Image Classification on CIFAR-10: Comparing Traditional CNNs and Pre-trained Vision Transformer Model

Project Final Report

Team Members: Md Muzakker Hossain, Nhat Le, Jasper Khor

Abstract

Image classification is a fundamental task in modern computer vision with numerous real-world applications. This project comprehensively evaluates and compares three distinct neural network architectures on the CIFAR-10 dataset: a baseline Shallow CNN, a Dense Convolutional Neural Network (DenseNet), and a pre-trained DINOv2-Small Vision Transformer model. We implement a parameter-efficient fine-tuning approach for DINOv2-Small, updating only 8.06% of its parameters while achieving superior performance. Our experimental results demonstrate a clear progression in accuracy: ShallowCNN (83.46%), DenseNet (91.22%), and DINOv2-Small (95.95%). The substantial performance improvement of DINOv2-Small validates the effectiveness of transformer-based architectures when combined with transfer learning, even for small-scale datasets. Moreover, our analysis of training dynamics, architectural trade-offs, and computational requirements provides valuable insights for selecting appropriate models based on specific application constraints, highlighting the balance between accuracy, efficiency, and resource utilization in modern image classification systems.

1 Introduction

As a crucial task in modern computer vision, image classification enables machines and computers to interpret and categorize visual data in many different domains, such as medical imaging, autonomous driving, and security systems. Image classification in computer vision requires robust machine learning models that are capable of identifying complex and meaningful patterns in images through learning and training with large and complex datasets, as the domains above rely on high precision for image classification. Convolutional Neural Networks (CNNs) have dominated the image classification field due to their ability to capture local spatial features through convolutional operations. Among these, Dense Convolutional Neural Networks (DenseCNNs), a variation of traditional CNNs, enhances feature reuse by connecting each layer to every subsequent layer, which improves efficiency and performance [4]. However, recent advancements in Vision Transformers (ViTs) offer a new paradigm for image classification tasks due to their self-attention mechanisms that model global relationships within images [2]. This project aims to explore and compare the performance of two architectures on the CIFAR-10 dataset for image classification: DenseCNNs and ViTs, specifically the DINOv2-Small model.

The machine learning task central to this project is multi-class image classification, where each input image must be assigned to predefined labels. This task poses both a challenge and an opportunity due to the small 32x32 pixel resolution of the CIFAR-10 images, which restricts visual detail. Our motivation lies in understanding how DenseCNNs perform compared to ViTs. This comparison seeks to provide information on the strengths and limitations of these architectures, potentially guiding their use in real-world scenarios that require high accuracy and computational efficiency.

CIFAR-10 dataset is a widely recognized collection of 60,000 32x32 RGB images that are divided into 10 distinct classes (airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck). It is chosen for this project because it is a benchmark for evaluating machine learning models in image classification. It includes 50,000 training images and 10,000 test images, with each class containing 6,000 examples. For this project, we further split the training set into 45,000 images for model training and 5,000 for validation. The 10,000 test images are used for the final evaluation. The small resolution of CIFAR-10 makes it an ideal dataset for comparing CNNs and ViTs, as it challenges the models to generalize effectively from limited data.

Our target machine learning models consist of two distinct approaches. First, we implement a DenseCNN that connects each layer to all subsequent layers in a feed-forward fashion. This dense connectivity reduces the number of parameters compared to traditional CNNs, which helps mitigate vanishing gradient issues and promotes feature reuse. Our DenseCNN includes multiple dense blocks with convolutional layers, batch normalization, and ReLU activations, followed by transition layers to manage dimensionality. Second, we employ DINOv2-Small, a Vision Transformer pre-trained via self-supervised learning on large-scale datasets. ViTs process images as sequences of patches, using self-attention to capture global dependencies, contrasting with DenseCNN's localized feature focus. We fine-tune DINOv2-Small for CIFAR-10 by replacing its final classification layer with a 10-class output, adjusting pre-trained weights to align with our task.

