INDIAN INSTITUTE OF SCIENCE

ELECTRONICS AND COMMUNICATION ENGINEERING DEPARTMENT

Digital Image Processing

Assignment 5

Ansh Bansal

24250

Date: November 18, 2025

Contents

1 Feature Extraction with a Pretrained Model

1.1 Classification using pretrained ResNet50

Objective

To load a pretrained CNN model for classification.

Implementation Summary

The ResNet-50 pretrained on ImageNet was loaded and all the layers were frozen.

1.2 Transfer Learning for `CIFAR-10`

Objective

To train the loaded ResNet-50 on `CIFAR-10`.

Implementation Summary

- The last fully-connected layer of original ResNet-50 was replaced by 10 dimensional fully-connected layer, since there are 10 classes in `CIFAR-10`.
- Trained the `model_clean` over `CIFAR-10` test-data.

1.3 Results of Transfer Learning

Results

Accuracy on `CIFAR-10` : 81.48%

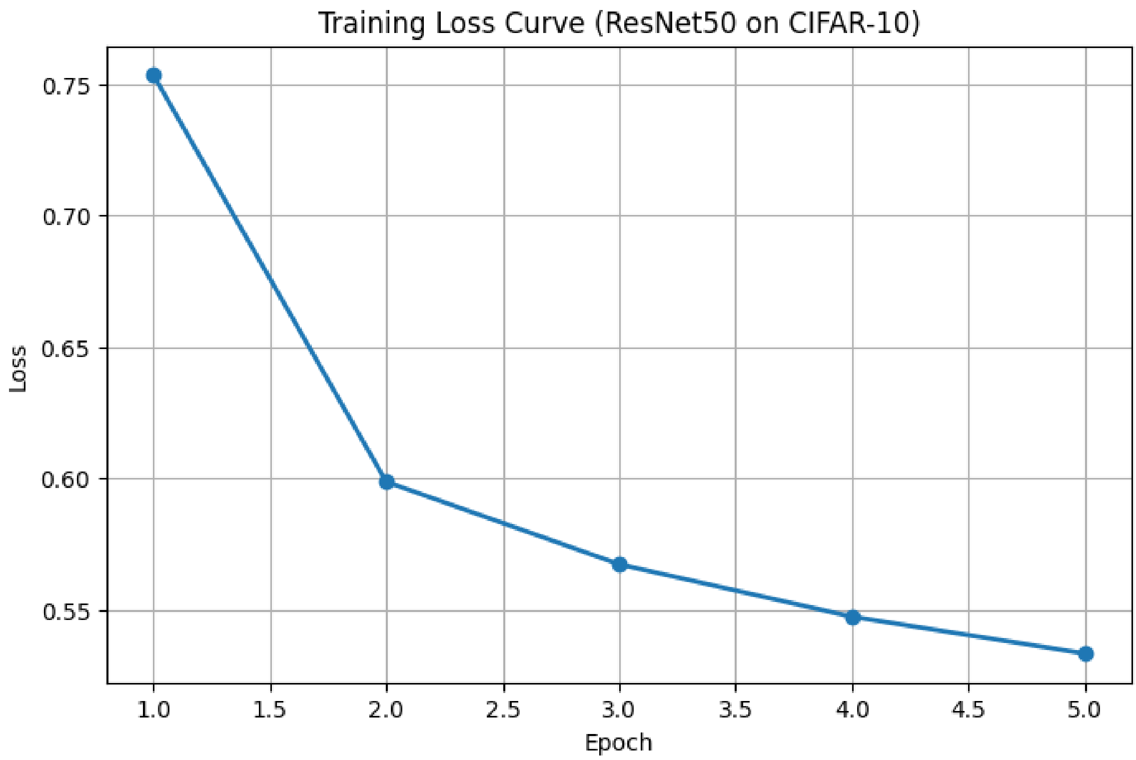


Figure 1.1: Training Loss Curve (`ResNet50` on `CIFAR-10`)

2 Domain Shift Evaluation and Fine-Tuning

2.1 Evaluation of Domain-Shifted Version of `CIFAR-10` (`CIFAR-10-C`)

Objective

To evaluate the performance of the pretrained **ResNet-50** model (trained on clean `CIFAR-10`) on a domain-shifted version of the dataset, **CIFAR-10-C**, which contains multiple corruption types.

