Project A: Knowledge Distillation for Building Lightweight Deep Learning Models in Visual Classification Tasks

Zhenhuan Sun

Department of Electrical & Computer Engineering, University of Toronto

Introduction

In recent years, Deep Neural Networks (DNNs) have become immensely popular and have found extensive use in various applications spanning different fields. Their proficiency in learning complex patterns and making accurate predictions can be attributed to their deep, layered structures, which enable them to model high-dimensional, non-linear relationships in data. However, accompanying this desirable capability are certain drawbacks arise from their sophisticated structures:

- **High Computational Demand** DNNs typically consist of millions of interconnected neurons, with parameters and computation associated with each connection, demanding considerable computational resources for training and inference.
- **Resource Intensive** The large size and complexity of DNNs necessitate a significant amount of memory for storage and considerable processing power for computation.
- **Rapid Energy Consumption** Executing DNNs models drains battery life rapidly.

All the essential components necessary for DNNs to operate effectively are typically absent in resource-constrained devices, i.e., mobile devices. Consequently, deploying DNNs on such devices is often impractical, and in cases where it is feasible, the resulting Quality of Experience (QoE) for users tends to be poor.

Objective

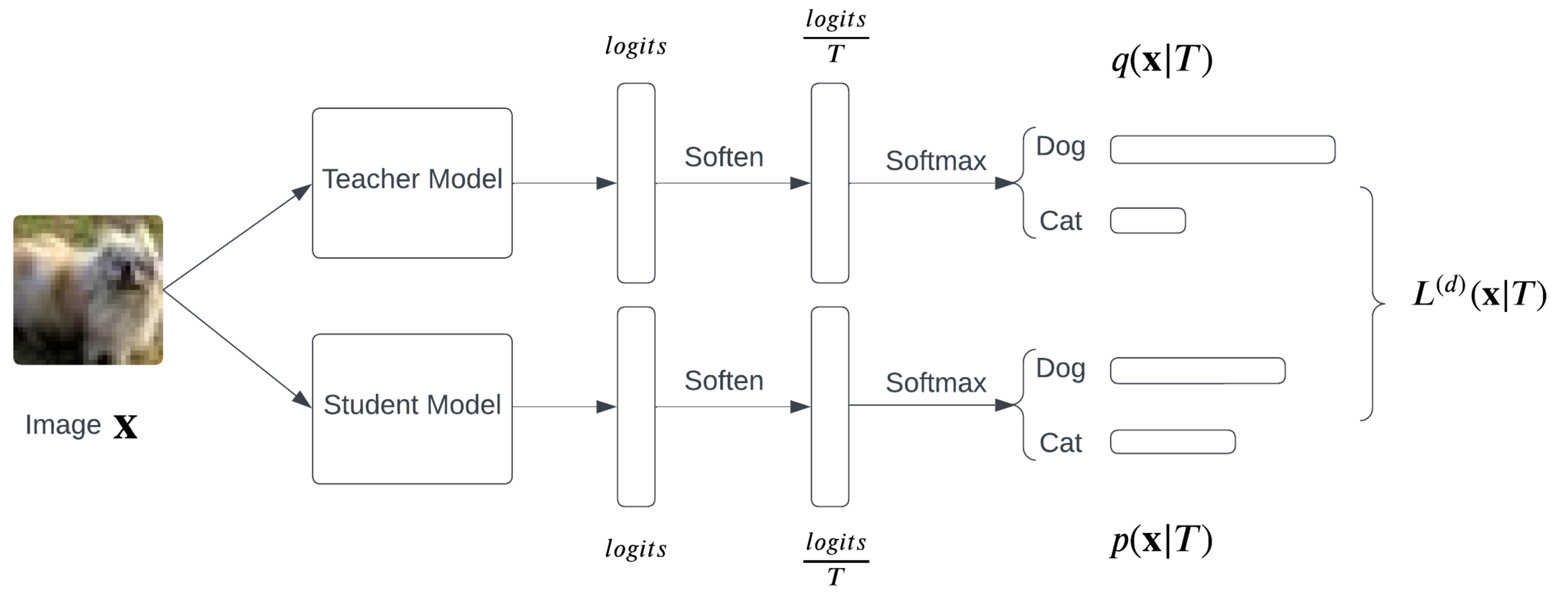
In this project, to facilitate the deployment of DNNs models on resource-constrained devices, we investigate the **knowledge distillation** process, a method known for its ability to create models that are both high-performing and resource-efficient. Specifically, through this project, we aim to answer the following three questions:

- What is the knowledge distillation process and how does it work?
- What are the advantages of employing knowledge distillation?
- What disadvantages are associated with the use of knowledge distillation?

