Vision Transformer, Hybrid CNN-MLP, and ResNet for CIFAR-10 Image Classification

SYED MUHAMMAD MURTAZA KAZMI
Department of Computer Science
National University of Computer & Emerging Sciences
Islamabad, Pakistan
i210685@nu.edu.pk

***Abstract*—This report presents the implementation and evaluation of three deep learning architectures—Vision Transformer (ViT), Hybrid CNN-MLP, and ResNet—for image classification on the CIFAR-10 dataset. The objective is to compare the performance of these models in terms of accuracy, precision, recall, F1-score, training time, memory usage, and inference speed. The Vision Transformer demonstrates superior performance among the three, showcasing the effectiveness of Transformer-based approaches in image classification tasks. The code repository for this project is available at https://github.com/smmk47/CIFAR10-Classification.**

***Index Terms*—Vision Transformer, CNN-MLP Hybrid, ResNet, CIFAR-10, Image Classification, Deep Learning**

I. Introduction

Image classification is a fundamental task in computer vision, with applications ranging from autonomous driving to medical diagnosis. Traditional Convolutional Neural Networks (CNNs) have been the cornerstone of image classification due to their ability to capture spatial hierarchies in data. However, recent advancements have introduced Transformer-based architectures, such as the Vision Transformer (ViT), which leverage self-attention mechanisms to model long-range dependencies in images. This report explores the efficacy of ViT in comparison to a Hybrid CNN-MLP architecture and a pretrained ResNet model using transfer learning on the CIFAR-10 dataset. The comparative analysis focuses on various performance metrics, training efficiency, and deployment considerations to understand the strengths and weaknesses of each architecture in the context of image classification.

II. Methodology

A. Dataset

The CIFAR-10 dataset consists of 60,000 32x32 color images across 10 classes, with 6,000 images per class. The dataset is divided into 50,000 training images and 10,000 test images. The classes include airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck. The dataset is widely used for benchmarking image classification models due to its diversity and manageable size.

B. Data Preprocessing and Augmentation

To enhance model performance and generalization, the following preprocessing and augmentation techniques were applied:

- **Normalization**: Images were normalized using the CIFAR-10 dataset's mean and standard deviation to ensure that the input features have zero mean and unit variance.
- **Random Cropping**: Randomly cropped the images with padding to introduce variance in the training data, helping the model become invariant to small translations.
- **Horizontal Flipping**: Applied random horizontal flips to augment the dataset, increasing the diversity of training samples.

The training dataset was further split into training and validation subsets in an 80:20 ratio to monitor the model's performance during training and prevent overfitting.

C. Model Architectures

Three distinct architectures were implemented and evaluated:

1) Vision Transformer (ViT): The Vision Transformer model was implemented from scratch using PyTorch. The architecture involves:

- **Patch Embedding**: Images are divided into non-overlapping patches, each flattened and projected into an embedding space using a convolutional layer.
- **Positional Encoding**: Positional information is added to the patch embeddings to retain spatial context.
- **Transformer Encoder**: Comprises multiple layers of multi-head self-attention and feed-forward neural networks to capture complex relationships between patches.
- **Classification Head**: A Multi-Layer Perceptron (MLP) is appended to the [CLS] token for final class predictions.

2) Hybrid CNN-MLP: The Hybrid CNN-MLP architecture combines the strengths of CNNs and MLPs:

- **Convolutional Layers**: Extract features from image patches using convolutional operations.
- **Flattening**: The extracted features are flattened to form a feature vector.
- **MLP Classifier**: A series of fully connected layers classify the flattened features into the respective classes.

This architecture leverages CNNs for feature extraction while utilizing MLPs for classification, offering a balance between spatial feature learning and classification capabilities.

3) ResNet with Transfer Learning: A pretrained ResNet18 model was employed, leveraging transfer learning to adapt to the CIFAR-10 dataset:

- **Pretrained Layers**: All layers of the ResNet18 model pretrained on the ImageNet dataset were frozen to retain the learned features.
- **Classifier Modification**: The final fully connected layer was replaced with a new classifier tailored for the CIFAR-10 classes.
- **Fine-Tuning**: Only the parameters of the new classifier layers were trained and fine-tuned on the CIFAR-10 dataset.

Transfer learning enables the model to utilize previously learned representations, reducing training time and improving performance on the target dataset.

