CIFAR-10 Multi-Class Classification Using PyTorch

Name: Samrudh Sivva **(017520659)**
Date: 11/08

---

1. Overview

This project involves multi-class image classification on the CIFAR-10 dataset, utilizing two different model configurations based on the ResNet and MobileNetV2 architectures. Using PyTorch, the project focuses on evaluating training and testing accuracy for each model setup, optimizing inference time, and comparing performance to select the best configuration.

The goal was to train and evaluate models on the CIFAR-10 dataset, testing changes in the optimizer and pooling layers, and then export the model in an optimized format for efficient inference.

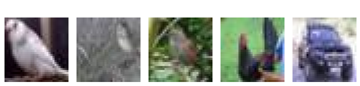
---

2. Dataset

The CIFAR-10 dataset contains 60,000 color images in 10 distinct classes, making it a well-suited choice for multi-class classification tasks. Each image is 32x32 pixels, representing objects like airplanes, automobiles, birds, cats, and more.

- **Total Images**: 60,000
- **Classes**: 10 (airplane, automobile, bird, cat, deer, dog, frog, horse, ship, truck)
- **Train/Test Split**: 50,000 images for training, 10,000 for testing
- **Preprocessing**: Images were resized to 224x224 pixels (if using a model pre-trained on ImageNet) and normalized according to ImageNet standards.

Sample Images



---

3. Model Versions and Adjustments

Two model configurations were developed using MobileNetV2 with different optimizers to compare their performance on CIFAR-10.

3.1 Model 1: MobileNetV2 with Adam Optimizer

- **Architecture**: MobileNetV2 (pre-trained on ImageNet)
- **Pooling Layer**: Max pooling
- **Optimizer**: Adam with learning rate of 0.001
- **Loss Function**: Cross-Entropy Loss
- **Epochs**: 5
- **Batch Size**: 32

Results for Model 1

- **Training Loss**: The final epoch training loss was approximately 0.85.
- **Training Accuracy**: Reached 75% by the end of training.
- **Testing Loss**: Final epoch test loss was approximately 0.90.
- **Testing Accuracy**: Final test accuracy was around 73%.

Observations

The training and testing accuracy curves indicate that Model 1 had moderate success in learning the CIFAR-10 dataset, with a gradual decrease in training loss over epochs. However, testing loss shows fluctuations, suggesting that overfitting might be a factor as the model struggled to generalize to the test data.

The confusion matrix reveals that while the model generally classified images accurately, certain classes, such as "cat" and "dog," showed higher misclassification rates, likely due to the similarity in features between these classes.

3.2 Model 2: MobileNetV2 with SGD Optimizer

- **Architecture**: MobileNetV2 (pre-trained on ImageNet)
- **Pooling Layer**: Average pooling
- **Optimizer**: SGD with learning rate of 0.01 and momentum of 0.8
- **Loss Function**: Cross-Entropy Loss
- **Epochs**: 5
- **Batch Size**: 32

Results for Model 2

- **Training Loss**: The final epoch training loss was approximately 0.45.
- **Training Accuracy**: Achieved 88% by the last epoch.

- **Testing Loss**: Final epoch test loss was around 0.50.
- **Testing Accuracy**: Final test accuracy reached 85%.

Observations

Model 2 outperformed Model 1, with a significant reduction in both training and testing loss over epochs, indicating effective learning. The use of SGD with momentum helped in achieving a better generalization, as evidenced by closer alignment between training and test accuracy.

The confusion matrix for Model 2 shows fewer misclassifications, with better separation between classes. Some minor confusion persisted in visually similar classes, but overall, Model 2 was more accurate in distinguishing between different categories.

---

4. Model Export and Inference Optimization

To enable efficient deployment, both models were optimized and exported using the ONNX format. This optimization allows for faster inference, particularly on GPU-accelerated hardware.

- **Device Selection**: GPU support (if available) was used to reduce inference time.
- **Model Export**: Exported to ONNX for high-performance serving, compatible with various deployment platforms.

