Deep Learning and Applications

(UEC642)

Deep Learning Project Report: CIFAR-10 Image Classification

Submitted by:

Siddharth jindal (102215019)

Nishant mittal (102215310)

Electronics and Communication Engineering Department

Thapar Institute of Engineering and Technology, Patiala

December 2025

Deep Learning Project Report: CIFAR-10 Image Classification

Baseline CNN vs ResNet-18 (Transfer Learning)

---

1. Abstract

Image classification remains a fundamental task in computer vision, with deep learning approaches consistently demonstrating state-of-the-art performance across a wide range of datasets and application domains. This project investigates image classification on the CIFAR-10 dataset using two contrasting deep learning architectures: a custom Baseline Convolutional Neural Network (CNN) trained from scratch, and a transfer-learning approach based on a pre-trained ResNet-18 model.

The objective of this work was to design and implement a complete end-to-end pipeline for data loading, preprocessing, model training, and quantitative evaluation, enabling a fair performance comparison between both methods. The models were trained for five epochs using the Adam optimizer with Cross-Entropy loss, and were evaluated using accuracy, precision, recall, F1-score, and confusion matrices.

Experimental results showed that the Baseline CNN achieved a test accuracy of **75.8%**, demonstrating effective feature learning and generalization despite its shallow architecture and limited training time. In contrast, the ResNet-18 model achieved only **40.28% accuracy**, indicating ineffective transfer of pre-trained features under the applied constraints. The poor performance of ResNet-18 can be attributed to factors such as freezing of the feature extractor layers, mismatch between input resolution and the model's original training domain, and insufficient fine-tuning.

This study highlights that transfer learning does not guarantee superior performance when model adaptation is limited or when data characteristics differ significantly from the pretraining dataset. Recommendations for improving transfer learning performance on smallscale datasets, including unfreezing layers, input resizing, and longer training schedules, are discussed.

---

2. Introduction

Deep learning has revolutionized the field of computer vision by enabling models to learn hierarchical, task-specific representations directly from raw data. Convolutional Neural Networks (CNNs), in particular, have proven to be highly effective for image classification tasks, achieving strong performance on benchmark datasets and enabling applications ranging from medical imaging to autonomous driving.

The CIFAR-10 dataset is a widely used benchmark for assessing image classification performance on small, low-resolution images. It contains 60,000 labeled images belonging to 10 object categories, covering both natural and artificial classes. Due to its relatively modest size, limited spatial resolution, and high inter-class similarity, CIFAR-10 serves as a challenging yet practical testbed for evaluating model performance and learning capacity under resource constraints.

Modern deep learning architectures such as ResNet, DenseNet, and Vision Transformers (ViT) have demonstrated **90–99% accuracy** on CIFAR-10 when trained extensively with large computational budgets, sophisticated augmentations, and advanced optimization strategies. However, such results may not be achievable in constrained environments, where computational resources and training time are limited.

This project investigates two contrasting deep learning strategies for CIFAR-10 classification:

1. A **custom CNN trained from scratch**, designed to be lightweight, computationally efficient, and optimized specifically for small image inputs.
2. A **pre-trained ResNet-18 model adapted via transfer learning**, representing a widely used approach in scenarios where training data is limited or training from scratch is infeasible.

The primary goals of this project were to:

- Implement both models within a unified pipeline
- Train them under identical settings to ensure fair comparison
- Evaluate performance using standardized quantitative metrics
- Analyze model behavior and identify the advantages and limitations of each approach under constrained training conditions

By comparing results from these two divergent strategies, this study aims to assess whether transfer learning offers meaningful benefits for small-scale classification tasks with limited fine-tuning, and to identify conditions under which simpler architectures may outperform larger pre-trained networks.

3. Methodology

This section describes the dataset, preprocessing pipeline, model architectures, training configuration, and evaluation strategy used in this experiment. The overall objective of the methodology was to develop a reproducible deep learning workflow for image classification on CIFAR-10, enabling fair comparison between a simple CNN trained from scratch and a pre-trained ResNet-18 model adapted via transfer learning.

3.1 Dataset

The CIFAR-10 dataset is one of the most widely used benchmarks in computer vision research, specifically for evaluating image classification algorithms on small-scale images. It consists of **60,000 RGB images**, each with a resolution of **32×32 pixels**, belonging to **10 mutually exclusive classes**.