D. Training Procedure and Hyperparameter Tuning

Each model was trained using the Adam optimizer with carefully selected learning rates and learning rate schedulers to ensure efficient convergence. Early stopping was implemented based on validation loss to prevent overfitting. Key hyperparameters tuned for each model included:

- **Number of Transformer Layers (ViT)**
- **Number of Attention Heads (ViT)**
- **Learning Rate**
- **Batch Size**
- **Patch Size (ViT and Hybrid CNN-MLP)**

The hyperparameter tuning was guided by validation performance, ensuring optimal settings for each architecture.

III. Results

A. Performance Metrics

The models were evaluated on the following performance metrics:

- **Accuracy**: The proportion of correctly classified images.
- **Precision**: The ability of the classifier not to label a negative sample as positive.
- **Recall**: The ability of the classifier to find all positive samples.
- **F1-Score**: The harmonic mean of precision and recall, providing a balance between the two.

Table I summarizes the performance of each model on these metrics.

B. Confusion Matrices

Confusion matrices provide a detailed breakdown of correct and incorrect classifications across different classes. Figures 1, 2, and 3 display the confusion matrices for the Vision Transformer, Hybrid CNN-MLP, and ResNet models, respectively. These matrices help in understanding the specific classes where each model excels or struggles.

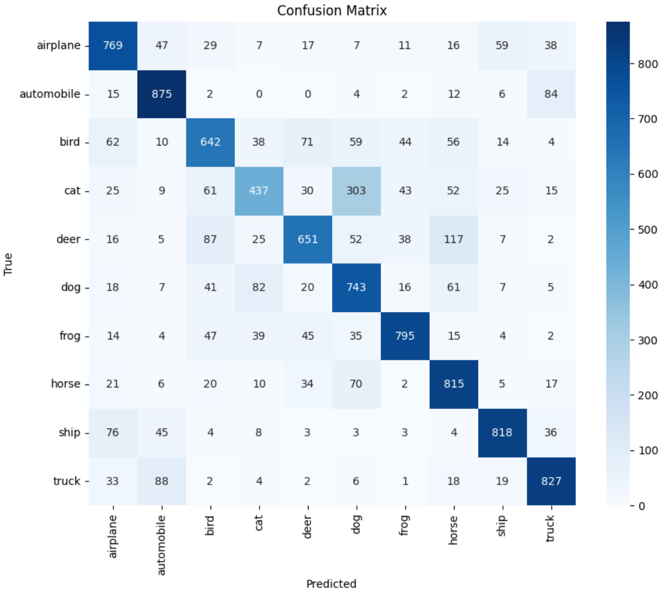

Fig. 1. Confusion Matrix for Vision Transformer

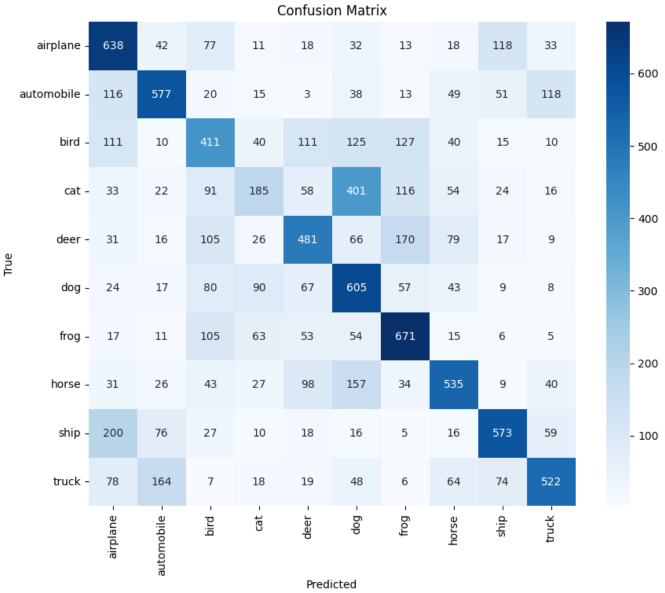

Fig. 2. Confusion Matrix for Hybrid CNN-MLP

C. Training and Validation Loss and Accuracy

Training and validation loss and accuracy curves offer insights into the learning dynamics of each model. Figures 4, 5, and 6 depict these curves for the Vision Transformer, Hybrid CNN-MLP, and ResNet models, respectively. These plots aid in identifying issues like overfitting, underfitting, and convergence rates.

