Enhancing CIFAR-10 Image Classification using Transfer Learning with ResNet18

Karthik Reddy Kallu
(6420063)
Masters in Computer Science
Florida International University

Likhith Krishna Movva
(6488905)
Masters in Computer Science
Florida International University

Venkata Sesha Sai Malneeedi
(6421300)
Masters in Computer Science
Florida International University

Abstract—

Through the use of a pretrained ResNet18 model and a transfer learning approach, we sought to improve classification accuracy on the CIFAR-10 dataset by switching from custom-designed convolutional neural networks (CNNs).

We first created a number of manually constructed architectures, such as a simple three-layer convolutional network, a more sophisticated bespoke CNN, and a two-layer fully connected network. Although these models established a solid basis and produced respectable results, their shallow depth, limited capacity for feature extraction, and requirement for intensive hyperparameter adjustment to get greater accuracy were their main drawbacks. During this phase, the highest validation accuracy achieved was approximately 73%.

We used a pretrained ResNet18 model, which was first trained on the large ImageNet dataset, to get around these restrictions. By solving vanishing gradient problems, ResNet18's deep residual learning with skip connections makes it possible to train deeper networks effectively. Using the Adam optimiser with a learning rate of 1e-4, we adjusted the last fully connected layer to provide 10 classes unique to CIFAR-10. With only minor modifications, we were able to adapt ResNet18 to our task through transfer learning.

This change increased validation accuracy from 73% to 82%, resulting in a notable performance improvement. In addition to improving accuracy, using a pretrained network allowed for a far deeper and more complex architecture than manually created models, while also reducing training time and processing overhead. Confusion matrices, classification reports, training/validation accuracy and loss curves, and other supporting visualisations confirmed the model's enhanced capacity for generalisation.

In conclusion, this effort bridges the gap between the development of educational models and industry-standard solutions by showing the useful benefits of transfer learning and pretrained architectures in image classification tasks.

1. INTRODUCTION

Computer vision has been completely transformed by the quick development of deep learning, which has enabled machines to do complex visual recognition tasks that were previously thought to be only possible by humans. Image classification has emerged as a key difficulty among these jobs, with extensive applications in consumer electronics, security systems, medical diagnostics, and autonomous driving The development of complex neural network structures, the availability of huge annotated datasets, and improvements in processing capacity are all major factors in deep learning's impressive achievement in this field.

CIFAR-10 is one of the most popular benchmark datasets for image classification. The 60,000 colour, 32x32 pixel images in this dataset are evenly split among 10 different categories, such as cars, trucks, aeroplanes, frogs, horses, birds, cats, deer, dogs, and cars. CIFAR-10 is a great testbed for assessing model robustness and generalisation ability, despite having a smaller resolution than contemporary datasets. Its notable challenges include significant variation in object pose, background clutter, lighting differences, and high inter-class similarity.

In the first stage of this research, we used the PyTorch framework to create and test unique convolutional neural networks (CNNs) from scratch. A simple two-layer fully connected network and a three-layer CNN that gradually collected information from input photos were examples of early architectures. The CIFAR-10 dataset was used to train and validate these models, and they obtained an acceptable validation accuracy of about 73%. However, further performance advances were hindered by problems including slow convergence, overfitting, and limited feature extraction because of the shallow architectures of the networks.

In order to overcome these constraints, we used a pretrained ResNet18 model and switched to a transfer learning strategy. Using residual connections, which successfully address the vanishing gradient issue and allow for the training of deeper networks, ResNet18—first presented by He et al.—achieved ground-breaking results on extensive datasets such as ImageNet. Strong feature representations were transferred to the CIFAR-10 problem by employing a model pretrained on millions of different images, which greatly sped up training and improved model performance.

We adjusted the last fully connected layer of ResNet18 for this adaption in order to provide 10 classifications that correspond to the CIFAR-10 categories. With a learning rate of 0.0001, the network was adjusted using the Adam optimiser, which enabled the model to learn dataset-specific patterns while maintaining strong low-level feature detectors.

Performance significantly improved with the use of ResNet18, increasing validation accuracy from 73% with bespoke CNNs to about 82%. This significant improvement shows the value of transfer learning, especially in settings with constrained data and processing power. Improved class-wise prediction consistency and decreased errors were found through additional analysis using confusion matrices and classification reports, especially for categories where previous models had trouble.

In addition to showing the technical advantages of using pretrained deep learning models, this research offers insightful information about model selection, optimisation techniques, and deployment procedures in real-world settings. All things considered, it shows how transfer learning has a revolutionary effect on contemporary deep learning processes by bridging the gap between scholarly experimentation and industry-level solutions.