In this project, we investigate two distinct knowledge distillation approaches: **conventional knowledge distillation** as introduced by Hinton et al. [1] and **subclass distillation** as proposed by Müller et al. [2].

Conventional Knowledge Distillation

In the conventional knowledge distillation, knowledge is defined as the **softened class probabilities** generated by the teacher model.

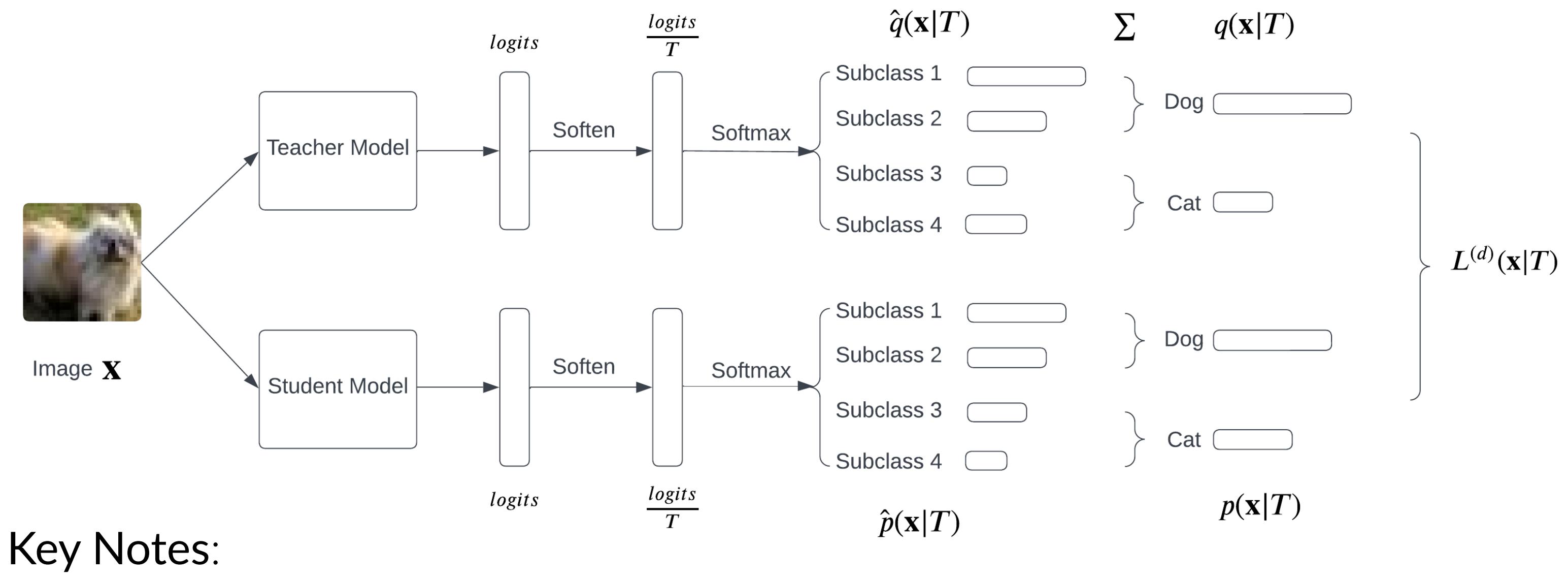


Student model is trained to generate a softened class probability distribution that closely matches the softened probability distribution produced by the teacher model through minimizing distillation loss defined as

$$L^{(d)}(\mathbf{x}|T) = -\sum_{i=1}^{K} q_i(\mathbf{x}|T)\log(p_i(\mathbf{x}|T)) \tag{1}$$

Subclass Distillation

Instead of using the softened probability distribution over the main classes as soft labels, subclass distillation forces the teacher model to divide each existing classes into multiple subclasses, and the **probability distribution over all subclasses** is transferred from the teacher model to the student model as the knowledge.



Key Notes:

1. **The Use of Softened Probability Distribution over Classes/Subclasses** Reduce the spikiness of probability distribution, thereby amplifying the differences between probabilities of different classes and enabling in a greater amount of information to be transferred as knowledge.
2. **The Use of Distillation Loss Function** Allow teacher model teach the student model how to mimic its behavior to generalize to different data as it does.

Preliminary Results

The efficacy of both conventional knowledge distillation and subclass distillation was evaluated on the MNIST and MHIST dataset.

MNIST Dataset: A Convolutional Neural Network (CNN) featuring 2 convolutional layers and 2 fully connected layers serves as the teacher model, while the student model is a Multi-layer Perceptron (MLP) comprising 3 fully connected layers.