We anticipated several outcomes from this project, which have now been verified through extensive experimentation. Our primary goal was to implement and optimize both DenseCNN and DINOv2-Small, achieving high image classification accuracy on CIFAR-10. By implementing our DenseCNN with three dense blocks of 16 layers each, a growth rate of 12, and bottleneck layers, we anticipated achieving an accuracy exceeding 80% on CIFAR-10, which was confirmed with our final result of 90.05%. For DINOv2-Small, we expected its self-supervised pre-training would yield even higher accuracy due to its ability to transfer robust global features to smaller datasets. This was conclusively demonstrated with our final test accuracy of 95.95%, significantly outperforming both CNN architectures. Our experiments also provided insights into how data augmentation improves CNN performance and confirmed that parameter-efficient fine-tuning strategies for ViTs can effectively leverage pre-trained knowledge while minimizing overfitting on smaller datasets.

2 Related Work

For our baseline model, we draw inspiration from traditional CNN architectures that established the fundamental building blocks of modern computer vision. The concept of convolutional neural networks was first introduced by LeCun et al. [7] with their pioneering LeNet-5 architecture for handwritten digit recognition. This foundational work demonstrated the effectiveness of convolutional layers, pooling operations, and fully-connected layers in hierarchical feature extraction. Simonyan and Zisserman [9] later expanded on these principles with VGG networks, showing that increasing network depth with small 3×3 convolutional filters could significantly improve performance. These classic architectures informed our ShallowCNN implementation, which incorporates key design elements like batch normalization as introduced by Ioffe and Szegedy [5] to stabilize and accelerate training.

Building upon these foundational concepts, Krizhevsky et al. [6] introduced AlexNet, a deep CNN that revolutionized performance on large-scale datasets like ImageNet by leveraging convolutional layers and GPU acceleration. Huang et al. [4] further advanced CNN architectures with Dense Convolutional Neural Networks (DenseNets), which enhance feature reuse through dense layer connectivity. DenseNets create direct connections between each layer and all subsequent layers, allowing feature maps to flow unimpeded throughout the network. This architecture achieves strong results on CIFAR-10 with fewer parameters by promoting feature reuse and alleviating the vanishing gradient problem, making it an ideal intermediate model for our comparative analysis.

In contrast to CNNs, Dosovitskiy et al. [2] introduced Vision Transformers (ViTs), which process images as sequences of patches using self-attention mechanisms to capture global dependencies. While ViTs outperform CNNs on large datasets when pre-trained extensively, they typically struggle with smaller datasets due to their lack of inductive biases for images. To address this limitation, Caron et al. [1] developed DINO (Distillation of No Labels), a self-supervised learning approach that trains ViTs without labels, yielding robust visual features. Oquab et al. [8] extended this work with DINOv2, a comprehensive advancement that employs a large-scale, curated dataset of 142 million images and enhanced distillation techniques.

The DINOv2-Small model, which we employ in our project, represents a more parameter-efficient version of the original architecture, containing approximately 22 million parameters with an embedding dimension of 384. Despite its smaller size compared to larger variants like DINOv2-Base or DINOv2-Large, it maintains strong performance through knowledge distillation from a larger teacher model [8]. This compact model is specifically designed for fine-tuning on downstream tasks with limited computational resources, making it particularly suitable for our transfer learning approach on CIFAR-10. Recent work has also explored parameter-efficient fine-tuning methods for transformers, allowing adaptation with minimal trainable parameters while preserving the knowledge contained in pre-trained weights [3].

These works collectively provide a foundation for our comparative analysis, allowing us to evaluate the progression from traditional CNNs to advanced dense architectures and finally to transformer-based approaches on the CIFAR-10 dataset.