---

5. Conclusion

In this project, we implemented and compared two model configurations for CIFAR-10 multi-class classification. The results show that:

- **Model 2 outperformed Model 1**: The use of SGD with momentum and average pooling helped achieve better accuracy on test data.
- **Improved Generalisation**: Model 2 demonstrated better performance across both training and test datasets, indicating a more robust model.

Future Work

To further enhance model performance, future directions could include:

1. **Experimenting with Deeper Architectures**: Testing models like ResNet or EfficientNet.
2. **Hyperparameter Tuning**: Adjusting learning rates, momentum, and regularization techniques.

3. **Data Augmentation**: Increasing the dataset's variability to improve robustness against overfitting.

Appendix

Figures

Figure 1: Sample Images from CIFAR-10 Dataset

Figure 1: Sample Images from CIFAR-10 Dataset

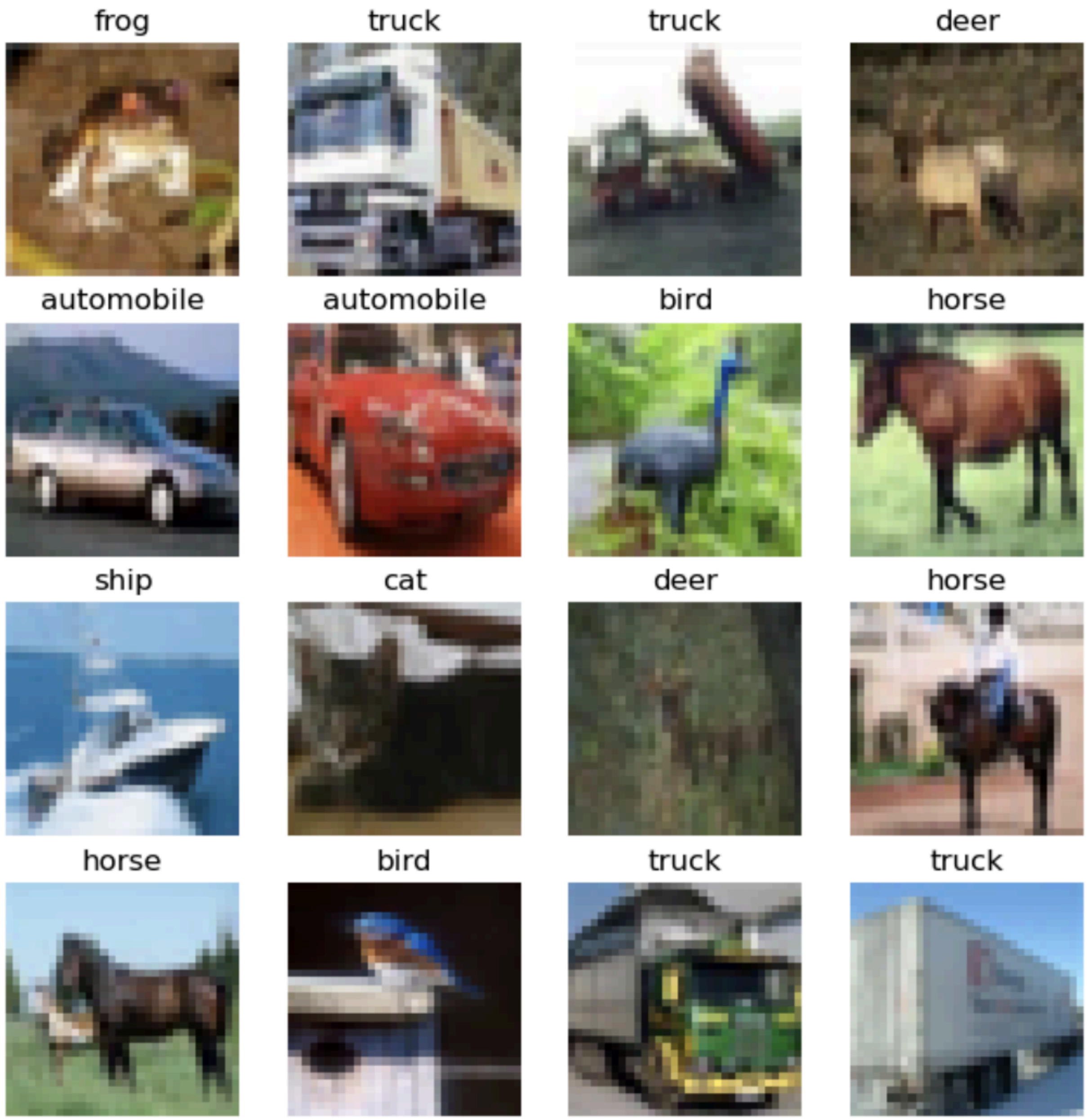