The dataset is split into:

- **50,000 training images**, used for model optimization
- **10,000 testing images**, used for performance evaluation

The classes represent common object categories, including **airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck**, covering both natural and man-made classes with high intra-class variability.

CIFAR-10 presents several challenges that make it a good benchmark for deep learning research:

- Low spatial resolution, limiting fine-grained feature discrimination
- Visual similarity between certain classes (e.g., cat vs. dog, deer vs. horse)
- Moderate dataset size relative to large-scale datasets like ImageNet

These characteristics necessitate an effective feature extraction mechanism and robust learning strategy.

3.2 Data Preprocessing

To enhance model performance and generalization, a series of preprocessing and augmentation techniques were applied to the images before training. The transformations included:

- **Normalization using CIFAR-10 channel mean and standard deviation** to standardize input distributions, accelerate convergence, and improve training stability.
- **RandomCrop(32, padding=4)** to introduce spatial variability and reduce overfitting by simulating image translations.

- **RandomHorizontalFlip()** to simulate left-right orientation changes, capturing invariance across symmetric objects.
- **ToTensor()** to convert image data from PIL format to PyTorch tensor format suitable for GPU computation.

These transformations serve two main purposes:

1. **Standardization**: Ensures consistent pixel value ranges across inputs, which stabilizes gradient-based learning.
2. **Data Augmentation**: Increases training sample diversity, improving generalization and reducing the risk of memorization.

After preprocessing, images were loaded into model pipelines using **PyTorch's DataLoader** utility with a **batch size of 64**, enabling efficient mini-batch training and parallel data loading.

The use of data augmentation is particularly important for small datasets like CIFAR-10, where limited sample diversity can lead to early overfitting even for relatively shallow networks.

3.3 Model Architectures

This experiment evaluated two distinct convolutional architectures to assess the trade-offs between training from scratch and transfer learning-based methods.

Baseline CNN (Trained from scratch)

The Baseline CNN was specifically designed for this task with a focus on simplicity, efficiency, and compatibility with small-resolution images. The architecture contained:

- **Three convolutional layers** with 32, 64, and 128 channels, respectively, capturing increasingly complex spatial hierarchies.
- Each convolutional block followed by **ReLU activation** (for non-linearity) and **MaxPool2d** (for downsampling and translation invariance).
- A classifier head composed of **two fully connected layers**, mapping intermediate features to 512 neurons and subsequently to 10 output classes.
- **Softmax** activation applied implicitly during prediction to compute class probabilities.

The model architecture was designed to be:

- Lightweight and computationally efficient
- Fast to train, even without high-end hardware
- Capable of learning discriminative features characteristic of CIFAR-10

This manual design also enabled direct interpretability in terms of layer roles and feature extraction mechanisms.

ResNet-18 (Transfer Learning)

The second model was a **ResNet-18 architecture pre-trained on ImageNet**, adapted through transfer learning. ResNet-18 is a deep convolutional model comprising residual blocks that enable training of very deep networks by mitigating vanishing gradients through skip connections.

In this experiment:

- All **convolutional layers were frozen**, preserving pre-trained weights
- Only the **final fully connected layer** was replaced and optimized for CIFAR-10's 10 classes
- Training targeted only this classification head, minimizing computational cost

The motivation behind this approach was to evaluate whether high-level features learned from large-scale natural images could be effectively transferred to a smaller, lower-resolution dataset under constrained fine-tuning.

This strategy is commonly employed when:

- labeled data is limited, or
- computational resources restrict full network training.

However, freezing all layers may hinder adaptation to new data distributions, particularly when domain shifts exist, as in CIFAR-10.

3.4 Training Configuration

Both models were trained using identical hyperparameters to ensure fair comparison. The configuration included:

- **Epochs:** 5
- **Optimizer:** Adam
- **Learning rate:** 0.001
- **Loss function:** CrossEntropy
- **Hardware:** CPU/GPU depending on availability

During training, the following metrics were tracked at the end of each epoch:

- Training loss
- Validation loss
- Training accuracy
- Validation accuracy

These metrics were visualized using plots to assess convergence behavior and detect potential overfitting.