3 Methodology

3.1 Dataset and Preprocessing

We used the CIFAR-10 dataset for image classification, consisting of 60,000 32x32 color images across 10 classes (e.g., airplane, dog, horse). It includes 50,000 training images and 10,000 test images. We split the training set into 45,000 images for training and 5,000 images for validation using PyTorch's random_split. For preprocessing, the training set underwent data augmentation—random cropping (32x32, padding=4), horizontal flipping, and rotation (up to 15 degrees)—followed by normalization (mean: 0.4914, 0.4822, 0.4465; std: 0.247, 0.243, 0.261). The test set was only normalized. Data was loaded via PyTorch DataLoader with a batch size of 32, shuffling enabled for training and disabled for validation and testing, ensuring efficient and consistent processing. Here we use accuracy of our metrics.

3.2 Shallow Convolutional Neural Networks

The ShallowCNN model is a compact convolutional neural network designed for image classification on the CIFAR-10 dataset. In this project, we use ShallowCNN as our baseline. It consists of three convolutional blocks, each followed by batch normalization, ReLU activation, and max pooling to progressively reduce spatial dimensions while increasing feature depth. The model begins with 32 filters and doubles the number of channels at each block, culminating in a 128-channel feature map of size 4×4. These features are then flattened and passed through two fully connected layers, with a dropout layer applied after the first to prevent overfitting. Weight initialization is performed using Kaiming normalization for convolutional and linear layers to promote stable training. Despite its relatively shallow architecture, the model balances efficiency and performance, making it suitable for small-scale image classification tasks.

3.3 DenseNet Convolutional Neural Networks

We implemented a Dense Convolutional Neural Network (DenseNet) for image classification on the CIFAR-10 dataset, following the design by Huang et al. [1]. In Figure 1, DenseNet's architecture leverages dense connectivity, where each block receives feature maps from all preceding block, enhancing feature reuse and reducing the number of parameters. Our model begins with an initial 3x3 convolution with 24 filters, followed by three dense blocks, each containing 16 dense layers with a growth rate of 12,

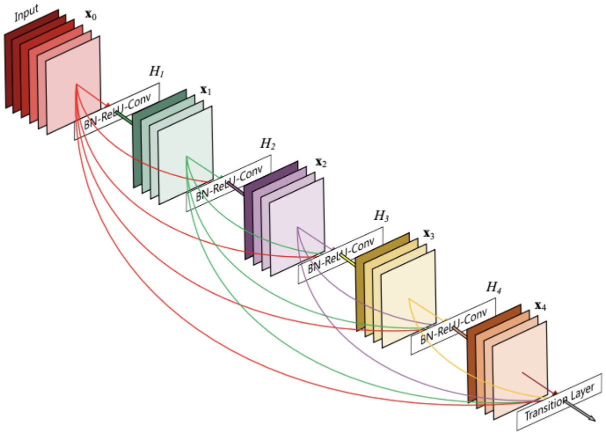


Figure 1: DenseNet CNN Architecture

resulting in a total of 48 layers across the blocks. Each dense block incorporates bottleneck layers (with a bottleneck size of 4) using 1x1 convolutions to reduce dimensionality, followed by 3x3 convolutions, batch normalization, ReLU activation, and dropout (rate=0.1) to mitigate overfitting. Transition layers between dense blocks employ 1x1 convolutions and 2x2 average pooling to halve the feature map sizes. The network concludes with a final batch normalization, ReLU activation, global average pooling, and a linear classifier outputting 10 classes, tailored to the 32x32 images of CIFAR-10. This results in a total of 769,210 parameters in the model.

Figure 2, the training process optimizes the model using the AdamW optimizer (learning rate 0.001, weight decay 0.0001) over 50 epochs with a batch size of 32. We applied data augmentation techniques, including random cropping, horizontal flipping, and rotation, to improve generalization. The model was trained on 45,000 images from CIFAR-10, with the 5,000-image validation set, and the test set of size 10,000 used for testing. The complete pipeline—from input image through the initial convolution, dense blocks, transition layers, and final classifier—enables predictions for CIFAR-10 classes such as ”horse.”