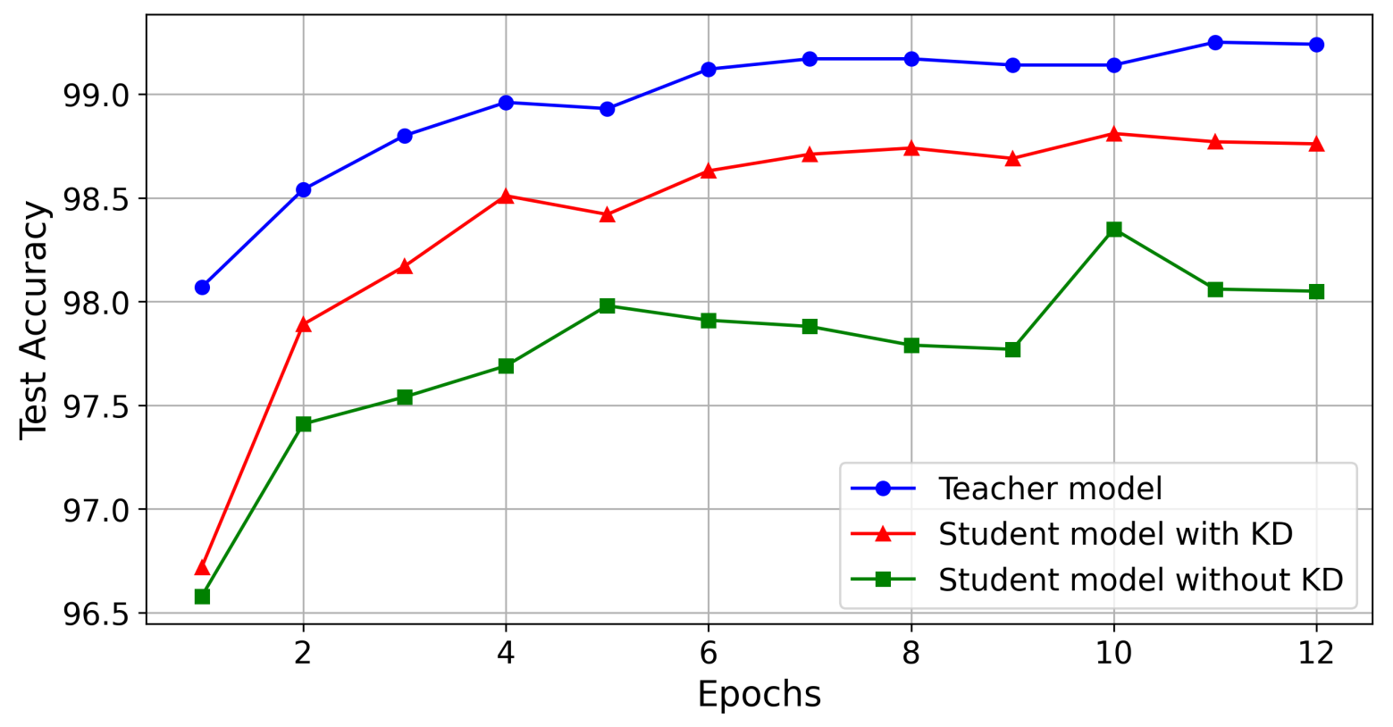


Figure 1. Conventional knowledge distillation