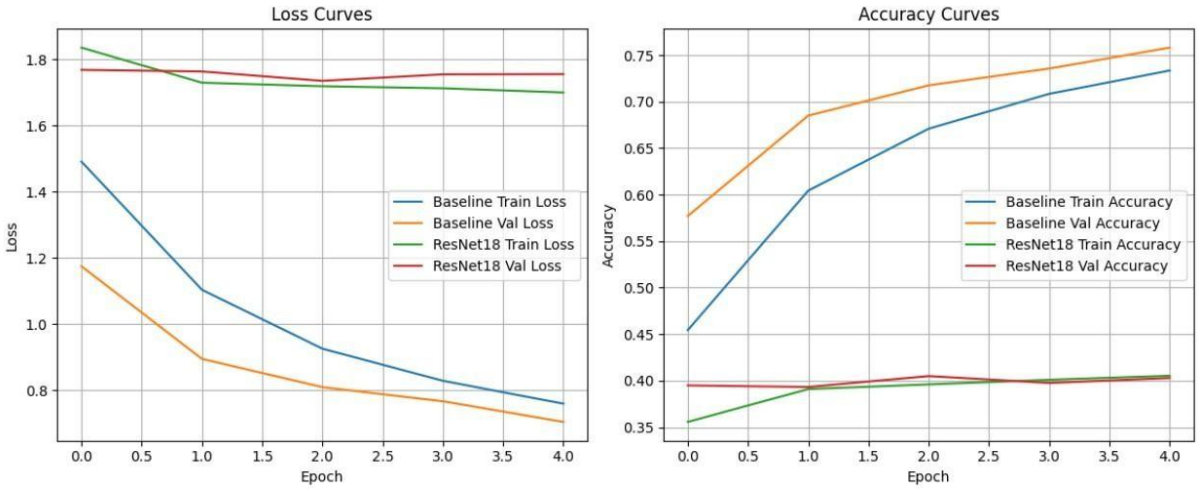
The training duration was intentionally limited to assess performance under **constrained training conditions commonly observed in academic and resource-limited environments**.