—----

2. Techniques

In order to tackle the image classification challenge on the CIFAR-10 dataset, we used a number of machine learning and deep learning approaches. Our strategy changed from manually creating simple models to using a pretrained network and transfer learning. The following describes the main strategies employed:

2.1 Custom Neural Network Architectures

In order to establish an intuitive understanding of the fundamentals of image classification using deep learning, we concentrated on creating simple neural network models from scratch throughout the project's first phase. These specially constructed networks provided an essential basis for learning about training techniques, evaluation techniques, and model architecture design.
During this stage, two important models were put into practice:

Two-Layer Fully Connected Network:
The first model constructed was a simple two-layer fully connected (dense) network. The architecture followed a basic structure:

Input Layer:
The initial step was to flatten each 32x32 RGB image from CIFAR-10 (which had three colour channels) into a 1D vector with $3 \times 32 \times 32 = 3072$ features.

Hidden Layer:
Basic input feature transformations were made possible by the application of a dense layer made up of many hidden units. This layer was followed by a non-linear activation function (ReLU), which added non-linearity and helped the model in learning complex correlations.

Output Layer:
The final output layer was another fully connected layer that mapped the hidden representations to 10 output nodes, each corresponding to one of the CIFAR-10 classes.

Strengths:

- Easy to implement and visualize.
- Provided a simple baseline to measure improvements when moving to more advanced architectures.

Limitations:

- Unable to take advantage of spatial structure in images, handling each pixel separately without taking into consideration the links between neighbouring pixels.
- Ineffective use of parameters, multiple weights are needed to link all input features to the hidden layer.
- prone to poor generalisation and overfitting as dataset complexity and input size rise.

As a result, although useful for preliminary testing, the two-layer fully linked network was unable to adequately capture the rich spatial data required for image classification with high accuracy.

Three-Layer Convolutional Neural Network:
We then used a three-layer convolutional neural network after realising the limitations of simply dense architectures for image tasks. Through the use of convolutional procedures, this design aims to take use of the spatial hierarchies present in visual data.
The architecture was structured as follows:

First Convolutional Layer:
Applied to the input images a number of learnable filters (kernels) with small receptive fields (e.g., 5x5). As a result, the model was able to identify regional patterns such as textures, corners, and edges.

Activation Function (ReLU):
After the convolution procedure, the ReLU activation function was used to induce non-linearity, which allowed the model to learn intricate mappings.

Second Convolutional Layer:
In order to create higher-level feature maps and capture more abstract patterns, like object pieces, another convolutional layer was stacked.

Fully Connected Layer:

The feature maps were flattened, and then the retrieved features were mapped into final class scores using a fully linked layer.

Strengths:

- Convolutions improved memory efficiency by drastically lowering the number of parameters as compared to dense layers.
- With the introduction of translation invariance, the model was able to identify items in the image regardless of where they were located.
- By concentrating on significant areas rather than treating every pixel equally, local connection allowed for improved generalisation.

Limitations:

- Shallow CNNs (with only two convolutional layers) could capture only basic visual features, limiting their ability to model complex objects.
- In the absence of extra layers like dropout, batch normalisation, or pooling, the network was still susceptible to overfitting.

Despite these drawbacks, the three-layer CNN outperformed the fully connected network in terms of validation accuracy (~73%), proving the usefulness of convolutional architectures for computer vision tasks.

2.2 Transfer Learning with Pretrained ResNet18:

We implemented transfer learning, a potent technique used in modern deep learning processes, to get over the intrinsic drawbacks of shallow, custom-built networks. Adapting a model that has previously been trained on a sizable and varied dataset to a new but similar task is known as transfer learning. This method makes use of the pretrained model's stored prior knowledge, which speeds up convergence and frequently improves performance especially when the new dataset is smaller.

In this project, transfer learning was instrumental in improving the validation accuracy from around **73% to 82%** on the CIFAR-10 dataset.

Model Selection: ResNet18

For transfer learning, we chose ResNet18 as the pretrained model. The concept of residual learning was first introduced by the seminal deep learning architecture ResNet18, which is a member of the Residual Network (ResNet) family. ResNet's primary innovation is the use of skip connections, also known as shortcut connections, which enable the gradient to pass straight through layers with minimal loss.

Why ResNet18?

Moderate Depth:

ResNet18's 18 layers provide a reasonable compromise between computational efficiency and model complexity, which makes it appropriate for fine-tuning on smaller datasets such as CIFAR-10.

Skip Connections:

By minimising the vanishing gradient issue that frequently arises during deep network training, these links improve gradient flow and facilitate optimisation.