D. Examples of Completed Predictions

To qualitatively assess the models' performance, several examples of model predictions on test images are provided. Figures 7, 8, and 9 show instances where the models correctly

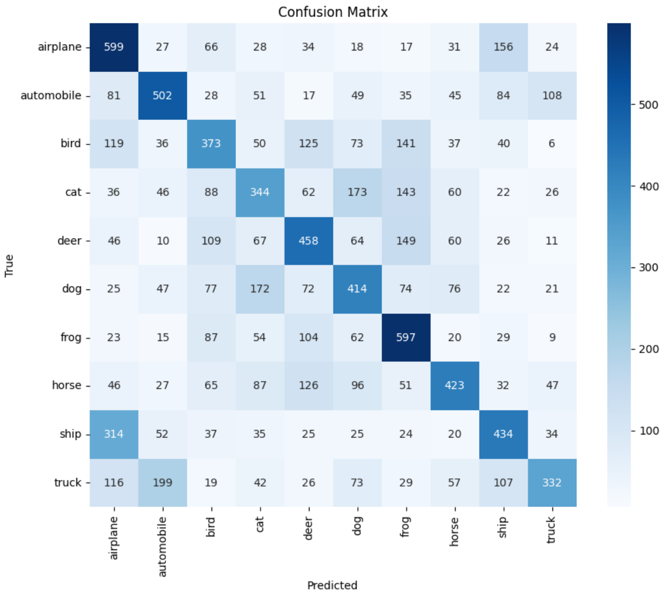

Fig. 3. Confusion Matrix for ResNet

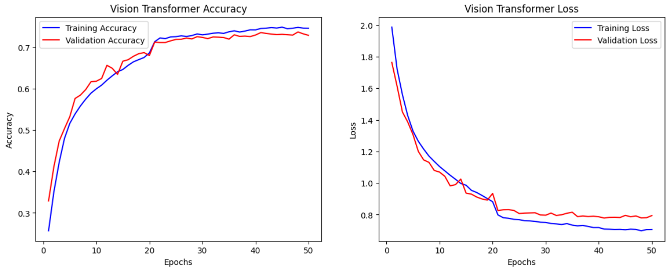

Fig. 4. Training and Validation Curves for Vision Transformer

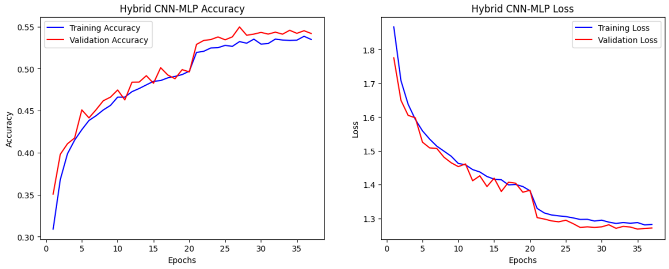

Fig. 5. Training and Validation Curves for Hybrid CNN-MLP

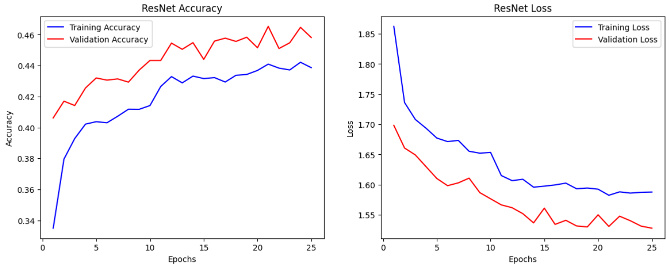

Fig. 6. Training and Validation Curves for ResNet

and incorrectly classified images. These examples highlight the models' strengths and areas for improvement in classifying specific categories.