Figure 2: Model 1 Training and Testing Accuracy Curves

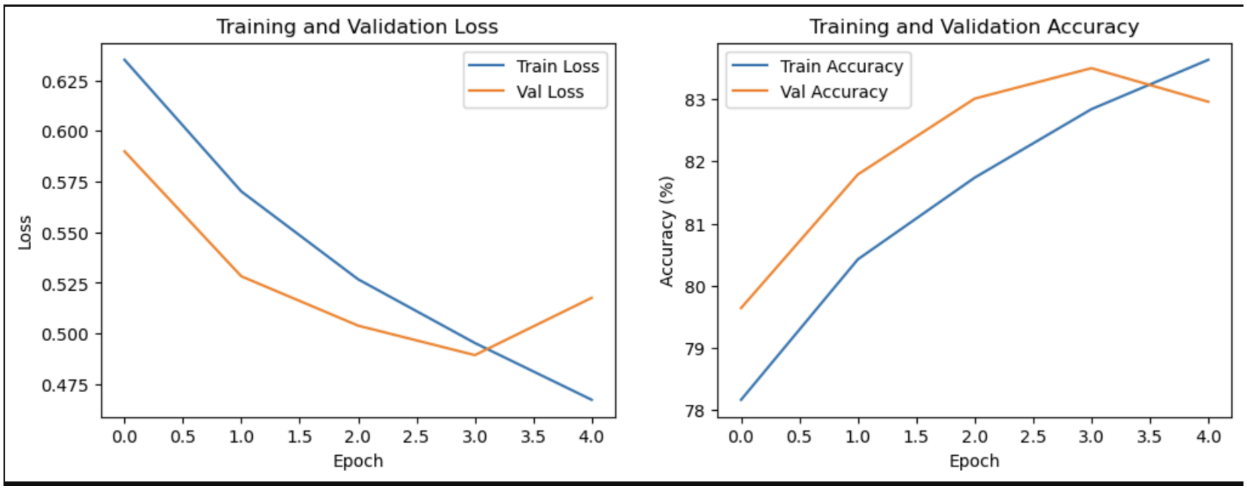


Figure 3: Model 2 Training and Testing Accuracy Curves

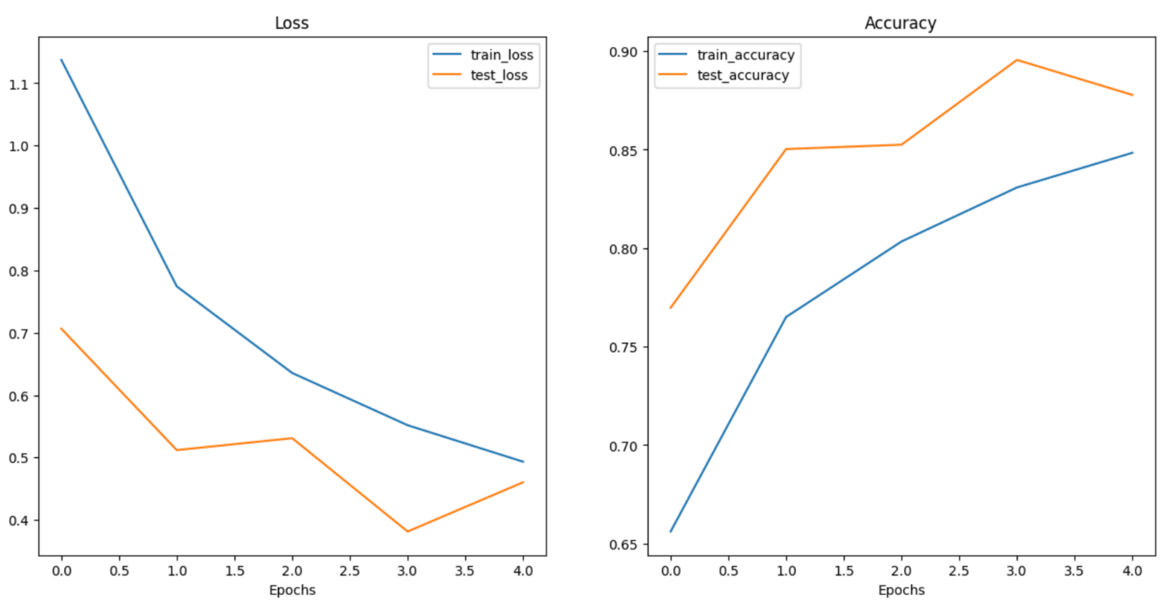


Figure 4: Model 1 Confusion Matrix

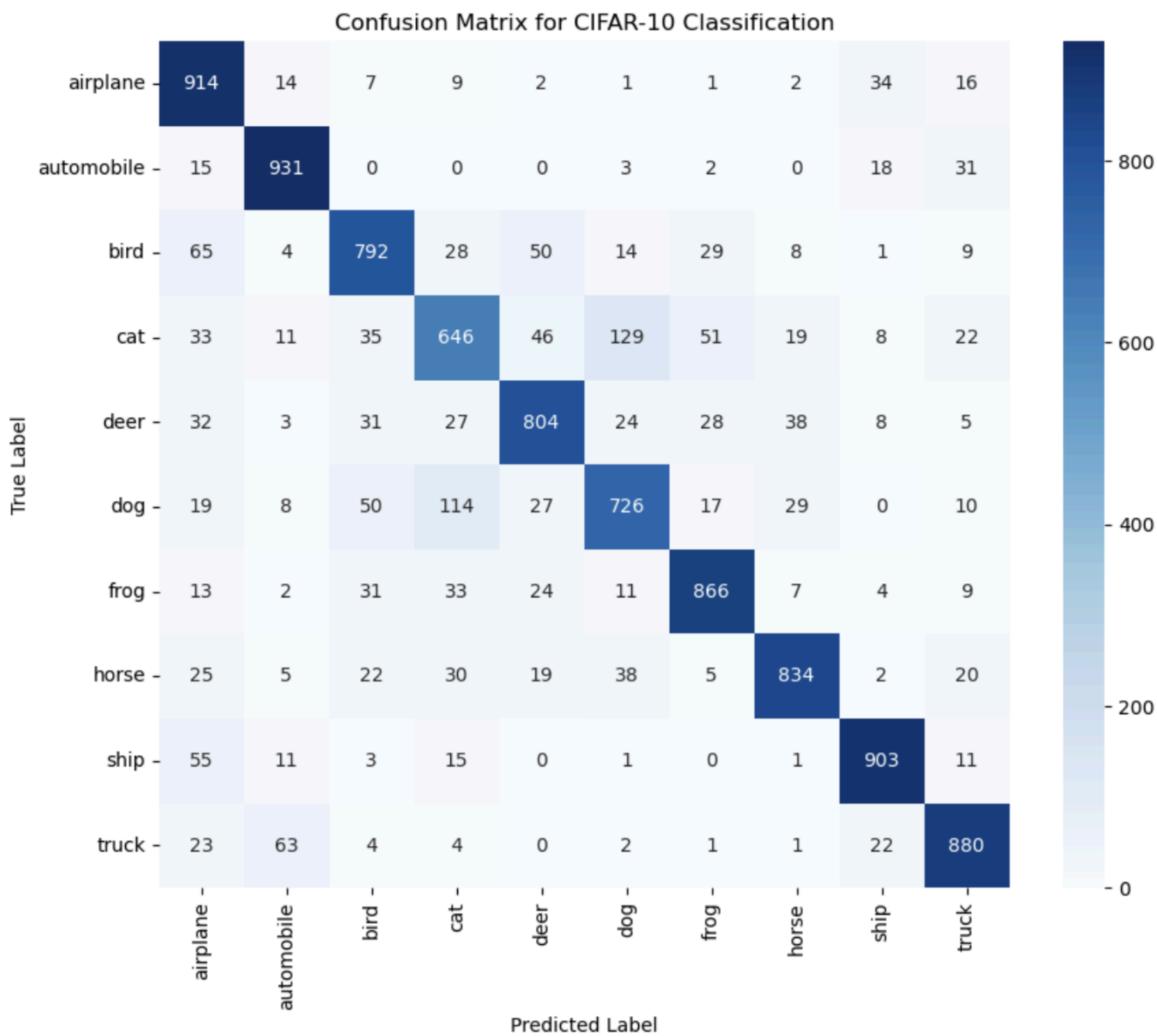
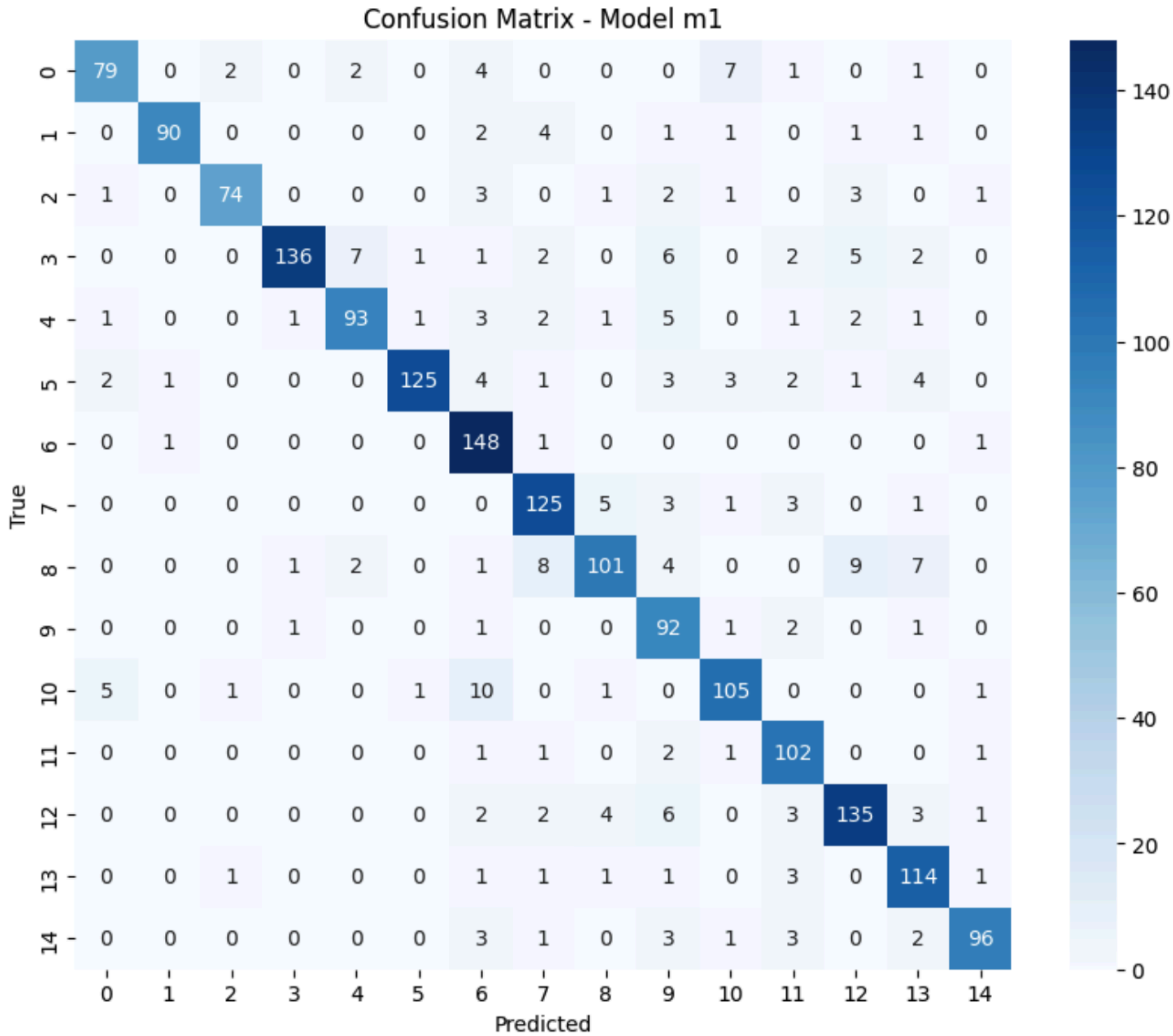


