Efficient Image Classification via Knowledge Distillation: A Case Study Using ResNet Models in PyTorch

Changning Kang
Auckland, New Zealand

***Abstract*— This project applies the classic knowledge distillation method to transfer knowledge from the pre-trained ResNet34 teacher model to the lightweight ResNet18 student model, achieving model compression and acceleration in image classification tasks. During training, soft targets and real labels are combined for joint supervision. Experiments demonstrate that the distilled student model achieves significant reductions in parameter count and model size while maintaining comparable accuracy, validating the practical value of knowledge distillation in efficient deployment scenarios.**

***Keywords*— Knowledge Distillation, ResNet, Deep Learning, Model Compression, Image Classification**

I. INTRODUCTION

Image classification is an important task in computer vision. Although convolutional neural network (CNN) models perform well in this task, high performance often arises at the a significant cost of computational resource consumption. For resource-limited devices or scenarios, this sort of model is often difficult to deploy. Therefore, researching how to compress models while maintaining high accuracy, making them easier to deploy and promote, is crucial.

Knowledge distillation is a technique aimed at transferring knowledge from large models to smaller ones. Basically, it enables smaller models to learn validated knowledge structures from larger models during training, consequently enhancing their own performance.

This project implements the specific process using PyTorch, using ResNet34 as the teacher model to transfer its knowledge to the ResNet18 student model, and compares their performance in image classification tasks.

II. LITERATURE REVIEW

Knowledge distillation (KD) was proposed by Hinton et al. in 2015[1] to extract knowledge from large teacher models and transfer it to small student models. The core idea is to train students to mimic the probability distribution of the teacher's softmax output at high temperature T, namely, soft targets. Soft targets contain more inter-class correlation information than hard labels, thus are considered to embody the implicit knowledge of the teacher model. During training, a combined training approach is used, which optimizes the student model by adding the cross-entropy loss of the soft target with the loss of the true label. The temperature T controls the smoothness of the soft target. A higher T makes the teacher output distribution smoother and provides more information, though an excessive temperature may result in noise [1]. Empirical results show that the performance of small models trained through distillation can approach the performance of large teacher models [5].

In the field of image classification, the classic KD algorithm and its many variants have been widely used and achieved good achievements. For example, FitNets [2], proposed by Romero et al., apply the intermediate layer features of the teacher as “prompts” to guide training, enabling a student network with only one-tenth of the teacher's parameters to exceed the teacher's accuracy on CIFAR-10 [2]. Zagoruyko et al. proposed Attention Transfer (AT) [3], which allows students to mimic the attention mapping from teachers, thereby improving the accuracy of student models across multiple datasets [3]. Relational KD (RKD) by Park et al. [4] preserves inter-sample distance structures [4]. In addition, Tian et al. proposed Contrastive Representation Distillation (CRD) [5] based on contrast-based learning, which exceeded traditional KD methods on the CIFAR-100 benchmark [5]. Considering the excessive gap between teacher and student capacity, Mirzadeh et al. proposed Teacher Assistant Distillation (TAKD) [6], introducing a medium scale “teacher assistant” network to transmit knowledge in two steps. Experiments have shown that the introduction of teacher assistants can significantly improve the accuracy of student models [6].

Deep residual networks (ResNet) have successfully trained deeper networks by introducing residual connections, achieving high accuracy in ImageNet image classification and winning the ILSVRC2015 championship [7]. ResNet maintains good trainability while providing high accuracy and becoming one of the most used models in the field of vision. In knowledge distillation tasks, to ensure that the teacher-student model is similar and appropriately sized, many studies have used a combination of ResNet-34 as the teacher and ResNet-18 as the student for experiments [8]. This model combination has been proven to enable effective knowledge transfer and improve the performance of the student model [8].

In summary, this project adopted the classic knowledge distillation (KD) method introduced by Hinton et al., instead of the various improved versions that appeared in subsequent research (such as FitNets, AT, TAKD, etc.). This choice is motivated by the simplicity, efficiency, and controllability of the classical approach. It only requires using the final soft target of the teacher model to control student training, without the need for additional intermediate feature alignment or special training processes, therefore reducing implementation and parameter tuning complexity. In contrast, many existing KD variants rely on large teacher networks, intermediate outputs (such as attention maps or feature layers), or require complex training plans, all of these which increase memory and computational burden; or they require multi-stage training processes or the introduction of teacher assistant networks, increasing training complexity. Regarding the above situations, this project incorporates the core ideas of knowledge distillation in the actual process and focuses on the simplicity and deployability of the model: specifically, we first train the ResNet34 teacher model using preprocessed data, and then use ResNet18 as the student model for distillation training. This design demonstrates that this project successfully applied knowledge distillation technology in practice, achieving model compression and performance improvements at a low cost, enabling smaller student models to exhibit good performance and practical deployment value.