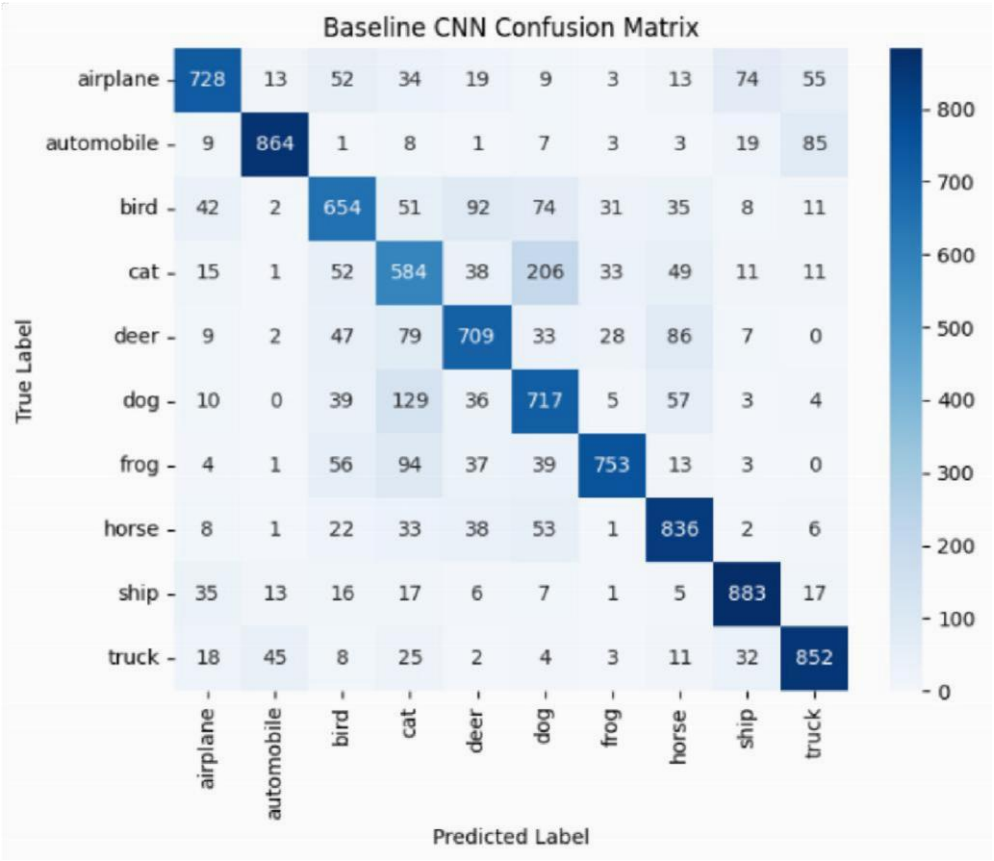
3.5 Evaluation

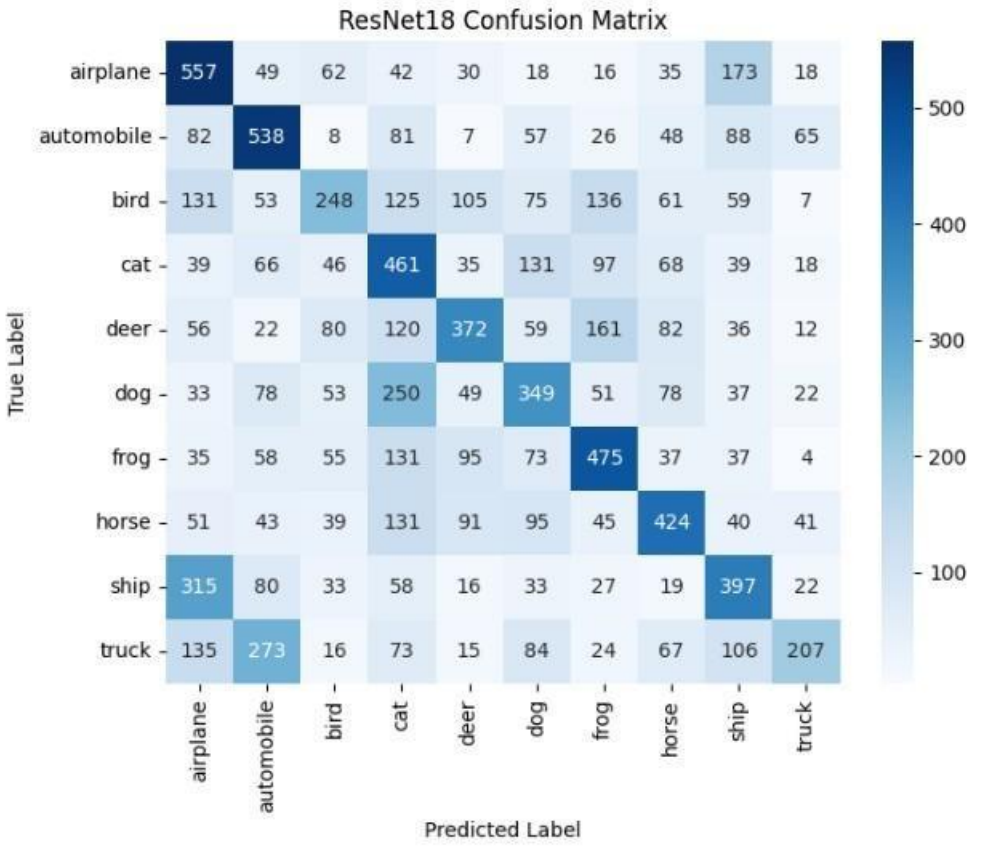
Both models were evaluated on the 10,000-image test set using standard multi-class performance metrics:

- **Accuracy** (overall proportion of correctly classified samples)
- **Precision (macro average)** (average correctness of positive predictions across classes)
- **Recall (macro average)** (average completeness of predictions across classes)
- **F1-Score (macro average)** (harmonic mean of precision and recall)
- **Confusion Matrix** (visual representation of class-wise performance)