Pretraining on ImageNet:

The ImageNet dataset, which included more than 1 million photos in 1000 classes, served as the initial training set for ResNet18. Consequently, the model has already acquired general-purpose visual features that are helpful for classifying CIFAR-10 images, such as texture recognition, edge detection, and form representation.

Thus, ResNet18 was an ideal candidate for our task, providing a strong, pretrained backbone upon which we could fine-tune for specific CIFAR-10 classification.

Model Modification:

Despite its great effectiveness, the pretrained ResNet18 architecture was initially intended for ImageNet classification with 1000 output classes. Certain changes were required to adapt the model to the CIFAR-10 dataset. Initially, a new fully connected layer that produced 10 outputs—which corresponded to the 10 classes of CIFAR-10—replaced the original final completely connected layer, which produced 1000 classes (such as aeroplane, automobile, bird, cat, and others). All previous layers, including residual blocks, batch normalisation layers, and convolutional layers, were simultaneously kept with their pretrained weights. Essential visual elements are captured by these layers, including mid-level elements like forms and patterns and low-level elements like edges and textures. Compared to training a network from start, the fine-tuning process was far more effective and efficient since the model benefited from a strong initialisation by keeping these pretrained layers. To better adapt to the CIFAR-10 domain, the prior layers were either frozen initially or meticulously adjusted during training, leaving only the recently added final layer to be randomly initialised and retrained.

Fine-tuning:

The network was fine-tuned after the architecture was changed to match the broad features that ImageNet had taught to the unique distributions and properties of the CIFAR-10 dataset.

Fine-tuning Details:

- **Entire Network Training:** We decided to fine-tune the entire network rather than freezing the convolutional basis, which is typical in transfer learning.This enhanced model performance by enabling low-level and mid-level features to marginally adapt to CIFAR-10-specific patterns.

- **Optimizer:** We used the Adam optimizer, known for its adaptive learning rates and robustness to noisy gradients, making it a good choice for fine-tuning tasks.
- **Learning Rate:** A comparatively low value of 0.0001 was chosen as the learning rate. Pretrained weights are changed gradually without erasing the crucial features acquired during ImageNet training thanks to a low learning rate.
- **Training Process:** Utilising the train_part34() code specified in the project notebook allowed us to consistently measure accuracy and loss across epochs. The network rapidly converged to a greater validation accuracy throughout the course of the ten epochs of training.

Benefits of Fine-tuning:

- Accelerated convergence compared to training from scratch.
- Better generalization on unseen CIFAR-10 images.
- More effective utilization of limited computational resources.

2.3 Optimization Techniques

In order to effectively train deep learning models, optimisation is essential. This ensures that the model parameters converge towards values that minimise the loss function. To optimise performance in this project, several optimisation techniques were used at various phases of the model development process.

Optimizer Selection:

Initially, the main optimiser for the specially designed neural network designs was Stochastic Gradient Descent (SGD), which was used for the two-layer fully connected network and the three-layer convolutional neural network. Faster and more memory-efficient training is made possible by SGD, which uses tiny batches of data to update the model parameters. But occasionally, especially when the learning rate is not adjusted correctly, basic SGD might experience delayed convergence rates and become stuck in saddle points or local minima in the loss landscape.

We used the Adam optimiser (Adaptive Moment Estimation) to improve optimisation efficiency while fine-tuning the pretrained ResNet18 model. By calculating distinct adaptive learning rates for various parameters, Adam combines the benefits of two further SGD extensions: Adaptive Gradient Algorithm (AdaGrad) and Root Mean Square Propagation (RMSProp).

Adam is able to dynamically modify the learning rate for every parameter by keeping an exponentially decaying average of previous gradients (first moment) and past squared gradients (second moment). Because of this, Adam was especially well-suited to optimising ResNet18's deeper and more complex design, which resulted in quicker convergence and improved noise-cancelling capabilities.

Loss Function:

Cross-Entropy Loss was selected as the loss function for all training phases.

Because it measures the difference between the actual distribution (the true labels) and the projected probability distribution (the model's output), cross-entropy loss is frequently employed for multi-class classification issues.

In terms of mathematics, it penalises inaccurate predictions more severely by measuring the difference between the true and expected label distributions. The anticipated probabilities are more in line with the actual labels when the cross-entropy loss is smaller.

Key reasons for choosing Cross-Entropy Loss include:

- It encourages models to not just predict the correct class but to do so with high certainty because it is sensitive to prediction confidence.
- It guarantees that probability distributions are appropriately formed and works well with the softmax output layer used in classification models.

