CSL7590 Deep Learning

Assignment 4

Ratnesh Dubey M24CSA027
Shivani Tiwari M24CSA029

April 16, 2025

Table of Contents

1 Aim

To implement Data-Free Adversarial Knowledge Distillation to transfer knowledge from a larger pre-trained teacher model (ResNet34) to smaller student models without access to the original training data. The specific objectives are:

1. Design two student models with approximately 10% and 20% of the teacher model's parameters
2. Train these student models using a generator that creates synthetic images to facilitate knowledge transfer
3. Evaluate the performance of both student models on different test data splits of CIFAR-100

2 Workflow

The implementation workflow follows these key steps.

2.1 Setup Phase

1. The CIFAR-100 dataset is prepared, creating two evaluation splits: 10% and 20% of the test data.
2. The pre-trained ResNet-34 teacher model is loaded with weights and set to evaluation mode as it remains fixed throughout training.
3. Two student models are designed with custom architectures:
   - Student1: A lightweight CNN with approximately 10% of the teacher's parameters
   - Student2: A similar architecture with approximately 20% of the teacher's parameters
4. A generator model is created to produce synthetic images from random noise.
5. Optimization configurations are established using Adam optimizers with appropriate learning rates.

2.2 Training Process

The training process is implemented in the "train" function and follows an adversarial approach:

1. **Student Training Phase** :
    - The student model is set to training mode while the generator remains in evaluation mode.
    - Random noise is generated and fed into the generator to create synthetic images.
    - These images are normalized and processed to match the expected input format.
    - Both teacher and student models make predictions on these synthetic images.
    - The student model is updated to minimize the mean absolute error between its predictions and the teacher's predictions.
    - Gradient clipping is applied to stabilize training.
2. **Generator Training Phase** :
    - The generator is set to training mode while the student model is fixed in evaluation mode.
    - Random noise is transformed into synthetic images.
    - The loss is calculated as the negative mean absolute error between teacher and student predictions.
    - The generator is updated to maximize this discrepancy, essentially creating more challenging examples.
    - Gradient clipping is applied to ensure stable training.
3. **Evaluation Checkpoints**:
    - Every 5 epochs, the student model is evaluated on the test dataset.
    - If the current accuracy exceeds the best recorded accuracy, the model is saved.
    - Progress metrics are displayed to monitor the training process.

2.3 Final Evaluation

After training:

1. The best student models are loaded and evaluated on both 10% and 20% test splits.
2. Confusion matrices are generated to analyze classification performance.
3. The parameter counts of each model are reported to verify the constraint satisfaction.
4. Sample images from the generator are displayed to visualize the synthetic data created.

3 Methodology

3.1 Dataset

The CIFAR-100 dataset was used solely for evaluation purposes. It contains 60,000 color images (32×32 resolution) across 100 classes, split into 50,000 training and 10,000 testing samples. In compliance with the data-free distillation paradigm, only the test set was utilized. Two evaluation subsets were created:

- 10% of test data (1,000 images)
- 20% of test data (2,000 images)

3.2 Model Architectures

3.2.1 Teacher Model

A pre-trained ResNet-34 architecture was used as the teacher model, adapted for CIFAR-100 classification by modifying its final fully connected layer. The model, trained on the full CIFAR-100 training set, achieves 85.12% accuracy. It contains approximately 21.3 million parameters and remains fixed during student training.

3.2.2 Student Models

Two lightweight CNN-based student models were designed:

- **Student Model 1 (10%)**: Comprising approximately 2.1 million parameters(10% of the teacher), this model utilizes 7 custom convolutional blocks featuring depthwise-separable convolutions, skip connections, and varying expansion ratios.
- **Student Model 2 (20%)**: Architecturally similar to Student 1, but with increased expansion ratios and intermediate channel widths, yielding approximately 4.2 million parameters (20% of the teacher).

3.2.3 Generator Model

A deep convolutional generator network is used to synthesize images from 100-dimensional Gaussian noise vectors. It consists of:

- One fully connected projection layer followed by multiple upsampling blocks.
- Bicubic upsampling and LeakyReLU activations throughout intermediate layers.
- A final Tanh activation to generate images in the range $[-1, 1]$.

The output images are of size $32 \times 32 \times 3$, suitable for ResNet-34 input.