III. Methodology

This section provides a detailed description of the implementation process for this project, including data preprocessing, teacher model training, knowledge distillation training process, and hyperparameter settings.

1. Data Preprocessing

The image dataset used in this project contains 10 object categories. First, all images are uniformly resized to 224×224 (compatible with commonly-used CNN structures, balancing feature computation and efficiency) and converted to float32 type. Next, pixel values are scaled to the [0,1] range and standardized using the mean and standard deviation of the training set's own channels, ensuring better adaptation to the actual data distribution. The images are then converted from NHWC format to NCHW format and saved as PyTorch .pt files to improve data loading efficiency during subsequent training. For labels, dictionary mapping is used to enable bidirectional conversion between class names and indices, facilitating both model training and result decoding.

2. Teacher Model Training (ResNet34)

The teacher model employed is a pretrained ResNet34, with its output layer modified by adding a Dropout layer (dropout rate = 0.5) to mitigate overfitting and enhance generalization. The final layer's output dimension is set to 10, corresponding to the number of classes in the dataset. During training, all parameters except for the last fully connected layer are frozen, and only the output layer is fine-tuned. The Adam optimizer is selected due to its adaptive learning rate, which suits the fine-tuning scenario. The initial learning rate is set to 0.0001 to ensure stable updates based on the pre-trained model. To limit the parameter size and reduce the risk of overfitting, a weight decay parameter of 1e-4 is used, which is equivalent to L2 regularization. The batch size is set to 64, balancing training efficiency and GPU memory constraints. The loss function is cross-entropy with label smoothing (coefficient 0.1), which helps reduce overfitting to specific labels and improves generalization. The entire model is trained for 10 epochs, and experiments have verified that this number of epochs is sufficient for convergence. After each epoch, the loss and accuracy are recorded, and the model parameters that perform best on the validation set are saved as the teacher model for the subsequent distillation stage.

3. Knowledge Distillation Training

The student model employed an untrained ResNet18, with an output layer structure consistent with the teacher model, including a Dropout layer (ratio of 0.5) and a fully connected layer with an output dimension of 10. The training procedure adopts classical knowledge distillation, where the student learns from both the cross-entropy loss using ground-truth labels and the KL divergence loss from the teacher's soft targets. The total loss is formulated as:

$$Loss\_total = \alpha \times T^2 \times KL(softmax(z_t / T) \parallel softmax(z_s / T)) + (1 - \alpha) \times CE(y, softmax(z_s))$$

where z_t and z_s denote the logits of the teacher and student models, y is the ground-truth label, T is the temperature, and α is the balanced coefficient. The teacher model's output is smoothed using a softmax function with a temperature of 5, which helps the student model learn the relationships and knowledge between categories. The two parts of the loss are weighted and summed using a weighting coefficient α = 0.5 to balance the teacher's knowledge transfer with the supervision signal of the true labels. The optimizer is Adam, with an initial learning rate of 0.001, suitable for rapid convergence when training the student model from scratch. Weight decay is set to 5e-5 to implement L2 regularization and reduce the risk of overfitting. A StepLR scheduler is employed to reduce the learning rate by a factor of 0.1 every 10 epochs, supporting later-stage convergence. Throughout training, the model's performance on both the training and validation sets is monitored, and the version achieving the best validation accuracy is saved for final deployment.

IV. Results

The experiment compared the performance of ResNet34 and distilled ResNet18 on the validation set. As shown below:

Models	Validation Accuracy	Parameters (Million)	Model Size (MB)
ResNet34 (Teacher)	97.42%	2.13	81.35
ResNet18 (Student)	88.42%	1.12	42.73

Table 1. Performance Comparison Between Teacher and Student Models

As illustrated in the table (Table 1), the distilled ResNet18 model achieves a significant improvement in deployment efficiency, with a reduction of approximately 47% in the number of parameters and a reduction of nearly half in model size, while only experiencing a slight decrease in accuracy.

In addition, we plotted the loss and accuracy curves during training and validation (figure 1 & figure 2) to show the convergence trend and performance stability of the model at different iterations.

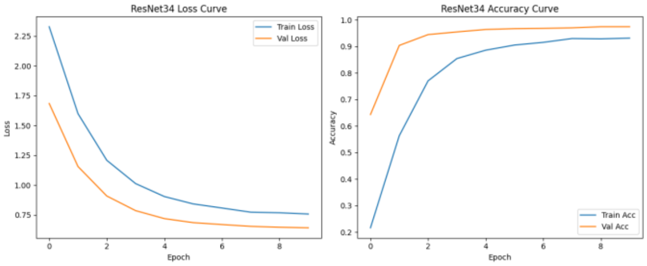

Figure 1. Training and validation loss/accuracy curves of ResNet34 (Teacher).