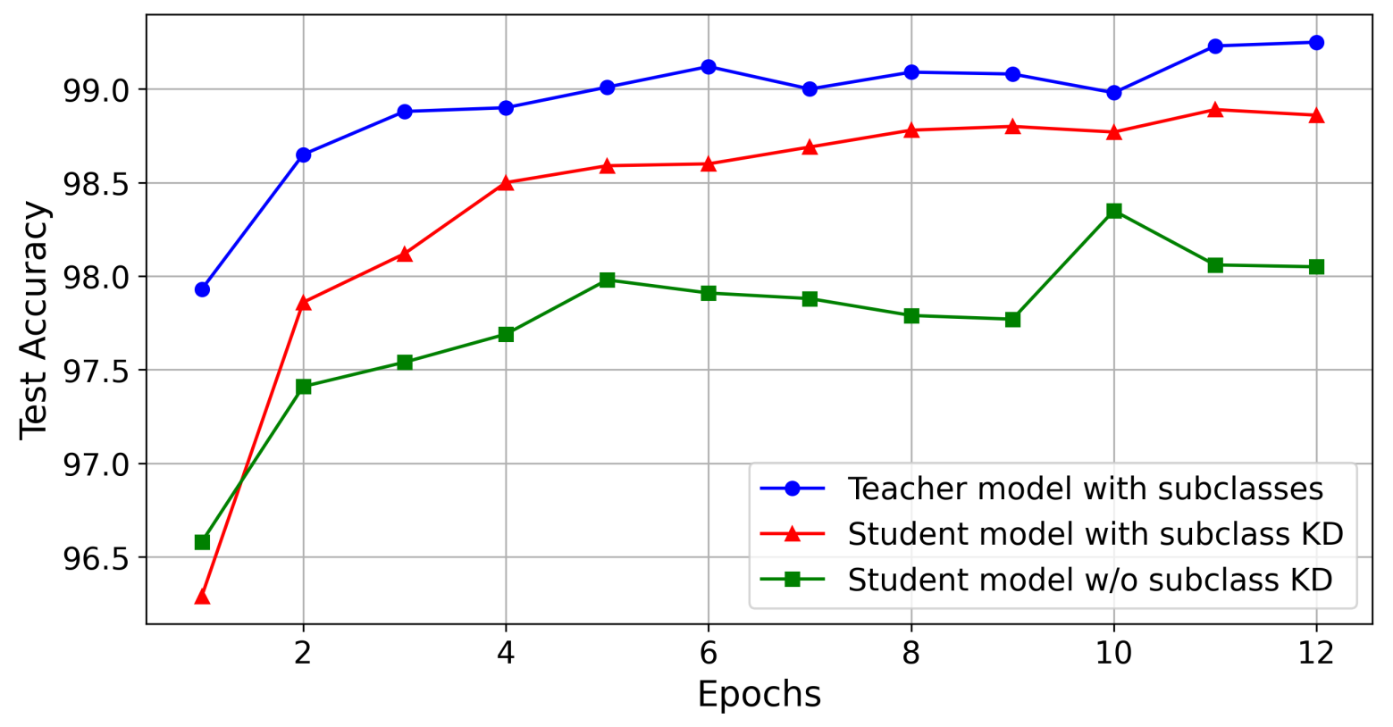


Figure 2. Subclass distillation

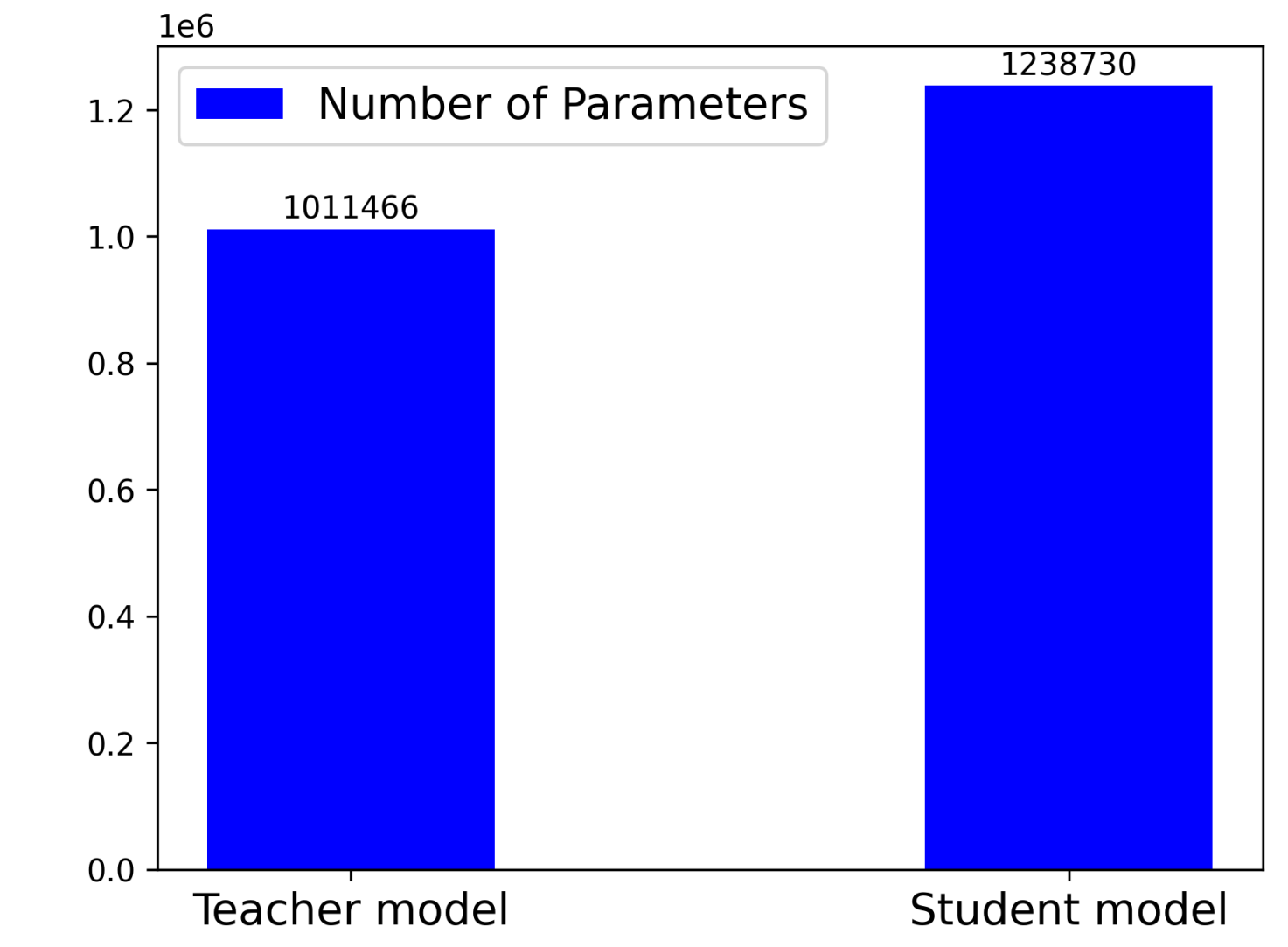