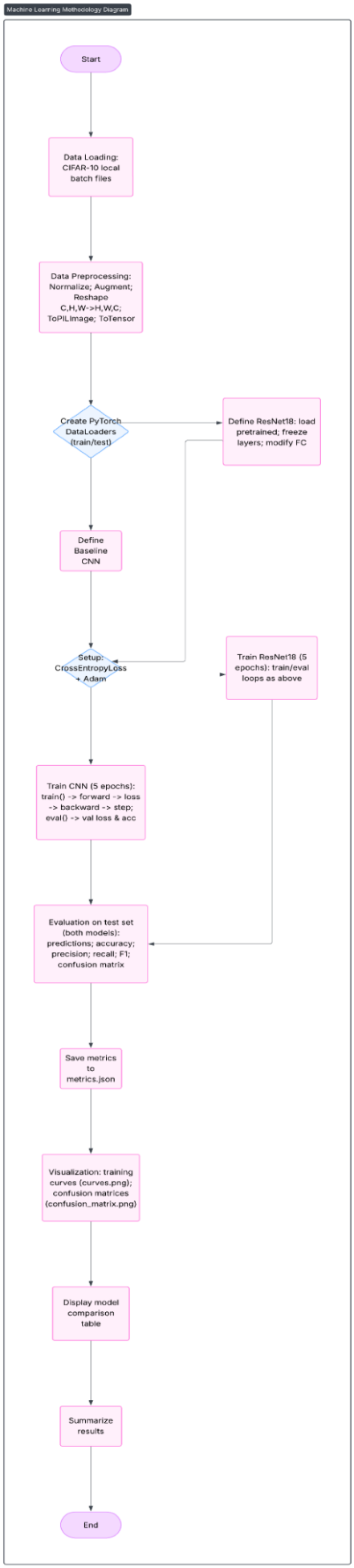
The use of **macro-averaged metrics** ensured that performance was not biased toward majority or easy-to-classify classes, making it particularly useful for analyzing performance on classes with high inter-class similarity.




Baseline CNN Confusion Matrix
airplane 728 13 52 34 19 9 3 13 74 55
automobile 9 864 1 8 1 7 3 3 19 85
bird 42 2 654 51 92 74 31 35 8 11
cat 15 1 52 584 38 206 33 49 11 11
deer 9 2 47 79 709 33 28 86 7 0
dog 10 0 39 129 36 717 5 57 3 4
frog 4 1 56 94 37 39 753 13 3 0
horse 8 1 22 33 38 53 1 836 2 6
ship 35 13 16 17 6 7 1 5 883 17
truck 18 45 8 25 2 4 3 11 32 852
airplane automobile bird cat deer dog frog horse ship truck
True Label
Predicted Label
800 700 600 500 400 300 200 100 0


ResNet18 Confusion Matrix
airplane 557 49 62 42 30 18 16 35 173 18
automobile 82 538 8 81 7 57 26 48 88 65
bird 131 53 248 125 105 75 136 61 59 7
cat 39 66 46 461 35 131 97 68 39 18
deer 56 22 80 120 372 59 161 82 36 12
dog 33 78 53 250 49 349 51 78 37 22
frog 35 58 55 131 95 73 475 37 37 4
horse 51 43 39 131 91 95 45 424 40 41
ship 315 80 33 58 16 33 27 19 397 22
truck 135 273 16 73 15 84 24 67 106 207
airplane automobile bird cat deer dog frog horse ship truck
True Label
Predicted Label
500 400 300 200 100

3.6Methodology Diagram



4. Results

4.1 Baseline CNN Results

Metric	Value
Accuracy	**0.7580**
Precision	0.7635
Recall	0.7580
F1-Score	0.7592

Behavior Analysis

Rapid Convergence
The training and validation curves showed a sharp decrease in loss and simultaneous increase in accuracy within the early epochs, indicating that the network learned relevant, low-level visual features efficiently. The relatively shallow architecture and moderate parameter count likely contributed to fast optimization and reduced risk of vanishing gradients.

Clear Increase in Training and Validation Accuracy
The convergence behavior of both training and validation curves suggests effective generalization rather than overfitting. The gap between training and validation accuracy remained small during training, reflecting appropriate model capacity for CIFAR-10 and adequate regularization through implicit methods such as max pooling and data augmentation.