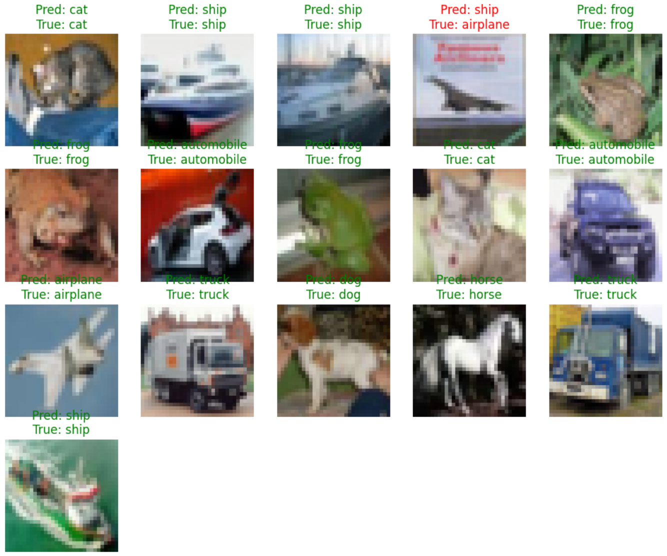

Fig. 7. Vision Transformer Predictions on Test Images

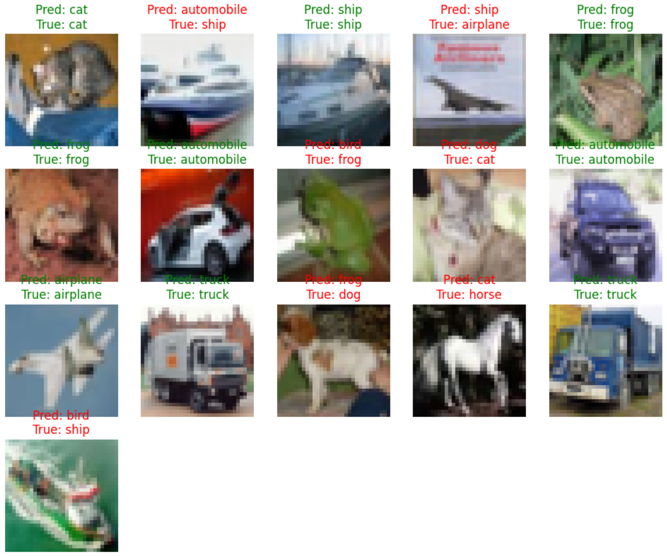

Fig. 8. Hybrid CNN-MLP Predictions on Test Images

E. Performance Comparison

The comparative analysis of the models is presented in Table I and Figures 10 to 13. The Vision Transformer outperformed the Hybrid CNN-MLP and ResNet models across all metrics, indicating its superior capability in capturing complex image patterns and dependencies.

IV. Discussion

The Vision Transformer (ViT) outperformed both the Hybrid CNN-MLP and ResNet models across all evaluated met-

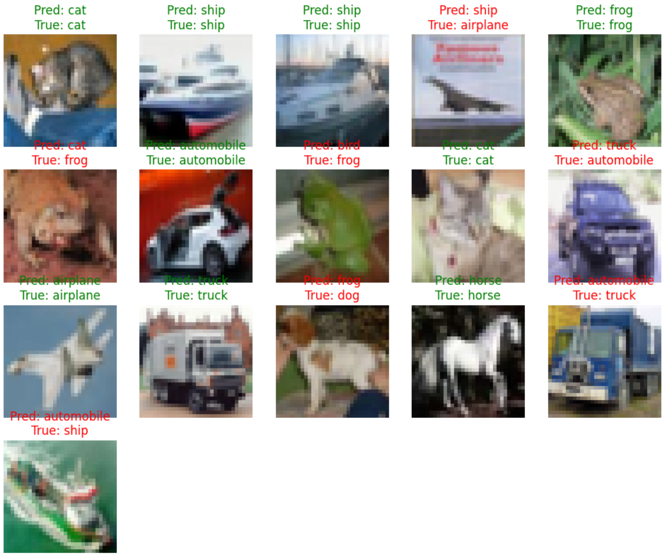

Fig. 9. ResNet Predictions on Test Images

TABLE I
PERFORMANCE COMPARISON OF MODELS

Model	Accuracy	Precision	Recall	F1-Score	Trai
Vision Transformer	73.72%	73.99%	73.72%	73.39%	
Hybrid CNN-MLP	51.98%	52.34%	51.98%	51.35%	
ResNet	44.76%	45.21%	44.76%	44.47%	