2.4 Model Evaluation Techniques

Accuracy Metrics: The model's learning behaviour was evaluated by continually measuring training and validation accuracy throughout epochs. Monitoring these indicators made it easier to determine whether the model was experiencing problems like performance loss or stagnation or if it was continuously improving.

Loss Curves: Both training and validation sets' loss values were recorded and shown against the number of epochs. It was possible to recognise signs of overfitting (when validation loss rises while training loss falls) or underfitting (where both losses stay high) by analysing loss curves, which gave insight into optimisation trends

.

Confusion Matrix: Post model testing, prediction performance for each CIFAR-10 class was visualised using a confusion matrix. This matrix helped us in identifying class-specific deficiencies and helped us identify which courses were most commonly mistaken with one another.

Classification Report: A thorough classification report was produced, including the F1-score, precision, and recall for every class separately. In addition to identifying imbalances or differences in the model's predictions, these measures provided a deeper knowledge of the model's capacity to accurately identify each category than just total accuracy.

3. Results

In this section, we present and analyze the performance of our models on the CIFAR-10 dataset after applying transfer learning using the pretrained ResNet18 architecture.

3.1 Training and Validation Accuracy

Ten epochs were used to record the accuracy of the training and validation processes.
The training accuracy rose gradually, as seen in Figure 1, and by the tenth period, it had reached about 97%. At almost 83%, the validation accuracy also increased.
This suggests that there was minimal overfitting and good generalisation of the model.

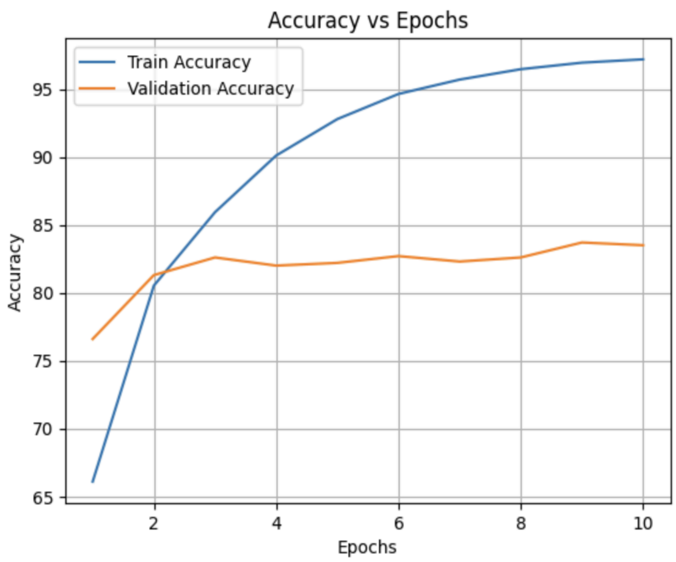

Figure 1: Training and Validation Accuracy vs Epochs

3.2 Training and Validation Loss

Throughout the training procedure, the validation and training losses were also monitored.
The model was effectively learning over time, as seen by Figure 2, which displays a notable and steady decrease in training loss. The model's capacity to generalise was further supported by the validation loss, which stayed low and steady.

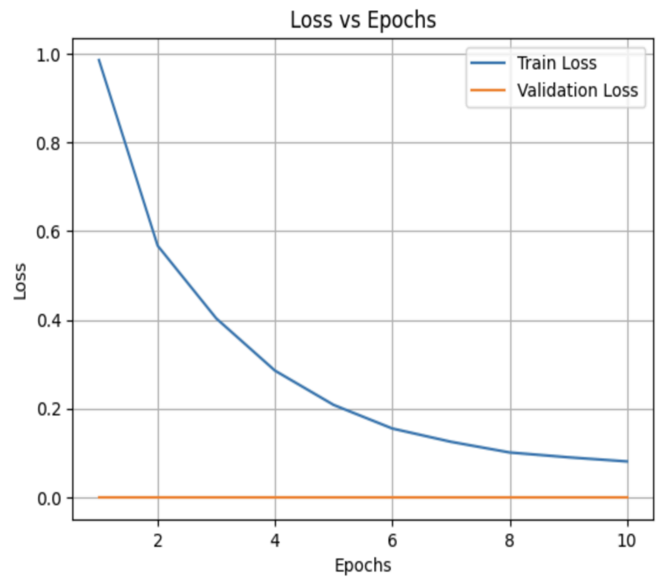

Figure 2: Training and Validation Loss vs Epochs

3.3 Confusion Matrix Analysis

The model's performance in several classes is shown in detail via the confusion matrix.
The majority of classes were correctly classified, as shown in Figure 3, with very few instances of confusion between related categories (such as cars and trucks or cats and dogs).
The model's excellent classification performance across all categories is seen from the heatmap.