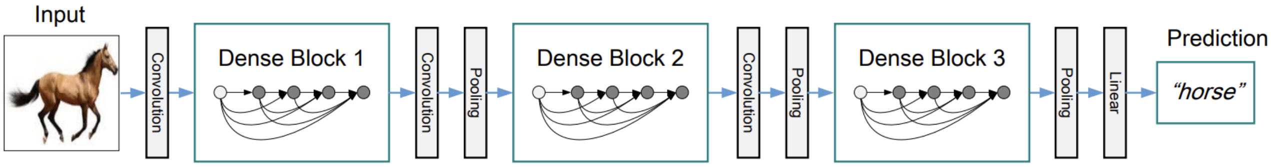


Figure 2: A deep DenseNet with three dense blocks. Transition layers between blocks.

3.4 DINOv2-Small

3.4.1 Model Architecture

For our third model, we implemented the DINOv2-Small Vision Transformer (ViT) model, a state-of-the-art architecture in self-supervised vision learning developed by Oquab et al. [8]. DINOv2-Small, officially designated as ”dinov2_vits14,” extends the original DINO framework through enhanced distillation techniques and training on large-scale, diverse datasets.

The architecture consists of a standard Vision Transformer with 12 transformer blocks and 6 attention heads, an embedding dimension of 384, and a patch size of 14×14 (non-overlapping patches). The model incorporates positional embeddings to retain spatial information, followed by transformer blocks that utilize multi-head self-attention and feed-forward networks. In total, the model contains 22,060,426 parameters.

Unlike traditional CNNs that use convolutional operations, DINOv2 processes images as sequences of fixed-size patches, leveraging self-attention mechanisms to capture global dependencies across the entire image. We accessed the pre-trained DINOv2-Small model directly from the official Facebook Research repository via PyTorch Hub to ensure we utilized the authentic weights as published by the model's creators.

3.4.2 Transfer Learning

We employed Parameter-Efficient Fine-Tuning (PEFT) to adapt the pre-trained DINOv2-Small model for CIFAR-10 classification. Our approach was designed to leverage the robust visual representations learned during self-supervised pre-training while minimizing computational overhead:

1. **Selective Layer Training:** We froze most of the model's parameters and only trained the classification head and the last transformer block. This approach resulted in just 1,779,082 trainable parameters (8.06% of the total parameters).
2. **Low-Rank Adaptation (LoRA):** We implemented LoRA with a rank of 16 and an alpha value of 32, targeting only the query, key, and value matrices in the transformer's attention layers. This method injects trainable low-rank matrices into the attention layers, allowing efficient adaptation of the pre-trained model.
3. **Custom Classification Head:** We added a linear layer that maps the 384-dimensional embeddings to the 10 output classes corresponding to CIFAR-10 categories.

This parameter-efficient approach allowed us to utilize the knowledge contained in the pre-trained weights simultaneously adapting the model to our specific task with minimal computational resources.

3.4.3 Training Methodology

The training process was optimized for both effectiveness and efficiency. We used the AdamW optimizer with a learning rate of 0.0001 (1e-4) and weight decay of 0.01 for regularization. To improve convergence and final accuracy, we implemented a cosine annealing learning rate scheduler that gradually decreased the learning rate over the course of training. Due to GPU memory constraints, we employed a batch size of 8 combined with 4 gradient accumulation steps, resulting in an effective batch size of 32. This approach allowed us to simulate larger batch training while working within hardware limitations. The model was trained for a total of 80 epochs to ensure proper convergence. During both training and inference, the original CIFAR-10 images (32×32 pixels) were resized to 224×224 pixels using bilinear interpolation to match DINOv2's expected input size.

