTBase while having only half the parameter count and FLOPs. Additionally, it reaches this performance with fewer training epochs (Figure 3). When fine-tuning is applied to EdgeViT, its comprehensive performance reaches 80% of ViTBase. Furthermore, fine-tuning saves approximately 60% of the training time per epoch compared to training from scratch.

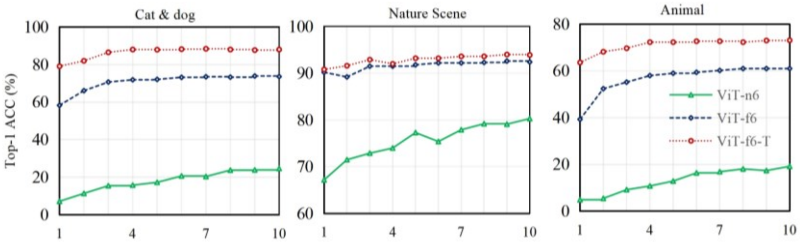

Figure 3. Accuracy of ViT-f6 on three datasets.
Note: With the change of epochs.

The experiments revealed that by removing some of the bottom EncoderBlocks of the ViT model, there is a significant reduction in the number of parameters and FLOPs. This reduction far exceeds the decrease in model performance. Therefore, we can flexibly adjust the scale of EdgeViT according to specific task requirements, thereby better adapting to the computational and storage capabilities at the edge.

4. CONCLUSION

The paper introduces a ViT model lightweighting method that combines transfer learning techniques. This method utilizes the bottom-layer parameters of CloudViT to initialize EdgeViT. Additionally, it involves randomly initializing the LayerNorm layer before MLPHead and fine-tuning it together with the classification head. Experimental results demonstrate that the EdgeViT constructed using this method exhibits higher training efficiency, facilitating deployment in cloud-assisted edge frameworks. In the future, we will integrate additional optimization techniques to enhance EdgeViT and evaluate its performance on tasks such as object detection and image segmentation.

ACKNOWLEDGEMENTS

This work was supported by the Natural Science Foundation of Jilin Province of China (Grant No. 20220101136JC).

REFERENCES

[1] Dosovitskiy, A., Beyer, L., Kolesnikov, A., et al., “An image is worth 16x16 words: Transformers for image recognition at scale,”arXiv abs/2010.11929, (2020).

[2] Liu, Z., Lin, Y., Cao, Y., et al., “Swin transformer: Hierarchical vision transformer using shifted windows,” Proc. IEEE/CVF International Conference on Computer Vision, 10012-10022 (2021).

[3] Chen, H., Wang, Y., Guo, T., et al., “Pre-trained image processing transformer,” Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition, 12299-12310 (2021).

[4] Chang, Z., Liu, S., Xiong, X., et al., “A survey of recent advances in edge-computing-powered artificial intelligence of things,” IEEE Internet of Things Journal 8(18), 13849-13875 (2021).

[5] Li, S., Liu, D., Xiang, C., et al., “Fitcnn: A cloud-assisted lightweight convolutional neural network framework for mobile devices,” Proc. 2017 IEEE 23rd International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA), 1-6 (2017).

[6] Ding, C., Zhou, A., Liu, Y., et al., “A cloud-edge collaboration framework for cognitive service,” IEEE Transactions on Cloud Computing 10(3), 1489-1499 (2022).

[7] Hsu, T.-H., Wang, Z.-H. and See, A. R., “A Cloud-Edge-Smart IoT Architecture for speeding up the deployment of neural network models with transfer learning techniques,” Electronics 11(14), 2255 (2022).

[8] Zheng, C., Zhang, K., Yang, Z., et al., “Savit: Structure-aware vision transformer pruning via collaborative optimization,” Advances in Neural Information Processing Systems 35, 9010-9023 (2022).

[9] Li, Z., Yang, T., Wang, P., et al., “Q-ViT: Fully differentiable quantization for vision transformer,” arXiv abs/2201.07703, (2022).

[10]Chen, X., Cao, Q., Zhong, Y., et al., “Dearkd: Data-efficient early knowledge distillation for vision transformers,” Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition, 12052-12062 (2022).

[11]Chen, Y., Dai, X., Chen, D., et al., “Mobile-former: Bridging mobilenet and transformer,” Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition, 5270-5279 (2022).

[12]Yosinski, J., Clune, J., Bengio, Y., et al., “How transferable are features in deep neural networks?,” Proc. 27th International Conference on Neural Information Processing Systems 2, 3320-3328 (2014).

[13]Parkhi, O. M., Vedaldi, A., Zisserman, A., et al., “Cats and dogs,” Proc. 2012 IEEE Conference on Computer Vision and Pattern Recognition, 3498-3505 (2012).

[14]Bansal, P., “Intel image classification,” Kaggle, 2021, <https://www.kaggle.com/datasets/puneet6060/intel-image-classification> (20 May 2024).

[15]Banerjee, S., “Animal image dataset,” Kaggle, 2022, <https://www.kaggle.com/datasets/iamsouravbanerjee/animal-image-dataset-90-different-animals> (20 May 2024).