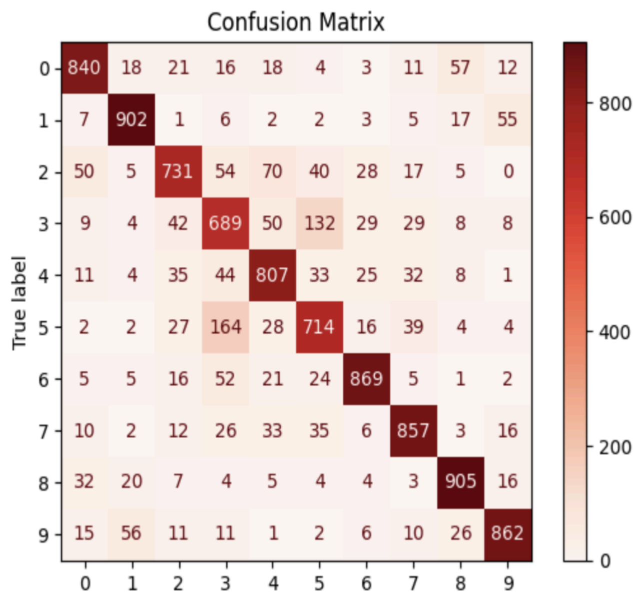

Figure 3: Confusion Matrix on CIFAR-10 Test Set

3.4 Classification Report

A classification report with the F1-score, precision, and recall values for each class separately was produced. The majority of classes had high precision and recall scores, as seen in Figure 4. The total weighted and macro averages roughly matched the stated test accuracy, coming in at about 0.82 (or 82%).

Predicted label

	precision	recall	f1-score	support
0	0.86	0.84	0.85	1000
1	0.89	0.90	0.89	1000
2	0.81	0.73	0.77	1000
3	0.65	0.69	0.67	1000
4	0.78	0.81	0.79	1000
5	0.72	0.71	0.72	1000
6	0.88	0.87	0.87	1000
7	0.85	0.86	0.85	1000
8	0.88	0.91	0.89	1000
9	0.88	0.86	0.87	1000
accuracy			0.82	10000
macro avg	0.82	0.82	0.82	10000
weighted avg	0.82	0.82	0.82	10000

Figure 4: Classification Report for CIFAR-10 Test Set

3.5 Key Observations

- With ResNet18 fine-tuning, the model's final test accuracy was 82%.
- Strong class-wise performance is shown by the confusion matrix, particularly for well defined classes like cars, aeroplanes, and birds.
- According to the classification report, performance was balanced across classes, with only minor difficulties noted in visually comparable courses.

4 Conclusion:

In this study, we investigated the use of deep learning techniques for image classification from the CIFAR-10 dataset. We began with custom-built neural networks then progressed to transfer learning using a pretrained ResNet18 model.

First, we created and assessed basic architectures including a three-layer convolutional network and a two-layer fully connected network. Despite offering a fundamental understanding of image categorisation, these models' shallowness and insufficient ability to extract complicated features limited their potential to achieve higher accuracy.

We used transfer learning to address these issues, using a pretrained ResNet18 model that was initially trained on ImageNet. We greatly enhanced the model's performance by swapping out and retraining the last layer to fit the CIFAR-10 classification problem. Compared to models trained from scratch, the pretrained feature extraction layers provide a solid learning foundation, leading to quicker convergence and greater accuracy.

Following fine-tuning, the ResNet18 model produced a test accuracy of 82% and a validation accuracy of almost 83%, which is a significant improvement over the previous models' performance of roughly 73%.

The confusion matrix and classification report attest to the model's high generalisation abilities and consistent performance across several classes after thorough optimisation.

The significant benefits of transfer learning in contemporary deep learning workflows are demonstrated by this project. In addition to saving training time and processing resources, using pretrained architectures improves performance on challenging classification problems.

Overall, the findings show that transfer learning is a very successful method for solving image classification issues, even on relatively small datasets like CIFAR-10, when paired with the right fine-tuning and optimisation strategies.

5 References

[1] Krizhevsky, A. (2009). Learning Multiple Layers of Features from Tiny Images. University of Toronto. https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf

[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 (CVPR), 770–778. https://doi.org/10.1109/CVPR.2016.90

[3] Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., & He, Q. (2020). A Comprehensive Survey on Transfer Learning. Proceedings of the IEEE, 109(1), 43–76. https://doi.org/10.1109/JPROC.2020.3004555

[4] Kingma, D. P., & Ba, J. (2015). Adam: A Method for Stochastic Optimization. In Proceedings of the 3rd International Conference on Learning Representations (ICLR). https://arxiv.org/abs/1412.6980