Implementation Summary

- The **ResNet-50** model trained on clean `CIFAR-10` was evaluated on the corrupted `CIFAR-10-C` dataset to assess its robustness under domain shift.
- The `CIFAR-10-C` dataset includes various corruption types such as Gaussian noise, motion blur, fog, and contrast distortions.
- The test accuracy was calculated across all corruption types to analyze performance degradation.

Results

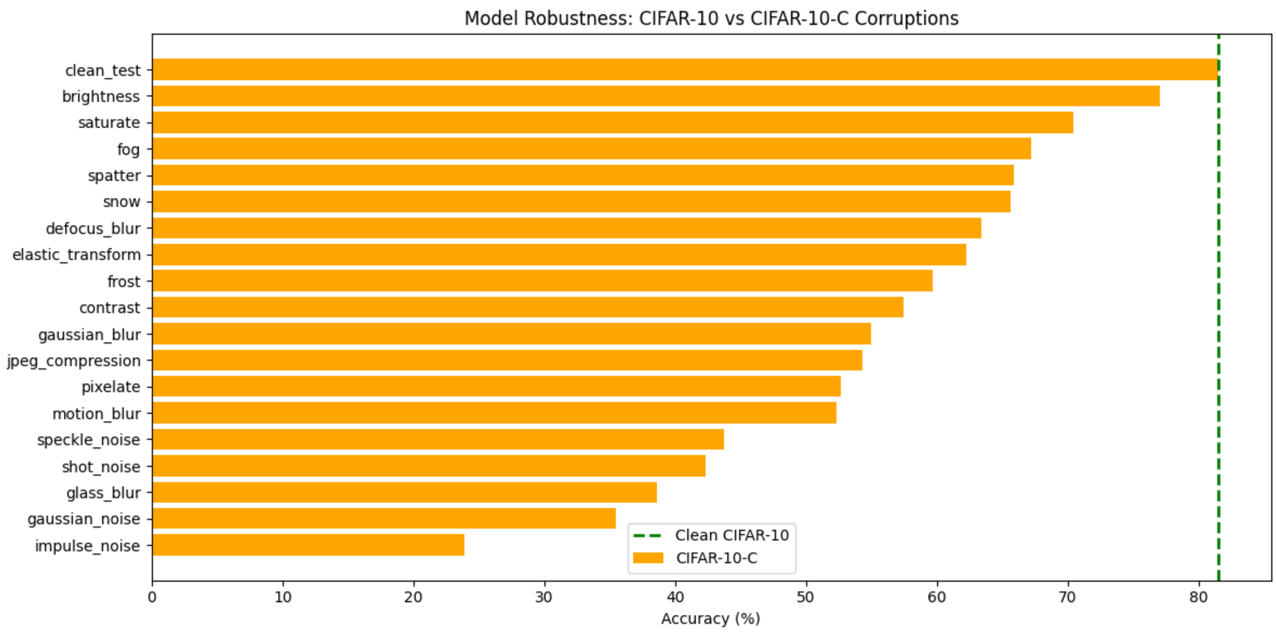


Figure 2.1: Plot of Accuracies of Clean Model on Individual Corruptions

The pretrained `model_clean` achieved an average accuracy of **56.25%** across all corruption types, showing a significant drop in performance compared to clean `CIFAR-10` due to the domain shift.

2.2 Fine-Tuning of ResNet-50 on `CIFAR-10-C`

Objective

To improve the model's robustness by fine-tuning the pretrained **ResNet-50** on a small labeled subset of the `CIFAR-10-C` dataset.

