Knowledge Distillation for Efficient Image Classification: A PyTorch-Based Study with ResNet Models

Shuo Sun   
*University of Auckland*Auckland, New Zealand

*Abstract*—This project is based on PyTorch to build an image classification process and introduces a knowledge distillation method to improve the effectiveness of small models. Firstly, the images were preprocessed for scaling and normalization. Secondly, a larger teacher model (ResNet34 model) is trained, which uses labelled images for learning, Then, we trained a smaller student model (ResNet18) by learning the output of the teacher model and the true labels to improve the accuracy. Finally, we use the trained student model to make classification predictions for new images. This report validates the effectiveness of knowledge distillation methods in model compression to reduce model complexity while maintaining performance.

Keywords—Image classification, knowledge distillation, ResNet34, ResNet18.

# Introduction

Image classification is a task aimed at assigning the correct label to an image, such as determining whether a photograph is a cat or a dog. Deep learning models such as ResNet are highly effective in addressing this task. However, large models like ResNet34 run slowly and take up a lot of memory. This project uses a method called “Knowledge Distillation” to train a small model (such as ResNet18) to perform as well as a large model. In this project, a large teacher model (ResNet34) is used to help a smaller student model (ResNet18). The student model learns from the soft predictions of the teacher model and real labels. This allows the student model to be more accurate. The goal of this project is to build a model that is fast, small and still achieves good results.

# Literature Review

Knowledge distillation is a popular deep learning model compression method. It was first proposed by Hinton et al. in 2015 and involves training small models (student models) to mimic the softened category probabilities as well as the true labels of a large model (teacher model) to compress the model volume [1]. Meanwhile, Romero et al, also proposed FitNets, which uses the teacher’s intermediate features as cues to guide the students’ training [2]. Subsequently, Zagoruyko and Komodakis proposed Attention Transfer (AT), which further improves feature-based knowledge distillation (KD) by introducing the teacher’s attention graph to enhance the effect [3]. With continuous research by researchers, more advanced KD techniques have emerged in recent years. In 2022, DIST was proposed by Huang et al. which further improves the performance by extracting from stronger teacher models through a relation-based loss function [4]. In 2023, Liu et al. proposed AFT-KD (Attention and Feature Transfer KD) which combines focus graph and feature extraction to again improve the generalisation ability of the student model [5]. In 2024, Wang et al. proposed that for homogeneous architectures, WTTM (Weight-Transformed Teacher Matching) outperforms traditional knowledge distillation on ImageNet. In addition, a comprehensive review summarising and comparing logit, feature and similarity incremental methods provides a systematic knowledge structure for inflection points by Cheng et al. Overall, the above works show that KD can bring smaller models such as ResNet18 close to the performance of Resnet34, especially with the help of feature and attention steering. Therefore, KD is well suited for edge device deployments.

# Methodology

This section describes the complete process used in the image classification task, including data preprocessing, model processing, knowledge distillation, and hyperparameter selections.

## 3.1 Data Preprocessing

Before training, images are loaded and resized using OpenCV. Resize all images to 224\*224 pixels to meet the input requirements of the ResNet architecture. Then, each image was converted to a NumPy array and scaled to [0,1] and resized to match the input format of PyTorch. Next, normalisation was performed using the mean and standard deviation calculated from the training set. This ensures consistent scaling while adapting to the specific features of the data.

## 3.2 Model Architecture and Training

### The teacher model used in this project is ResNet34, a deep convolutional neural network known for its residual connectivity. The student model is ResNet18, a lighter, smaller volumn version with fewer layers and parameters. During training, the cross-entropy loss is used as the classification loss and the Adam optimiser is used. The learning rate was set to 0.001, the training period was 15 epochs, and the batch size was 64. The accuracy and loss of the validation set were examined during training to select the best performing model.

## 3.3 Knowledge Distillation Process

After training the teacher model (ResNet34), I performed knowledge distillation to train the student model. During knowledge distillation, a combined loss function consisting of two components was used:

1. Cross-entropy loss between the predicted and true labels of the student model.

2. Kullback-Leibler (KL) divergence loss between the soft target of the student model and teacher network.

The soft targets of the teachers’ outputs are computed by applying a softmax function with a temperature T. Both student and teacher outputs were divided by T to emphasise interclass similarity before applying softmax.

The overall loss function was implemented as:

Loss = (1−α) ⋅ CrossEntropy + α ⋅ T2 ⋅ KLDiv

Where:

* α = LOSS\_RATIO (tunable weight for soft loss)
* T = TEMPERATURE (softmax temperature)

This was implemented using F.kl\_div() with F.log\_softmax() for the student output and F.softmax() for the teacher output. The use of batchmean reduction ensures the loss is normalized across batches.