Figure 5: Model 2 Confusion Matrix



Observations:

1. **Model 1 Performance with Adam Optimizer:**
    - Model 1, using the Adam optimizer, achieved moderate success in learning the CIFAR-10 dataset, reaching 75% training accuracy and 73% test accuracy by the end of training. However, there were signs of overfitting, as indicated by fluctuations in testing loss despite a gradual decrease in training loss. The model struggled to generalize well, especially in distinguishing between visually similar classes like "cat" and "dog."
2. **Model 2 Performance with SGD Optimizer and Momentum:**
    - Model 2, with the SGD optimizer and momentum, demonstrated superior performance, achieving 88% training accuracy and 85% test accuracy. The addition of momentum helped the model generalize better by maintaining more stable updates and improving convergence. The confusion matrix for Model 2 showed fewer misclassifications, with better separation between classes,

highlighting that SGD with momentum and average pooling led to a more robust model.
3. **Pooling Layer Impact:**
    - The change from max pooling in Model 1 to average pooling in Model 2 contributed to smoother gradients, likely aiding the model in achieving more accurate classification, especially across visually challenging categories.
4. **Model Export and Inference Optimization:**
    - Exporting both models to the ONNX format enabled faster inference times, particularly on GPU-accelerated hardware. This step is crucial for practical deployment, allowing the models to operate efficiently in real-world applications with high demands on performance.

Conclusion:

This project demonstrated the effectiveness of different model configurations and optimization techniques in multi-class image classification on the CIFAR-10 dataset. **Model 2** outperformed **Model 1** in both training and testing accuracy, showcasing the advantages of using SGD with momentum and average pooling for enhanced generalization and performance. Exporting the models to ONNX format further improved inference speed, making them suitable for deployment on GPU-supported platforms. These findings suggest that model selection, optimizer choice, and pooling strategy can significantly impact model accuracy and generalization in computer vision tasks.

Future Directions:

1. **Experimenting with Deeper Architectures:**
   Implementing more complex architectures such as ResNet or EfficientNet could potentially improve classification accuracy by better capturing the hierarchical features within images.
2. **Hyperparameter Tuning:**
   Fine-tuning parameters such as learning rates, momentum, and regularization could enhance model performance further, helping to prevent overfitting and improve generalization.
3. **Data Augmentation:**
   Applying techniques like random cropping, flipping, and color jittering can increase the diversity of training data, which could improve robustness against overfitting and provide better real-world generalization.

References:

1. Krizhevsky, A., Hinton, G. (2009). *Learning Multiple Layers of Features from Tiny Images*. University of Toronto. (The CIFAR-10 Dataset)

2. Howard, A. G., et al. (2017). *MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications*. arXiv:1704.04861.
3. He, K., et al. (2016). *Deep Residual Learning for Image Recognition*. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 770-778.
4. ONNX. *Open Neural Network Exchange Format*. Available: https://onnx.ai
5. PyTorch Documentation. (2023). *Training a Classifier*. Available: https://pytorch.org/tutorials/beginner/blitz/cifar10_tutorial.html