Figure 3. Number of parameters comparison

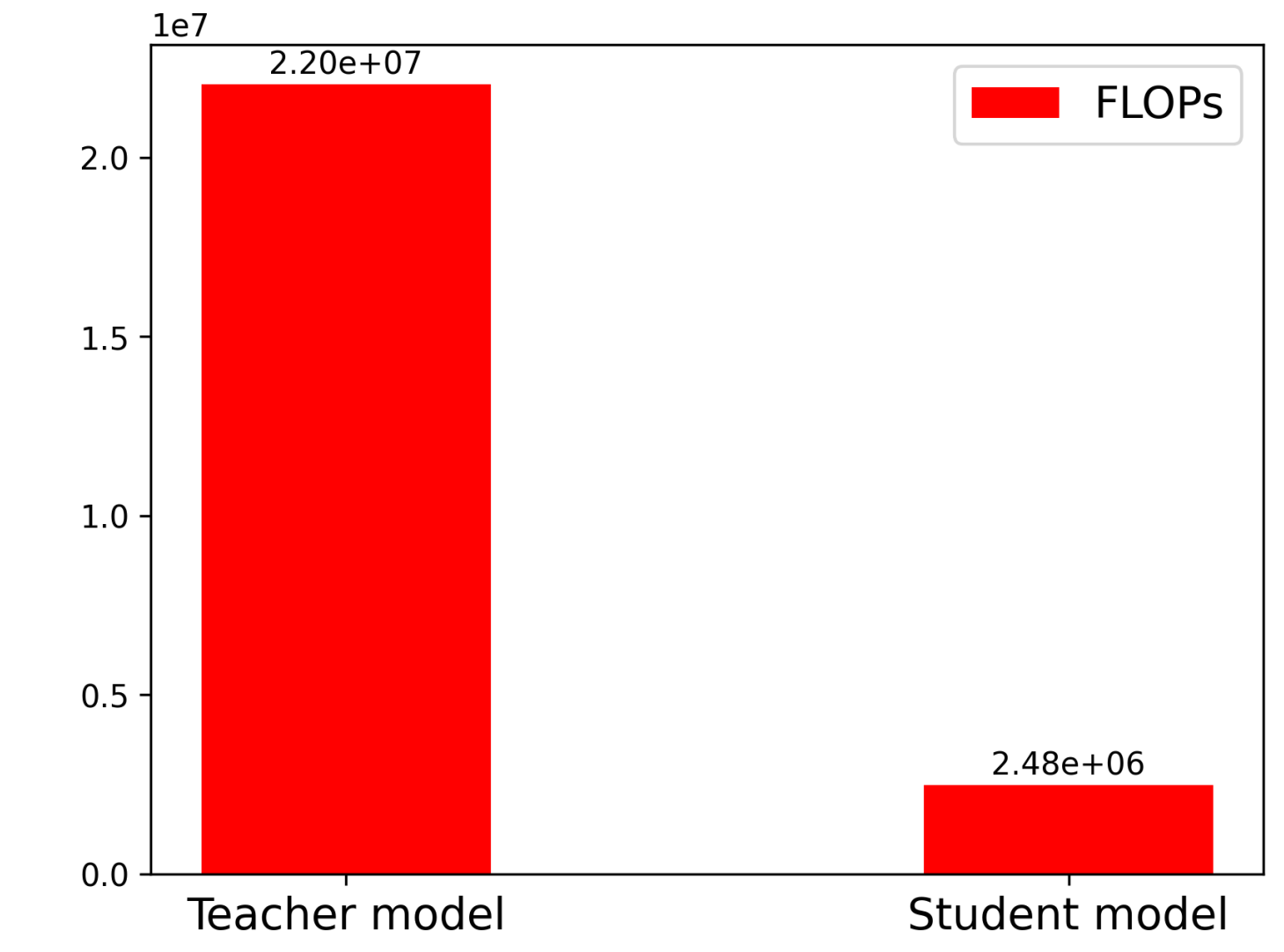


Figure 4. FLOPs comparison