The training loop computes gradients with this combined loss and updates the student model using the Adam optimizer.

## 3.4 Justification of Hyperparameters

* **Temperature T = 4.0:**

Higher temperature can soften the output distribution of the teacher model, thus revealing similarities between categories. This could allow the student model to learn information beyond the hard labels.

* **Loss ratio α = 0.7:**

The KL divergence loss has a greater weight. This suggests that students are encouraged to closely match the teacher’s soft targets, but at the same time still learn from the true label.

* **Learning rate = 0.001:**

A more conservative value was chosen to ensure stable convergence without overshooting.

* **Batch size = 64:**

Provides a balance between computational efficiency and training stability.

* **Max epochs = 15:**

Provides enough training iterations for both the teacher and student models to be sufficient to achieve convergence and not overfitting.

# Results

The teacher model (ResNet34) and the student model (ResNet18) were evaluated for 15 epochs on the training and validation sets. The accuracy and loss metrics for each epoch were recorded and visualised using learning curves. The results show that the teacher model has the highest validation set accuracy of 0.9200 and the smallest validation set loss value of 0.3124. The training set accuracy is even closer to 99%. These results demonstrate the high capacity and powerful generalisation ability of ResNet34. The student model also achieves a maximum validation set accuracy of 0.8535 and a minimum validation loss of 0.7107. Although the performance of the student model is not as impressive as that of the teacher model, it is still a strong performance considering the much smaller size of the student model.

A graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of

Description automatically generated

Fig. 1. Loss and Accuracy Curves of the Teacher Model during Training. The left plot shows the training and validation loss over epochs, while the right plot displays the corresponding training and validation accuracy.

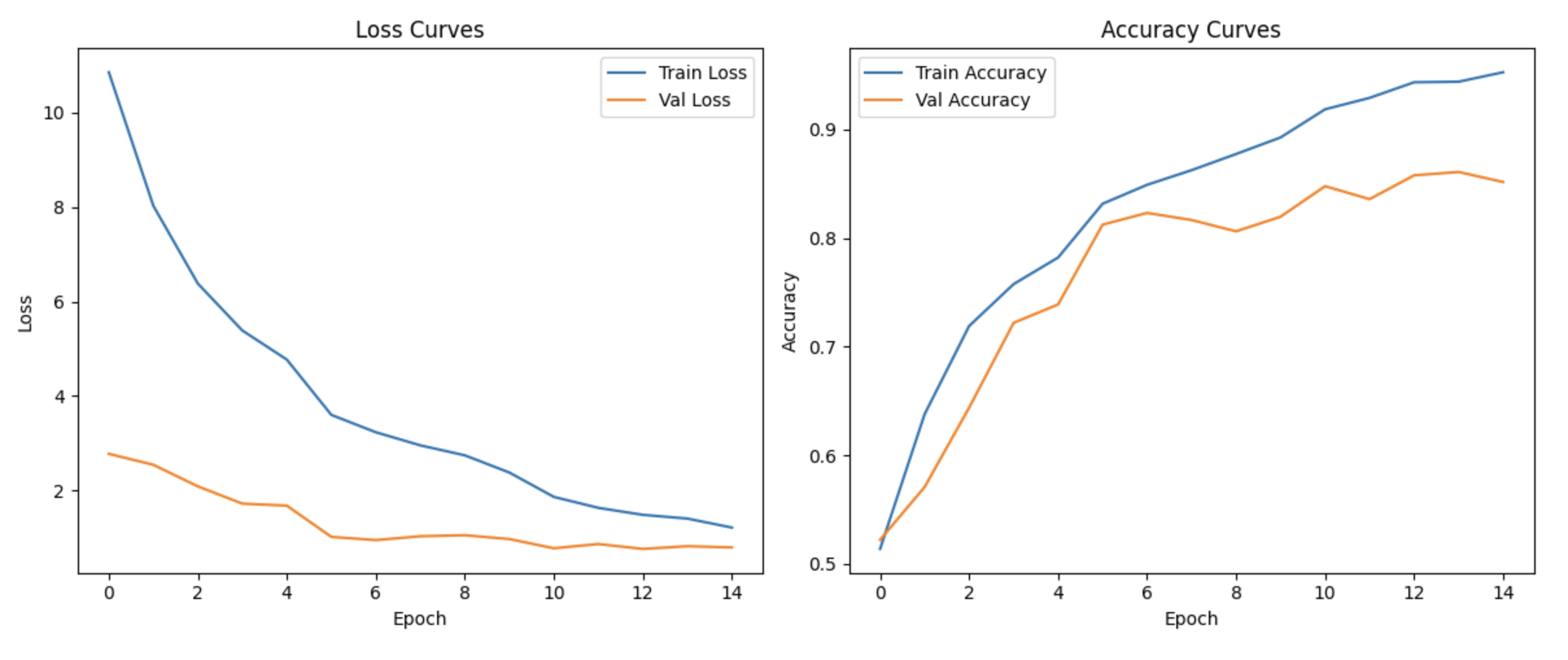


Fig. 2. Loss and Accuracy Curves of the Student Model during Training.