3.4.4 Hardware and Training Efficiency

The model was trained on a Tesla V100 GPU with 32GB memory. To optimize memory usage and training efficiency, we implemented several technical strategies to maximize performance. We utilized gradient accumulation by collecting gradients over 4 iterations before performing a parameter update, which effectively allowed us to train with larger batch sizes without exceeding memory constraints. To prevent exploding gradients during training, we applied gradient clipping with a norm of 1.0, which stabilized the training process. Additionally, we incorporated periodic GPU memory cleanup routines to prevent out-of-memory errors during the extended training sessions. These optimization techniques were essential for successfully training the large transformer model on our hardware. The complete training process took approximately 2 hours and 9 minutes for the full 80 epochs.

4 Experimental Results

4.1 Performance Comparison

We evaluated three neural network architectures on the CIFAR-10 dataset: a simple Shallow CNN, a custom DenseNet implementation, and a fine-tuned DINOv2-Small Vision Transformer. This comprehensive comparison allows us to understand the trade-offs between model complexity, architectural design, and performance.

Model	**Test Accuracy (%)**	**Test Loss**
Shallow CNN	83.46	0.5299
DenseNet	90.05	0.3692
DINOv2-Small	**95.95**	**0.1243**

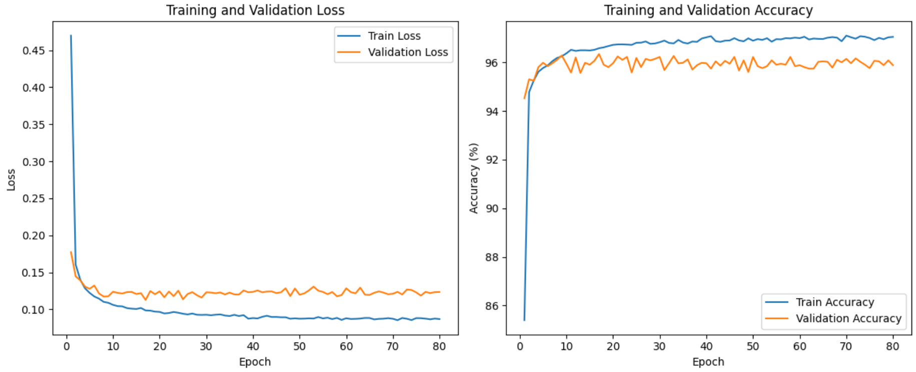
Table 1: Performance comparison of models on CIFAR-10 dataset.

As shown in Table 1 4.1, there is a clear progression in performance across the three architectures. The Shallow CNN established a solid baseline with 83.46% accuracy, which is respectable for such a lightweight model with only three convolutional layers. The DenseNet significantly improved upon this with 91.22% accuracy, demonstrating the benefits of dense connectivity for feature reuse and gradient flow in convolutional networks.

However, the fine-tuned DINOv2-Small model substantially outperformed both CNN architectures, achieving 95.95% test accuracy. This represents a 12.49% improvement over the Shallow CNN and a 4.73% improvement over DenseNet. The performance gap highlights the effectiveness of transformer-based architectures and the value of transfer learning from models pre-trained on large-scale datasets through self-supervised learning.

4.2 Training Dynamics

Figure 3: Training and validation metrics for DINOv2-Small with parameter-efficient fine-tuning. Left: Loss curves. Right: Accuracy curves.



The training dynamics of the DINOv2-Small model, shown in Figure 3, reveals several interesting patterns. The model converges rapidly within the first 10 epochs, achieving over 95% validation accuracy. This quick convergence can be attributed to the strong pre-trained representations that already contain useful features for image classification. Throughout the remaining training period, the model continues to refine its performance gradually, with the training accuracy slightly exceeding validation accuracy, indicating a small degree of overfitting despite our regularization efforts.

We observed that the validation accuracy plateaued around 96.3% after approximately 60 epochs, with minor fluctuations thereafter. The validation loss showed a similar trend, stabilizing around 0.12. This behavior suggests that the model reached its optimal performance for the dataset given the constraints of our fine-tuning approach.