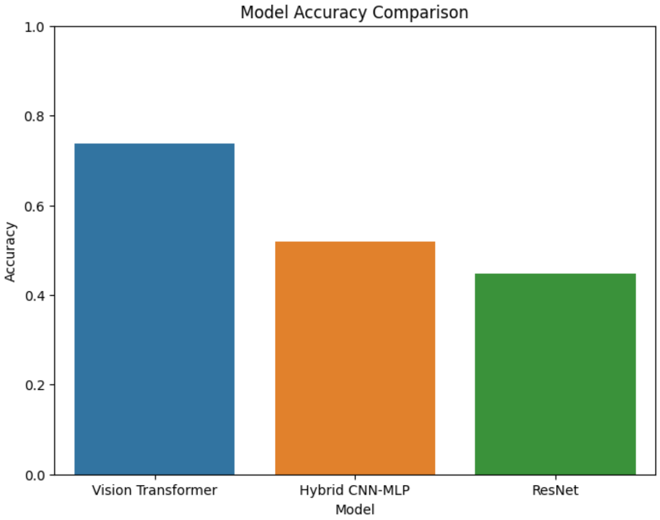

Fig. 10. Model Accuracy Comparison

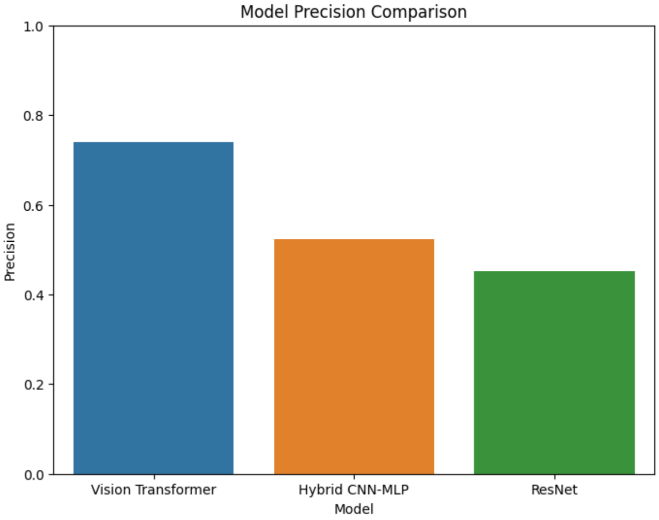

Fig. 11. Model Precision Comparison

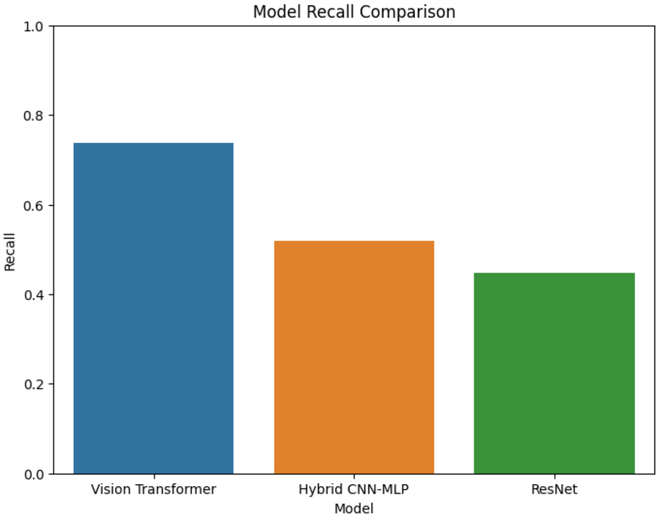

Fig. 12. Model Recall Comparison

rics, achieving an accuracy of 73.72%. This superior performance underscores the effectiveness of Transformer-based architectures in capturing complex and long-range dependencies within image data. The self-attention mechanism in ViT allows the model to focus on relevant parts of the image, leading to more accurate classifications.

The Hybrid CNN-MLP model demonstrated moderate performance with an accuracy of 51.98%. While the combination of CNNs for feature extraction and MLPs for classification is a viable approach, it lacks the depth and attention mechanisms that ViT employs, limiting its ability to capture intricate patterns in the data.

The ResNet model, utilizing transfer learning, achieved the lowest accuracy of 44.76%. This underperformance could be attributed to several factors:

- **Dataset Specificity**: ResNet18 is pretrained on ImageNet,

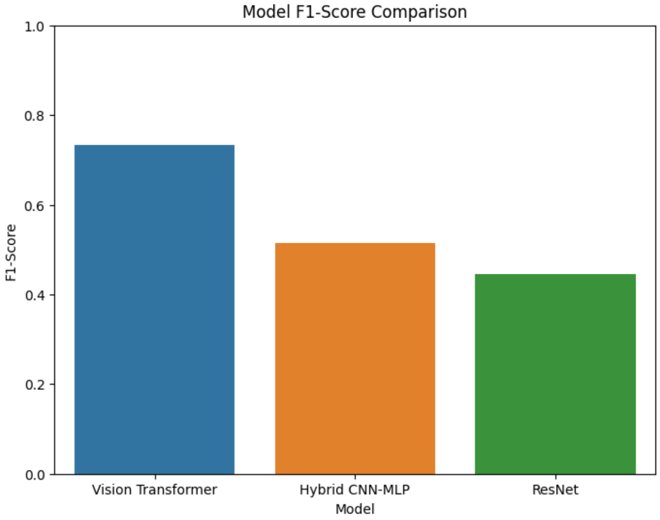

Fig. 13. Model F1-Score Comparison

which has different image characteristics compared to CIFAR-10. The limited fine-tuning might not have been sufficient to adapt the model to the new dataset.