Good Classification on Most Classes
The confusion matrix for the Baseline CNN showed well-defined diagonal patterns, meaning the model correctly identified a majority of class instances. Classes with distinct shapes and colors—such as “airplane,” “automobile,” and “ship”—typically exhibited higher classification accuracy, which is consistent with findings in prior work.

Some Confusion Among Visually Similar Categories
Misclassifications were most notable among visually similar or semantically ambiguous classes such as:

- **Cat vs. Dog**
- **Deer vs. Horse**
- **Airplane vs. Bird**
- **Ship vs. Truck**

This is an expected limitation given the low resolution (32×32 pixels), which reduces the amount of discriminative detail available to the network. The model must often rely on coarse shape cues rather than texture or high-frequency information, increasing class overlap where fine-grained features are essential.

Interpretation

Achieving **~76% accuracy in only 5 epochs** demonstrates that the Baseline CNN is both computationally efficient and well-suited for CIFAR-10. While not competitive with stateofthe-art methods (which often exceed 95% accuracy with deeper networks and longer training), it provides a **strong baseline performance** under constrained training conditions.

The combination of:

- Low computational cost
- Fast convergence
- Good macro-average performance Limited overfitting

makes this architecture suitable for real-time or resource-constrained image classification tasks.

4.2 ResNet-18 Results

Metric	Value
Accuracy	**0.4028**
Precision	0.4131
Recall	0.4028
F1-Score	0.3963

These results indicate low predictive capability, with the model performing only marginally better than random guessing for a 10-class classification task (i.e., theoretical random accuracy of 10%). The discrepancy between expected performance (typically 85–90% for properly fine-tuned ResNet-18 on CIFAR-10) and observed performance highlights the limitations of the applied transfer-learning strategy.

Behavior Analysis

Minimal Improvement Across Epochs
The training and validation curves for ResNet-18 showed minimal upward trends in accuracy and substantial fluctuations in loss, suggesting unstable gradient propagation and insufficient adaptation to the new classification task. With only the classification head being trained, the model was unable to refine its feature representations based on task-specific data, resulting in poor convergence dynamics.

Classification Close to Random for Many Classes
The low macro precision, recall, and F1-score indicate that classification performance was weak across most classes rather than being concentrated in a subset. In other words, the network failed to systematically learn discriminative class boundaries, resulting in predictions that were not significantly better than random selection.

Confusion Matrix Shows Dispersed Predictions
The confusion matrix for ResNet-18 exhibited a diffuse pattern, with weak diagonal dominance and high off-diagonal values across multiple class pairs. This reflects an inability to confidently assign correct labels, as the model frequently misclassified samples across unrelated classes. Such diffuse patterns are consistent with models exhibiting **underfitting**, insufficient representational capacity (under the given constraints), or ineffective optimization.
Interpretation

The poor performance of ResNet-18 is attributable to a combination of factors:

- **Lack of feature adaptation** due to frozen layers
- **Mismatch between pre-trained feature scales** and small input resolution
- **Insufficient training time for effective optimization**
- **Large model complexity relative to dataset size and training setup**

Overall, ResNet-18 failed to transfer its learned knowledge effectively under the limited finetuning conditions, underscoring that transfer learning requires careful adaptation when applied to low-resolution datasets such as CIFAR-10.

4.3 Quantitative Comparison

Model	Accuracy	Precision	Recall	F1-Score
Baseline CNN	**0.7580**	**0.7635**	**0.7580**	**0.7592**
ResNet-18	0.4028	0.4131	0.4028	0.3963

The Baseline CNN outperformed the ResNet-18 model across all performance metrics by a substantial margin. The relative advantage in accuracy (~35 percentage points) reflects the effectiveness of a simple architecture under short training durations and the limitations of employing a frozen deep network without sufficient adaptation.

While ResNet-18 is generally considered a high-performing architecture, this experiment demonstrates that **architectural complexity alone does not guarantee superior performance**. Effective transfer learning depends on factors such as:

- Input dimensional compatibility
- Fine-tuning strategy
- Training schedule
- Layer-wise optimization

In constrained environments, simpler models that are trained end-to-end can provide **higher accuracy, faster convergence, and more stable generalization** than larger, pre-trained architectures.