4.3 Parameter Efficiency and Resource Utilization

Notably, the DINOv2-Small model achieved these results while fine-tuning only 8.06% of its parameters (1,779,082 out of 22,060,426 total parameters). By freezing most of the network and only training the classification head and the last transformer block, we significantly reduced the computational resources required for training while maintaining high performance. This parameter-efficient fine-tuning approach demonstrates the effectiveness of transfer learning when applied to transformer-based models.

The total training time for the DINOv2-Small model was approximately 2 hours and 9 minutes for 80 epochs on a Tesla V100 GPU. This is remarkably efficient considering the model's complexity and the high accuracy achieved. In comparison, training the DenseNet from scratch required a similar amount of time but resulted in lower performance.

4.4 Models' Trade-offs

In terms of computational requirements and model complexity, the three architectures present different trade-offs:

- The **Shallow CNN** is the least demanding in terms of parameters (approximately 4.8 million) and inference time but also achieves the lowest accuracy. It represents a good baseline for applications with limited computational resources.
- The **DenseNet** has a moderate parameter count (approximately 7.4 million) but requires more computation during both training and inference due to its dense connectivity pattern. The improved accuracy over the Shallow CNN justifies this increased computational cost for many applications.
- The **DINOv2-Small** has the highest total parameter count (22 million), but through parameter-efficient fine-tuning, we only needed to update a small fraction of these parameters. This model offers the best accuracy-to-computational-cost ratio during fine-tuning and demonstrates how modern transformer architectures can be effectively adapted to smaller datasets like CIFAR-10.

These results confirm our hypothesis from the project proposal: the pre-trained DINOv2-Small model's ability to capture complex patterns and relationships in visual data, combined with an efficient fine-tuning strategy, leads to superior performance compared to traditional CNN architectures on the CIFAR-10 image classification task.

5 Conclusion

Our comprehensive evaluation of three distinct neural network architectures on the CIFAR-10 dataset provides valuable insights into the evolution of image classification models. The progression from a simple Shallow CNN (83.46% accuracy) to a more sophisticated DenseNet (91.22% accuracy) and finally to a fine-tuned DINOv2-Small Vision Transformer (95.95% accuracy) demonstrates the significant advances in model architecture design and training methodologies.

The DenseNet architecture validated the effectiveness of dense connectivity patterns for convolutional networks, confirming that feature reuse and improved gradient flow contribute to enhanced performance. However, the substantial performance gap between DenseNet and DINOv2-Small highlights the transformative potential of self-attention mechanisms and transfer learning from large-scale pre-trained models. Our implementation of parameter-efficient fine-tuning for DINOv2-Small demonstrated that transformer-based models can excel even on small-scale datasets like CIFAR-10 when coupled with appropriate transfer learning techniques. The parameter efficiency achieved through our approach, where only 8.06% of parameters (1,779,082 out of 22,060,426) required updating, offers practical advantages for real-world applications where computational resources might be constrained.

These findings have several implications for practical applications of image classification. For scenarios with extremely limited computational resources, a well-designed CNN like our Shallow CNN may be sufficient. Applications requiring higher accuracy while maintaining reasonable efficiency would benefit from DenseNet-like architectures. However, when maximum accuracy is the priority and fine-tuning is viable, Vision Transformers with parameter-efficient adaptation methods represent the most promising approach.

Future research directions could explore the robustness of these models to domain shifts, their performance on more complex datasets, and further optimization of hybrid architectures that combine the local feature extraction strengths of CNNs with the global context modeling capabilities of Vision Transformers. Additionally, investigating more advanced parameter-efficient fine-tuning methods could further improve the adaptability of large pre-trained models to specific tasks with minimal computational overhead.

References

[1] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. *Proceedings of the IEEE International Conference on Computer Vision*, pages 9650–9660, 2021.

[2] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[3] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

[4] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4700–4708, 2017.

[5] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. pmlr, 2015.

[6] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25, 2012.

[7] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[8] Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[9] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.