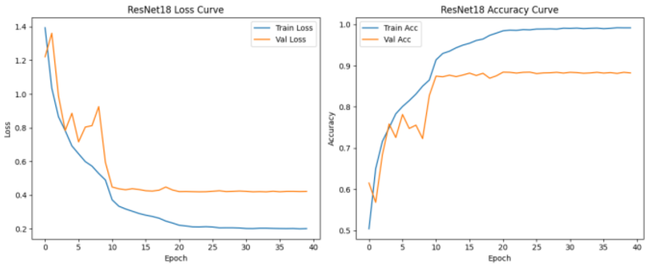

Figure 2. Training and validation loss/accuracy curves of ResNet18 (Student).

Figures 1 and 2 show the loss and accuracy curves of the two models during training. It can be seen that the teacher model ResNet34 converges rapidly in a short period of time and performs stably on the validation set; while the student model ResNet18 fluctuates greatly in the early stages, it stabilizes after 10 rounds and ultimately achieves a high accuracy rate. The distillation technique effectively alleviates the problem of weak learning ability in small models.

Computational speed tests show that, compared to ResNet34, the distilled student model achieves faster inference speeds while maintaining reasonable accuracy, making it more suitable for deployment in edge computing or real-time application scenarios.

V. Discussion

The experimental results show that through knowledge distillation, ResNet18 can significantly improve classification performance, approaching the level of the teacher model. The role of KD is reflected in two aspects: first, soft targets provide more detailed distribution information between categories; second, the teacher model's confidence in specific categories helps the student model make better decisions.

Additionally, the selection of hyperparameters α and T significantly impacts the distillation effect. If T is too low, soft targets lack sufficient information; if α is set unreasonably, the student model may overly rely on information from one side, leading to performance degradation. Empirical results indicate that using T=5 and α=0.5 in this task achieves a good balance.

Regarding limitations, KD training still requires additional models to participate, which increases the overall training time. In addition, the current KD method is sensitive to training data. If the teacher model itself is overfitted, the student may inherit the bias. In the future, performance and efficiency can be further improved in the following ways:

*1) **Optimized distillation strategies**: Introducing multi-stage KD with dynamically adjusted weights (such as decoupled KD and self-distillation) allows the student model to receive more appropriate guidance signals at different training stages.*

*2) **Feature-level alignment**: Incorporating intermediate-layer alignment or adaptation modules may help the student absorb deeper knowledge beyond output logits.*

*3) **Multi-teacher ensemble distillation**: Leveraging multiple diverse teacher models may increase robustness and generalization.*

Through this project, I gained a deeper understanding of the essence and practical value of knowledge distillation. Previously, my understanding of knowledge distillation was limited to the superficial concept of "having a small model mimic the output of a large model." However, during the actual training and parameter tuning process, I gradually realized that many details in the distillation process—such as adjusting the temperature of soft targets, weight distribution, and the initialization state of the student network—significantly impact the results. This not only enhanced my understanding of deep learning model compression strategies but also taught me how to make reasonable trade-offs between accuracy and efficiency, providing valuable experience for future deployment-oriented model development.

VI. Conclusion

This project effectively achieved knowledge transfer from a large ResNet34 teacher model to a lightweight ResNet18 student model by applying classic knowledge distillation methods in image classification tasks. Experimental results show that the distilled student model maintains high classification accuracy while significantly reducing the number of parameters and model size, verifying the practicality of KD technology in model compression.

Future research can further explore more flexible distillation algorithms (such as self-distillation and decoupled KD) to improve the performance of student models in more complex tasks and real-world deployment scenarios. Additionally, combining distillation strategies with model pruning and quantization techniques may further enhance overall system efficiency.

Overall, this project demonstrates the significant potential of knowledge distillation as an efficient and versatile model compression method in the lightweight optimization and deployment of deep learning models, laying a solid foundation for future applications in edge computing, mobile devices, and real-time systems.

References

[1] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," arXiv:1503.02531, 2015.

[2] A. Romero et al., "FitNets: Hints for thin deep nets," in *International Conference on Learning Representations*, 2015.

[3] S. Zagoruyko and N. Komodakis, "Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer," in International Conference on Learning Representations, 2017.

[4] W. Park et al., "Relational knowledge distillation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[5] Y. Tian, D. Krishnan, and P. Isola, "Contrastive representation distillation," in International Conference on Learning Representations, 2020.

[6] S. I. Mirzadeh et al., "Improved knowledge distillation via teacher assistant," in *AAAI Conference on Artificial Intelligence*, 2020.

[7] K. He et al., "Deep residual learning for image recognition," in IEEE Conference on Computer Vision and Pattern Recognition, 2016.

[8] Z. Hao et al., "One-for-All: Bridge the Gap Between Heterogeneous Architectures in Knowledge Distillation," in *Advances in Neural Information Processing Systems*, 2023.