Implementation Summary

- The final convolutional block (`layer4`) and the fully connected layer of **ResNet-50** were **unfrozen**, while all earlier layers remained frozen.
- A small subset of *CIFAR-10-C*, containing samples from multiple corruption types, was used for partial fine-tuning. The number of samples selected from each corruption type was determined using the following relation:

$$\text{sample\_size} = \max\left(N \times \frac{(100 - \text{clean\_model\_acc\_results}[corr])}{1000},\ \text{subset\_size}\right)$$

  where N denotes the total number of samples available for a given corruption type, and clean_model_acc_results$[corr]$ represents the accuracy of the clean model on that corruption.

- Optimization was performed using the **Adam** optimizer with a small learning rate to avoid overfitting.

Results

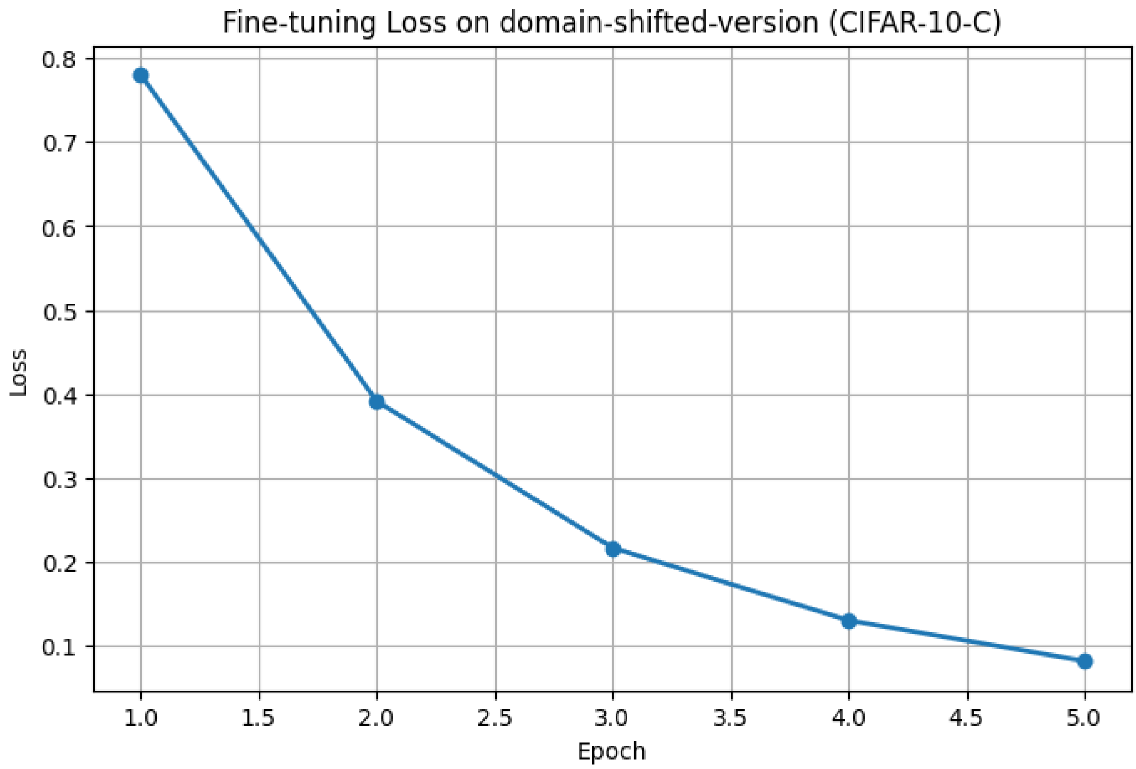


Figure 2.2: Training Loss Curve (ResNet50 on CIFAR-10-C)

2.3 Results of Fine-Tuning

Objective

To evaluate how partial fine-tuning affected performance under domain shift.