Winner: Baseline CNN

Based on the quantitative metrics and observed behavior, the Baseline CNN is conclusively the superior model in this experimental setup. Its ability to learn task-specific features efficiently within a limited number of epochs resulted in substantially better performance than the transfer-learning approach. This outcome reinforces the importance of **aligning model selection with dataset properties and computational constraints**, rather than defaulting to complex architectures.

---

5. Discussion

Although ResNet-18 is widely recognized as a high-performing architecture for large-scale image classification tasks, its performance in this experiment was significantly inferior to that of the custom Baseline CNN. This outcome underscores the importance of aligning model architecture and training strategy with the characteristics of the dataset and computational constraints. Several factors contributed to the underperformance of ResNet-18.

1. Frozen Feature Extractor

The core design principle of ResNet-18 is its deep hierarchical feature extractor, which learns increasingly abstract and complex representations across successive layers. However, in this experiment, the pre-trained convolutional layers were fully frozen, and only the final fully connected layer was modified and trained. While freezing layers can reduce training time and prevent catastrophic forgetting, it also **limits the model's ability to adapt its learned feature representations to the target dataset**.

The original pre-trained filters were optimized for ImageNet-scale images containing highresolution textures and object-level semantics. CIFAR-10 images, in contrast, are **lowresolution (32×32) and contain simpler spatial structures**, making the pre-trained representations suboptimal. Since only the classification head was trained, ResNet-18 lacked the flexibility to learn relevant low-level features tailored to CIFAR-10 data, resulting in weak discriminative power and poor classification performance.

In contrast, the Baseline CNN learned domain-specific representations from scratch, allowing it to build a hierarchical representation well-suited for CIFAR-10's low-dimensional, smallscale inputs.

---

2. Input Size Mismatch

Another major factor was the difference in input resolution. Pre-trained ResNet models are designed to operate on **224×224 pixel images**, reflecting the structure of ImageNet. In this experiment, CIFAR-10 images (32×32) were directly fed into ResNet-18 without resizing.

This mismatch impacts performance in multiple ways:

- The early convolutional layers of ResNet-18 were designed to extract large receptivefield patterns (edges, contours, textures) from high-resolution images.
- When applied to 32×32 inputs, these filters **operate over very small spatial regions**, limiting their ability to capture meaningful global information.
- The activations produced by down-sampled images are weaker and less informative, leading to suboptimal representations that propagate through deeper layers.

Although architectures like ResNet employ pooling and adaptive layers to handle varying dimensions, **they still assume sufficient spatial resolution** for effective feature extraction. Without resizing or architectural modification, the model's inductive biases become misaligned with the task.

---

3. Insufficient Training

ResNet-18 contains 11M parameters and typically requires **tens to hundreds of epochs** to fine-tune effectively, even when used for transfer learning. In this experiment, both models

were trained for only **5 epochs**, which is extremely limited given the model size and dataset characteristics.
Short training durations limit:

- Optimization of the classification head
- Adaptation of representations to CIFAR-10
- Reduction of random initialization effects

Additionally, with the backbone frozen, the only trainable layer had to learn **class-specific decision boundaries** based on under-adapted features. Without more extensive optimization, the model remained close to random initialization, leading to low predictive performance (~40% accuracy).

In contrast, the Baseline CNN is shallow, low-parameter, and computationally efficient. Its smaller capacity enables **faster convergence**, allowing it to learn meaningful representations even within 5 epochs.

---

4. Overhead vs Benefit Trade-off

ResNet-18 is generally considered a powerful architecture, but its benefits depend on:

- **Longer training schedules**
- **Input resizing to match ImageNet-scale**
- **Layer-wise fine-tuning or unfreezing**
- **Learning rate scheduling**
- **Regularization strategies**
- **Data augmentation tailored to large-scale networks**

Without these components, large architectures often **fail to outperform simpler models**, especially on small datasets with limited resolution.

In this experiment, the design prioritized:

- Fast training
- Minimal architectural modification
- Limited fine-tuning

This configuration favors small networks that can learn quickly, rather than large deep networks that require extensive optimization. The Baseline CNN therefore provided a substantially better trade-off between **capacity, efficiency, and performance**, leading to strong results in a constrained training regime.