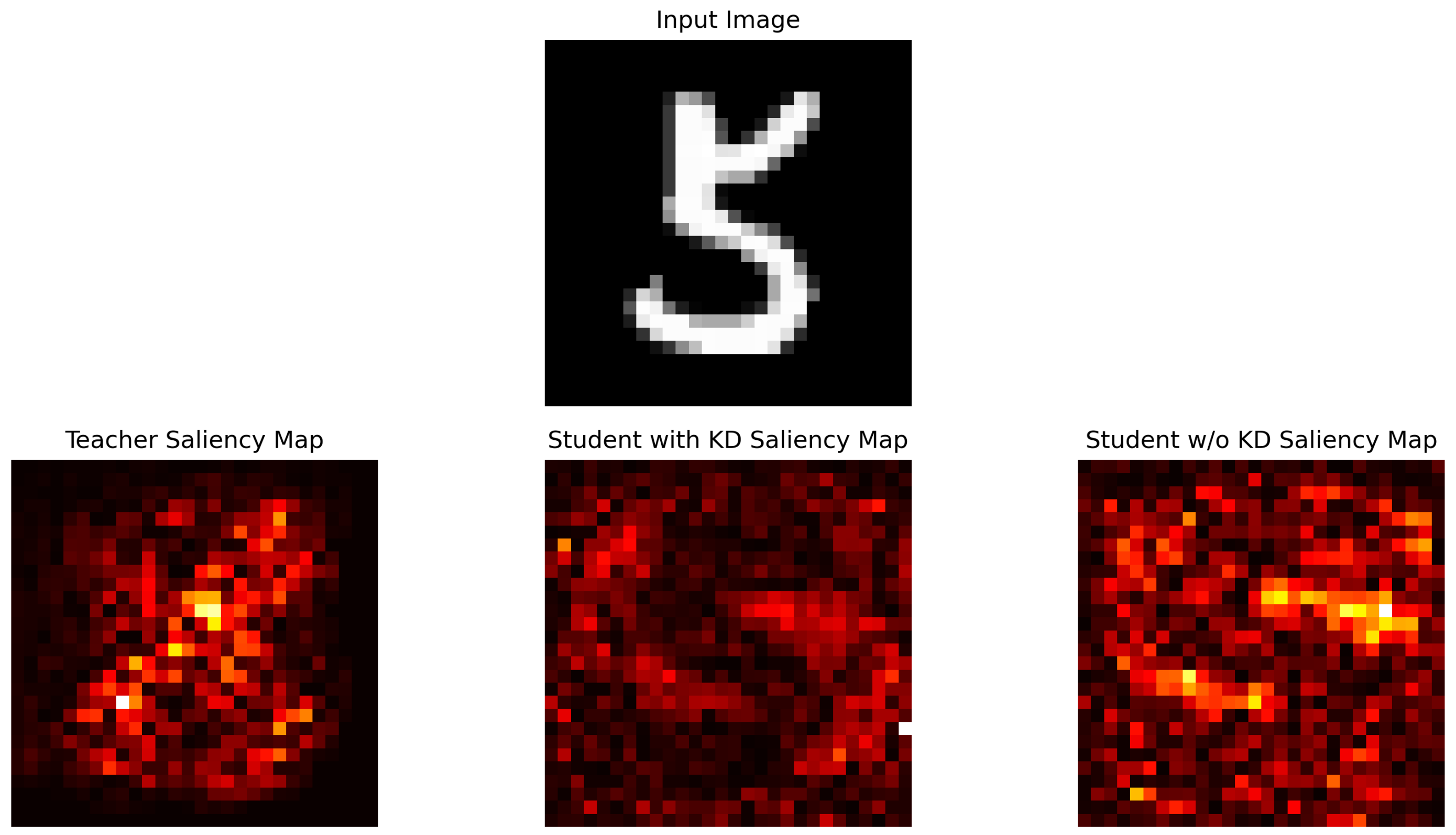


Figure 5. Saliency maps for the digit "5" generated by different models.