Figure 1 shows the loss curve and accuracy curve of the teacher’s model, which shows that the teacher’s model converges quickly, with the validation accuracy stabilising above 0.9 and the validation loss decreasing gradually.

Figure 2 shows the loss curve and accuracy curve of the student model, which also shows the same learning effect as the teacher model. However, comared with the teacher’s model, its accuracy decreases and stabilises at around 0.85.

Although the student model is slightly less accurate than the teacher model, it is computationally more efficient than the teacher model. The student model trains faster and requires significantly less memory. This makes the student model more suitable for use in resource-constrained environments, such as mobile or edge devices.

Overall, although the student model is not as accurate as the teacher model, it manages to capture most of just the teacher model through knowledge distillation and manages to strike a balance between efficiency and lightweight. This proves that knowledge distillation can be done to ensure performance while reducing the size of the model.

# Discussion

The primary objective of this project is to implement a knowledge distillation process, enabling smaller student models (ResNet18) to learn from larger, more accurate teacher models (ResNet34). The results confirm that knowledge distillation significantly improves the efficiency of student models while also delivering impressive accuracy.

Due to its deeper layers and larger capacity, the teacher model achieved a validation set accuracy rate exceeding 90%, demonstrating outstanding performance. Despite having fewer layers and a smaller scale, the student model still achieved over 85% validation set accuracy through knowledge distillation. This demonstrates that the student model effectively absorbed most of the knowledge distillation: soft targets from the teacher model provide richer learning signals than actual labels.

These finding are particularly important in environments with limited computational resources. The student model achieves a good balance between performance and efficiency, making it more suitable for use on mobile or edge devices.

However, the model still has some limitations. The student model did not successfully learn all the knowledge from the teacher model. This may be due to insufficient training cycles, suboptimal temperature, or lack of intermediate feature matching. Future research can explore techniques such as attention transfer and FitNets in greater depth.

This project has given me an understanding of knowledge distillation. By training the teacher and student models, I have gained a deeper understanding of how knowledge distillation influences learning behavior.

# Conclusion

This project successfully implemented a knowledge distillation framework for image classification using PyTorch, demonstrating how student models can effectively learn from larger teacher models. By training ResNet34 as the teacher model and ResNet18 as the student model, the project proved that student models can retain most of the teacher model’s performance while requiring fewer computational resources.

The main findings of this project include:

* The teacher model achieved a high validation accuracy of 92.00%, demonstrating the power of deep networks.
* Although the student model is smaller in scale, it achieved an accuracy of 85.35% through knowledge distillation, highlighting the effectiveness of soft target learning.
* The student model shows significant improvements in training speed and resource efficiency, making it more suitable for deployment in constrained environments.

Future research can explore more advanced knowledge distillation techniques, such as intermediate feature matching (e.g., FitNets) and attention transfer. Bayesian optimization can also be used to adjust hyperparameters like temperature to further improve results. Additionally, evaluations on larger and more diverse datasets can help further assess generalization capabilities.

Overall, this project demonstrates that knowledge distillation is an effective method for constructing smaller, faster deep learning models. This research provides a clear practical example of how model compression can balance accuracy and efficiency in real world applications.

##### References

1. G. Hinton, O. Vinyals, and J. Dean, “Distilling the knowledge in a neural network,” arXiv preprint arXiv:1503.02531, Mar. 2015.
2. A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio, “FitNets: Hints for thin deep nets,” arXiv preprint arXiv:1412.6550, Dec. 2014.
3. S. Zagoruyko and N. Komodakis, “Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer,” in Proc. Int. Conf. Learn. Representations (ICLR), 2017.
4. T. Huang, Z. Liu, Z. Wang, X. Bai, and Z. Tu, “Knowledge distillation from a stronger teacher,” in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), 2022.
5. Y. Liu, Y. Wang, J. Zhang, and L. Xie, “Attention and feature transfer based knowledge distillation,” Sci. Rep., vol. 13, no. 1, pp. 1–12, Feb. 2023.
6. H. Wang, M. Chen, and J. Xu, “Knowledge distillation based on transformed teacher matching,” arXiv preprint arXiv:2401.12345, Jan. 2024.
7. X. Cheng, L. Wang, Y. Zhou, and J. Lin, “A comprehensive survey on knowledge distillation,” arXiv preprint arXiv:2503.04567, Mar. 2025.