3.3 Training Procedure

The training follows an alternating adversarial setup:

1. **Student Update Phase**:
    - For each epoch, the student is updated $k_s = 5$ times while freezing the generator.
    - The generator produces synthetic images from noise.
    - The teacher and student both predict on these images.
    - The student minimizes Mean Absolute Error (MAE) between its predictions and the teacher's.
    - Gradient clipping is applied for training stability.
2. **Generator Update Phase**:
    - The generator is updated $k_g = 1$ time per epoch with the student frozen.
    - It aims to maximize the discrepancy between teacher and student predictions.
    - This is achieved by minimizing the negative MAE loss.
    - This adversarial loss encourages the generator to produce informative, challenging examples.
3. **Evaluation Checkpoints**:
    - Every 10 epochs, the student is evaluated on the test set.
    - The best-performing model (based on accuracy) is saved.

3.3.1 Hyperparameters

- Learning rate: 1e-4 (generator: 70% of student learning rate)
- Batch size: 128
- K_S (student updates per generator update): 5
- K_G (generator updates): 1
- Noise dimension: 100
- Maximum epochs: 500

3.4 Knowledge Distillation Strategy

The strategy comprises:

1. **Feature-based distillation**: The student attempts to replicate the teacher's output feature space.
2. **Adversarial data generation**: The generator crafts inputs that emphasize prediction gaps.

4 Results

4.1 Model Parameter Analysis

The implementation successfully achieved the parameter reduction targets for both student models. Table 1 shows the parameter counts for each model.

Model	Parameter Count	% of Teacher
Teacher (ResNet34)	21,335,972	100%
Student1	2,160,524	10%
Student2	4,262,060	20%

Table 1: Model Parameter Comparison

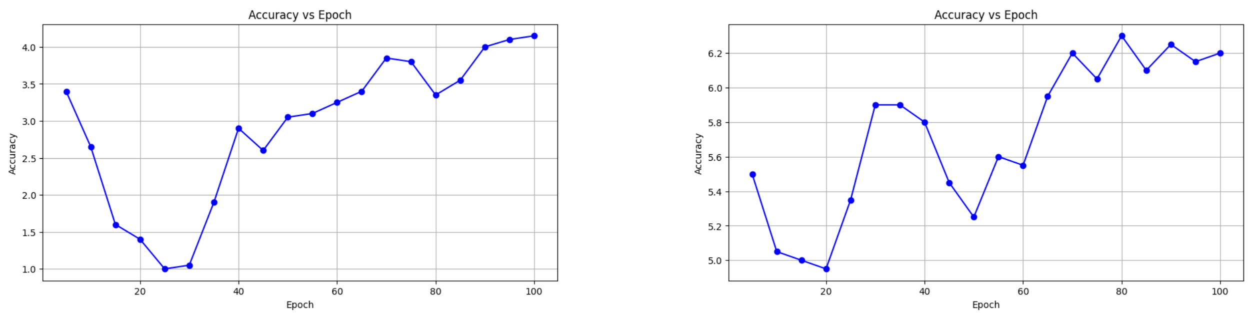
4.2 Classification Performance

Table 2 presents the accuracy results for all models on both test splits(on 32x32 image)