MHIST Dataset: A pre-trained ResNet50 model is utilized as the teacher model, while a MobileNetV2 model is adopted as the student model. While training the teacher model, we freeze the layers preceding the penultimate layer and conduct transfer learning on the MHIST dataset. A similar approach is employed in training the student model with knowledge distillation.

Table 1. Performance comparison between teacher and student models employing different knowledge distillation methods

Weighted F1 Score	Teacher	Student with KD	Student w/o KD
Conventional	0.827633	**0.728454**	0.716311
Subclass	0.842726	**0.720571**	0.716311

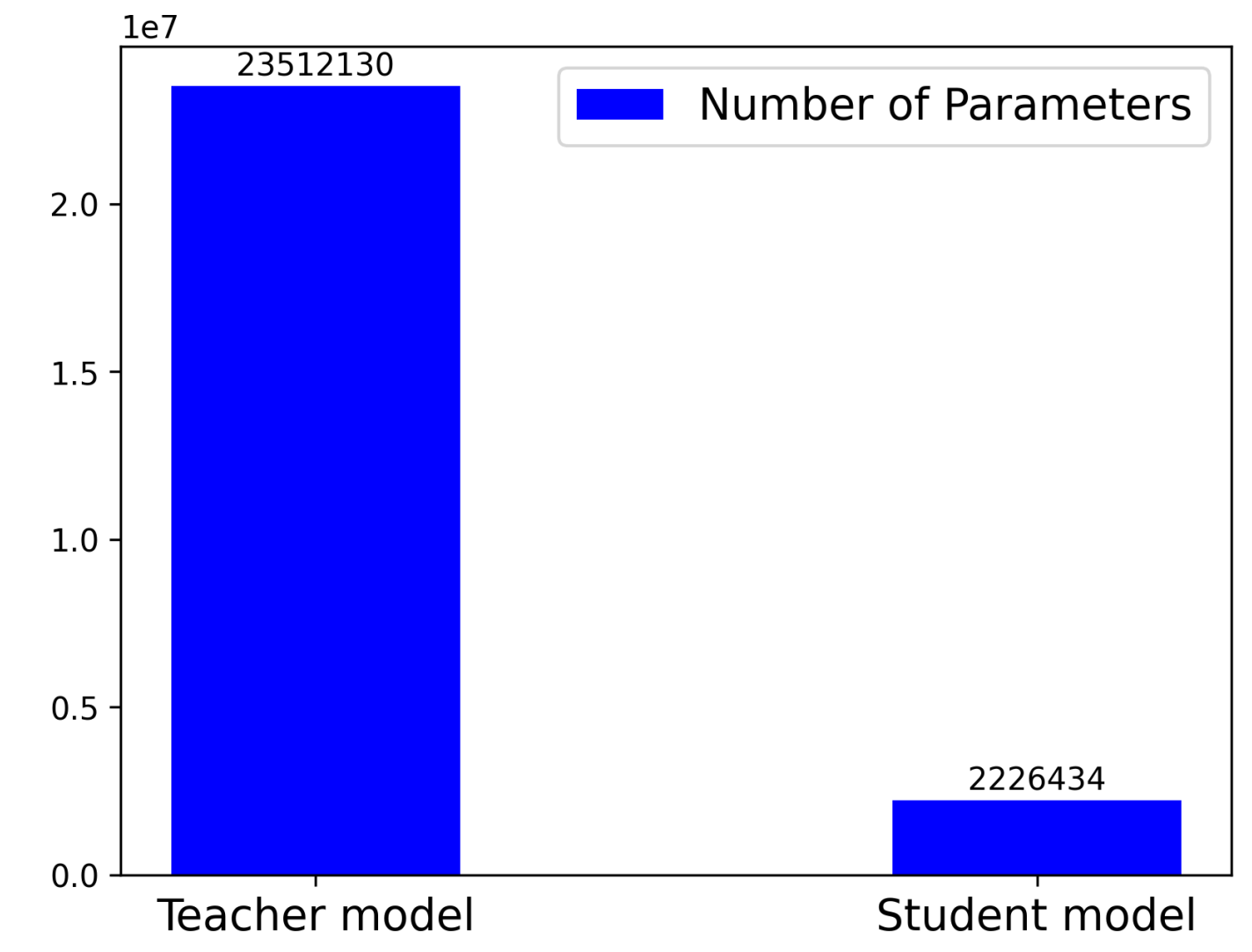


Figure 6. Number of parameters comparison

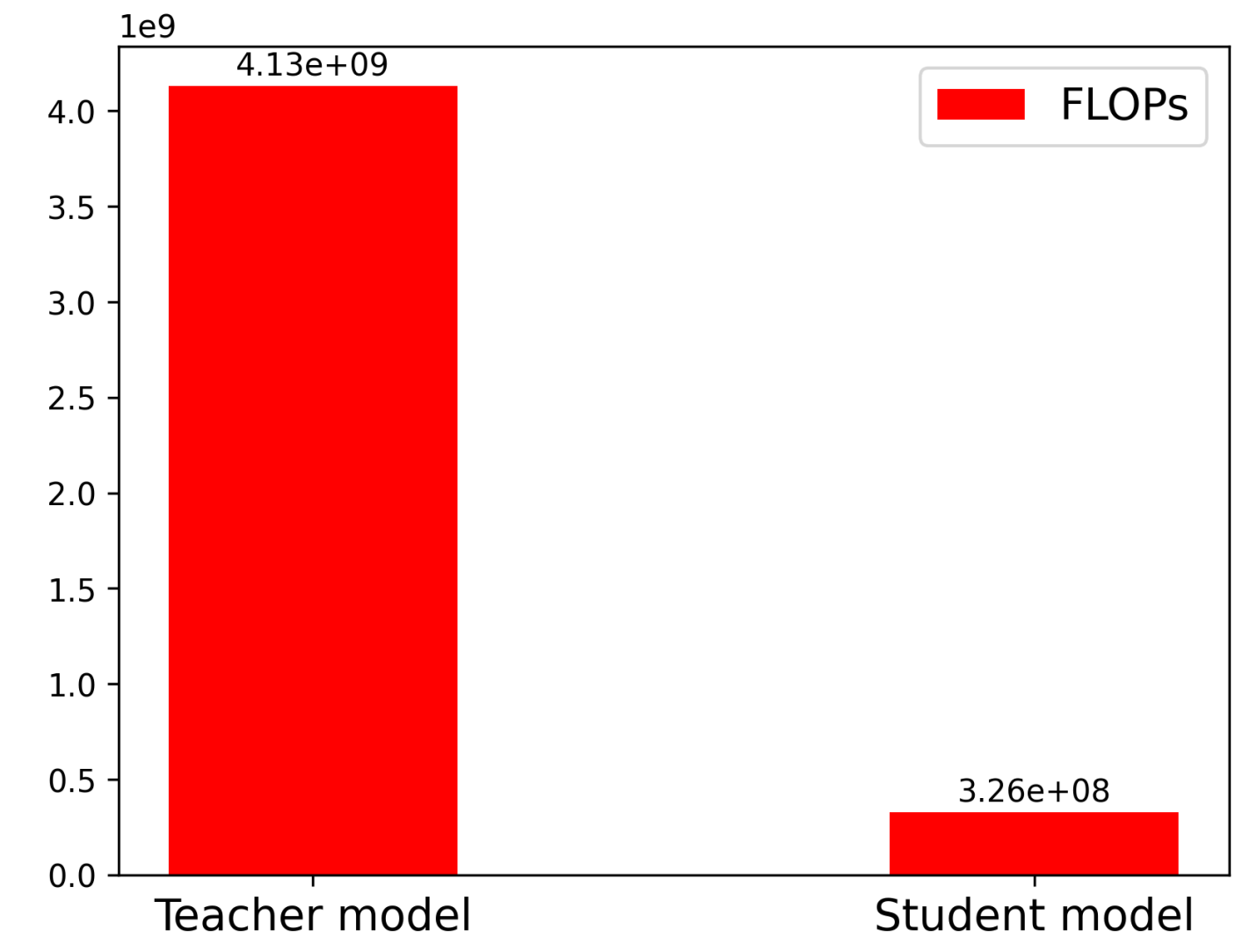


Figure 7. FLOPs comparison

Conclusion

In this project, we gained an understanding of the knowledge distillation process and its mechanisms. We tested two methods of knowledge distillation on the MNIST and MHIST datasets. The findings indicate that while knowledge distillation allows for the creation of smaller, resource-efficient models with performance comparable to larger, complex models, the interpretability of the results from these smaller models tends to be lower compared to their larger counterparts.

References

[1] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean.
Distilling the knowledge in a neural network.
arXiv preprint arXiv:1503.02531, 2015.

[2] Rafael Müller, Simon Kornblith, and Geoffrey Hinton.
Subclass distillation.
arXiv preprint arXiv:2002.03936, 2020.

Github: https://github.com/sunhuanhuang920/ECE1512_2023F_ProjectRepo_Zhenhuan_Sun/tree/main

zhenhuan.sun@mail.utoronto.ca