- **Model Complexity**: ResNet18 might be too complex for the relatively small CIFAR-10 images, leading to overfitting or inefficient learning.
- **Training Parameters**: The hyperparameters, such as learning rate and batch size, might not have been optimal for fine-tuning the pretrained layers.

A. *Analysis of Model Performance Over Time*

The training and validation curves (Figures 4, 5, 6) provide insights into how each model's predictions improved over time. The Vision Transformer showed consistent improvement in both training and validation accuracy with a steady decrease in loss, indicating effective learning and generalization. In contrast, the Hybrid CNN-MLP and ResNet models exhibited slower convergence rates and signs of overfitting, where the validation performance plateaued or even degraded despite continued improvements in training accuracy.

B. *Challenges Encountered*

Several challenges were encountered during the project:

- **Hyperparameter Optimization**: Finding the optimal set of hyperparameters for each model was time-consuming and required extensive experimentation.
- **Resource Constraints**: Training deep models like ViT demanded significant computational resources, especially with limited GPU memory.
- **Model Complexity**: Balancing model complexity with the capacity to generalize on the CIFAR-10 dataset was challenging, particularly for the ResNet model.
- **Data Augmentation Limitations**: While data augmentation techniques improved generalization, they were insufficient to fully mitigate overfitting in some models.

Despite these challenges, the project provided valuable insights into the strengths and limitations of different deep learning architectures for image classification.

V. Conclusion

This study demonstrated the comparative performance of Vision Transformer, Hybrid CNN-MLP, and ResNet models on the CIFAR-10 image classification task. The Vision Transformer emerged as the most effective model, highlighting the potential of Transformer-based approaches in computer vision. The Hybrid CNN-MLP and ResNet models, while less accurate, offer insights into the strengths and limitations of traditional and hybrid architectures. Future work may explore more sophisticated hybrid models, deeper fine-tuning of pretrained models, and the integration of additional data augmentation techniques to further enhance performance.

VI. Prompts

The following prompts were utilized during the development and analysis of the models:

- **Data Preprocessing and Augmentation:**
  - ”Apply random cropping with padding to the CIFAR-10 training images.”
  - ”Implement normalization using CIFAR-10 mean and standard deviation.”
  - ”Apply random horizontal flipping as a data augmentation technique.”
- **Model Implementation:**
  - ”Divide images into non-overlapping patches and project them into an embedding space for Vision Transformer.”
  - ”Construct a Hybrid CNN-MLP architecture by combining convolutional layers with a multi-layer perceptron.”
  - ”Utilize a pretrained ResNet18 model and modify the final fully connected layer for CIFAR-10 classification.”
- **Training Procedure:**
  - ”Train each model using the Adam optimizer with a learning rate of 0.001.”
  - ”Implement learning rate scheduling with step size 20 and gamma 0.1.”
  - ”Apply early stopping with a patience of 10 epochs to prevent overfitting.”
- **Evaluation:**
  - ”Evaluate models using accuracy, precision, recall, and F1-score metrics.”
  - ”Generate and visualize confusion matrices for each model.”
  - ”Plot training and validation loss and accuracy curves to assess model convergence.”

VII. References

References

[1] Touvron, H., Cord, M., Douze, M., Massa, F., Synnaeve, G., Usunier, N., ... & Joulin, A. (2021). Training data-efficient image transformers & distillation through attention. *arXiv preprint arXiv:2012.12877*.

[2] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).

[3] Krizhevsky, A., & Hinton, G. (2009). Learning multiple layers of features from tiny images. *Technical Report*.

[4] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., ... & Houlsby, N. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

[5] Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.

[6] SMMK47. (2024). *CIFAR10-Classification*. GitHub repository. Available at: https://github.com/smmk47/CIFAR10-Classification.