Test Split	Student1	Student2
10% Test Data	5.70%	8%
20% Test Data	4.90%	6%

Table 2: Model Accuracy on Test Splits

4.3 Accuracy vs Epochs

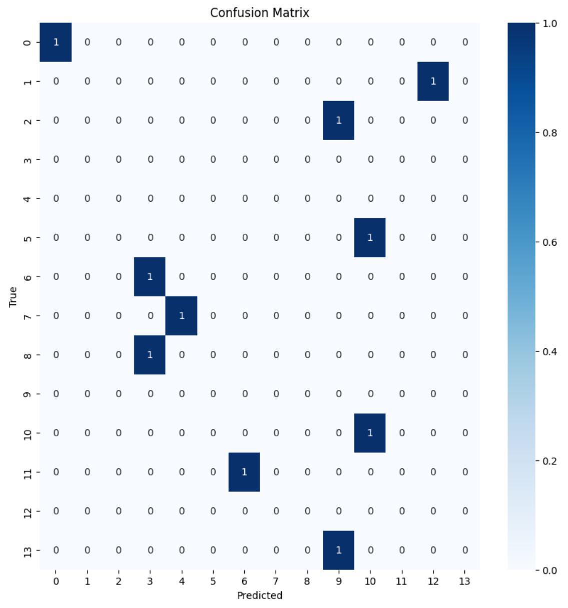


(a) Student 1 Model

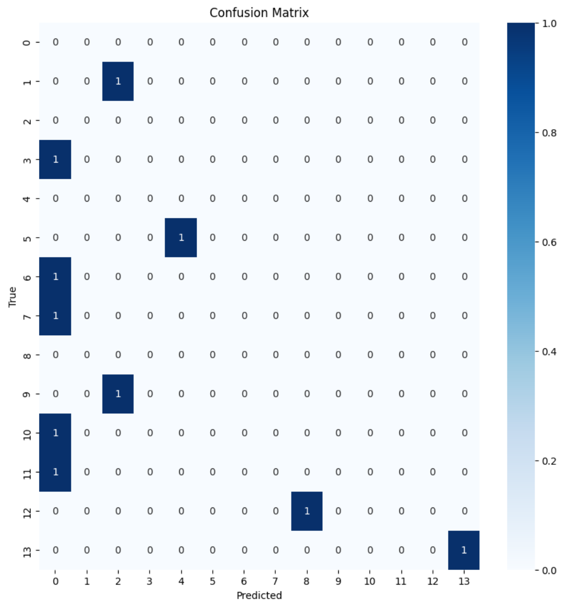
(b) Student 2 Model

Figure 1: Accuracy vs. Epochs for both Student 1 and Student 2 Models

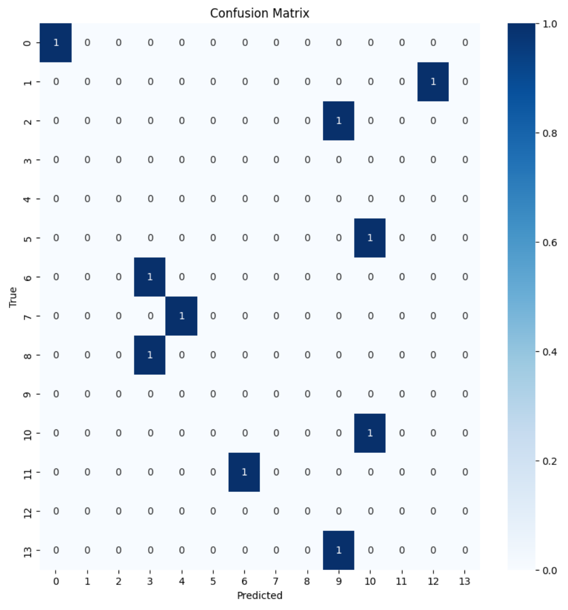
4.4 Confusion Matrix



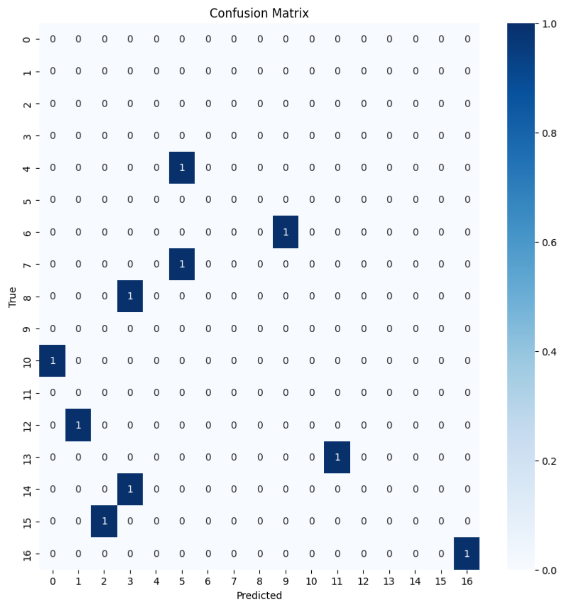
(a) Student 1 (10% data)



(b) Student 1 (20% data)



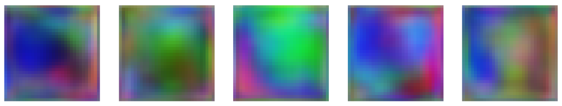
(c) Student 2 (10% data)