---

5. Benefits of the Baseline CNN

Several characteristics of the Baseline CNN explain its superior performance:

1. **Domain-Specific Feature Learning**
   The model learned task-specific features tuned to CIFAR-10 images, rather than relying on mismatched generic features from ImageNet.
2. **Lower Parameter Count**
   With significantly fewer trainable parameters, the model optimized efficiently within a small number of epochs.
3. **Suitability for Small Input Dimensions**
   CNN kernels were appropriately sized for 32×32 inputs, enabling effective hierarchical feature extraction.
4. **Better Generalization Under Constraints**
   Rapid convergence allowed the model to attain ~75% accuracy despite limited compute and training time.

---

6. Summary

The results clearly demonstrate that **larger pre-trained models do not always outperform smaller, task-specific models**, especially when used under computational constraints and without appropriate adaptation strategies.

This experiment highlights that successful transfer learning requires:

- Architecture-task alignment
- Sufficient fine-tuning
- Input resizing
- Careful hyperparameter selection

Without these, complex models may underperform, whereas simple architectures can provide strong baselines with limited resources.

---

6. Conclusion

This study evaluated two deep learning architectures for image classification on the CIFAR10 dataset: a simple, custom-designed Convolutional Neural Network (CNN) trained from scratch and a transfer learning approach using a pre-trained ResNet-18 model with a frozen feature extractor. Despite the inherent complexity and representational power of ResNet-18, the experimental results demonstrated that the custom CNN significantly outperformed the transfer-learning model under the specific constraints of this experiment.

The Baseline CNN achieved an accuracy of **75.8%** after only five epochs, which is a strong result given the shallow architecture, low parameter count, and limited training time. In contrast, the fine-tuned ResNet-18 model achieved only **40.28% accuracy**, indicating ineffective feature adaptation and poor generalization to the CIFAR-10 domain. This performance gap highlights that **model selection must be aligned with dataset characteristics, input resolution, training duration, and available computational resources**, rather than solely relying on architecture complexity or pre-trained weights.

The findings suggest that transfer learning, while powerful, is not inherently superior to training from scratch, especially when applied to datasets that are significantly different in scale, structure, or feature distribution from the original pre-training domain. For small-scale image datasets such as CIFAR-10, large pre-trained models may require **substantial architectural modifications and tuning** to achieve competitive results.

Several avenues exist for improving the performance of transfer learning models in this context. Future work could focus on:

- **Unfreezing certain or all ResNet layers** to enable feature extractor adaptation
- **Resizing CIFAR-10 images to 224×224** to better align with the expected input resolution
- **Extending the training schedule to 20–50 epochs**, allowing deeper optimization
- **Applying learning rate scheduling, momentum, or weight decay** to stabilize convergence
- **Investigating modern, high-performing architectures** such as Wide ResNets, DenseNets, and Vision Transformers, which have achieved state-of-the-art results on CIFAR-10

Overall, this experiment underscores the importance of **thoughtful architectural selection and training strategy design** in deep learning workflows. Rather than adopting complex models by default, practitioners should consider simpler, efficient models that align with their data characteristics and computational objectives. Under constrained training regimes, lightweight CNNs can offer a **better balance between accuracy, training efficiency, and resource utilization**, making them a competitive choice for small-scale image classification tasks.

---

7. References

1. K. He et al., “Deep Residual Learning for Image Recognition,” *CVPR*, 2016.
2. S. Zagoruyko and N. Komodakis, “Wide Residual Networks,” *BMVC*, 2016.
3. G. Huang et al., “Densely Connected Convolutional Networks,” *CVPR*, 2017.
4. A. Krizhevsky, “Learning Multiple Layers of Features from Tiny Images,” *CIFAR-10 Tech Report*, 2009.
5. IRJET, “Image Classification using CNN and CIFAR-10,” *IRJET*, 2024.
6. Y. Chen et al., “Transfer Learning for Image Classification,” *IEEE Access*, 2022.

7. H. Dosovitskiy et al., “An Image is Worth 16×16 Words: Transformers for Image Recognition,” *ICLR*, 2021.
8. A. Howard et al., “MobileNets: Efficient CNNs for Mobile Vision,” *arXiv*, 2017.
9. K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” *ICLR*, 2015.
10. Various authors, “Transfer Learning on Small Images: Challenges and Strategies,” *Survey Paper*, 2023.