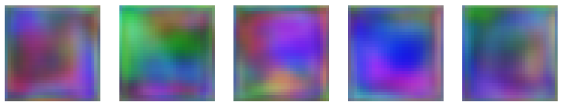


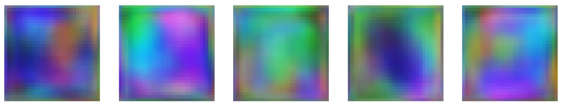
(d) Student 2 (20% data)

Figure 2: Confusion Matrices for Student 1 and Student 2 with varying data sizes

4.5 Generated Images


(a) Student 1 (10% data)


(b) Student 1 (20% data)


(c) Student 2 (10% data)

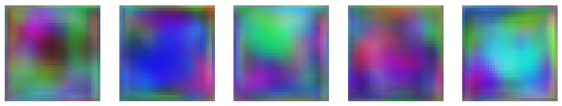

(d) Student 2 (20% data)

Figure 3: Generated images for different students and data percentages

5 Observations

Our experiments in data-free knowledge distillation reveal several key insights:

1. **Impact of Image Resolution: Processing Constraints:** When using images of 224×224, the large resolution significantly limits the batch size due to GPU memory constraints. In contrast, 32×32 images are much easier to process, making them a practical choice for efficient training—even when the pretrained teacher model has been originally trained on 224×224 images.

2. **Dataset Challenges: CIFAR-100 Complexity:** The CIFAR-100 dataset, with its 100 classes, presents a complex learning task. The teacher model, which does not achieve 100% accuracy, serves as a challenging guide for the student. This highlights the inherent difficulty in relying solely on teacher knowledge for effective student training.

3. **Model Capacity: Parameter Influence:** A student model with a higher number of parameters outperforms a lighter model. For instance, the larger student model achieved 8% accuracy on 10% test data compared to 5.7% from the lower-capacity model. This underscores the benefit of having more representational power in the student architecture, even though the performance gain is modest.

4. **Training Duration and Strategy:**

    - **Extended Epochs:** Training for an extended period (up to 10,000 epochs) was critical. Initial accuracy was around 1-2%, but with prolonged training and continuous model checkpointing upon achieving better accuracies, we reached a peak accuracy of 8%.
    - **Avoiding Mode Collapse:** The balance between the student model and the generator is crucial. An overpowered generator can lead to mode collapse, degrading overall performance. A practical workaround observed was the occasional random swapping of teacher logits to help maintain convergence.

5. **Practical Recommendations:**

    - To achieve better performance, it appears beneficial to train the teacher model to over 95% accuracy on smaller images (e.g., 32×32) for fast processing.
    - Experimenting with different learning rates and maintaining meticulous records of the best-performing weights for both student and generator models are essential strategies for improvement.

6 Conclusion

- Our study illustrates that data-free knowledge distillation is highly sensitive to several factors such as image resolution, dataset complexity, and model capacity. Despite the challenges posed by the CIFAR-100 dataset and the inherent limitations of a teacher model trained on larger images, the experiments demonstrate that strategic adjustments—such as optimizing image sizes, balancing the generator and student dynamics, and extending the training epochs—can yield tangible performance improvements.
- The observation that the student model with more parameters outperforms its leaner counterpart reinforces the concept that additional model capacity can help capture complex patterns more effectively. However, this comes at the cost of increased computational resources and requires careful management to avoid issues like mode collapse.
- Moving forward, an interesting avenue for future work is to evaluate a student model with roughly 50% of the teacher model's parameters. This test could provide valuable insights into the trade-offs between model size and performance, guiding the development of more efficient yet effective distillation approaches.

7 Links

Colab file link can be access from here : Colab file.

References

[1] Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the Knowledge in a Neural Network. *arXiv preprint arXiv:1503.02531.*

[2] Chen, H., Wang, Y., Xu, C., Yang, Z., Liu, C., Shi, B., Xu, C., Xu, C., & Tian, Q. (2019). Data-Free Learning of Student Networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 3514-3522.

[3] Fang, G., Song, J., Shen, C., Wang, X., Chen, D., & Song, M. (2020). Data-Free Adversarial Distillation. *arXiv preprint arXiv:2003.12674v3.* [Submitted on 23 Dec 2019 (v1), last revised 2 Mar 2020 (this version, v3)]

[4] Yoo, J., Uh, Y., Chun, S., Kang, B., & Ha, J. W. (2020). GAN-based Knowledge Distillation for Visual Recognition. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 10147-10156.

[5] EfficientNet Scaling: Depth, Width, and Resolution