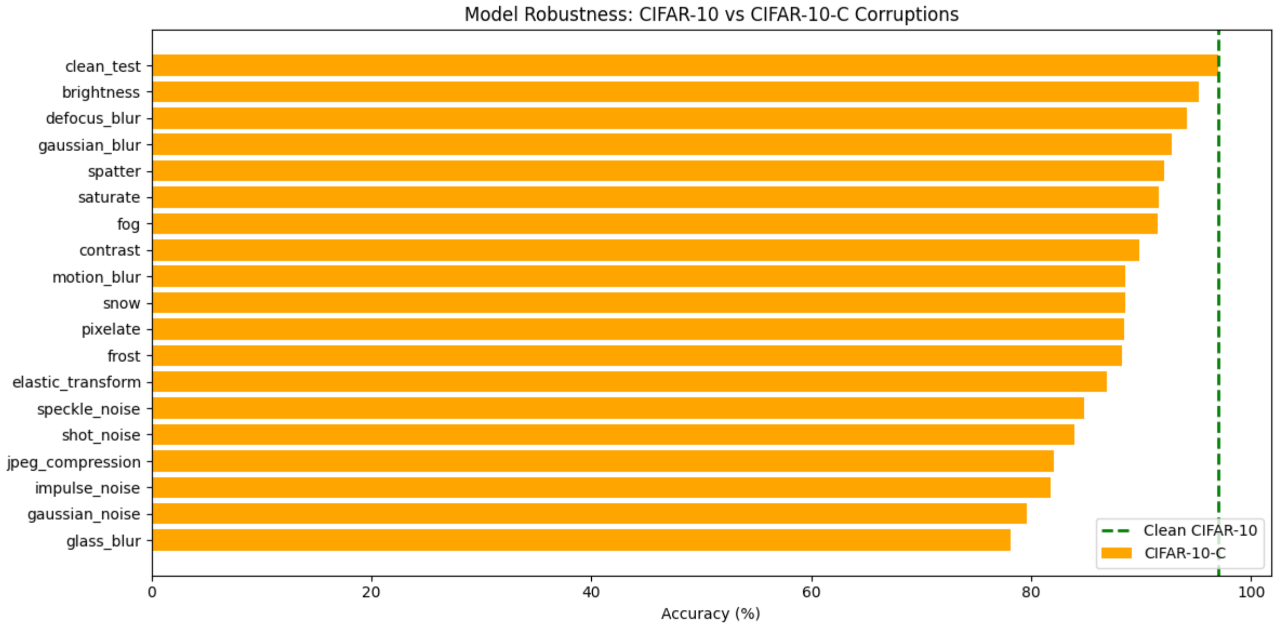


Figure 2.3: Plot of Accuracies of Fine-Tuned Model on Individual Corruptions

Results

Table 1: Comparison of Clean and Fine-tuned ResNet-50 Models on CIFAR-10-C

Corruption Type	Clean Model (%)	Fine-tuned Model (%)	Improvement (%)
clean_test	81.48	96.98	15.50
brightness	77.00	95.26	18.25
contrast	57.41	89.85	32.44
defocus_blur	63.42	94.16	30.74
elastic_transform	62.23	86.87	24.64
fog	67.20	91.52	24.32
frost	59.70	88.27	28.57
gaussian_blur	54.98	92.77	37.79
gaussian_noise	35.50	79.58	44.08
glass_blur	38.64	78.11	39.47
impulse_noise	23.93	81.72	57.80
jpeg_compression	54.28	82.08	27.80
motion_blur	52.34	88.55	36.22
pixelate	52.66	88.49	35.84
saturate	70.47	91.62	21.15
shot_noise	42.35	83.92	41.57
snow	65.68	88.54	22.86
spatter	65.87	92.12	26.26
speckle_noise	43.71	84.77	41.06
Average	**56.25**	**88.12**	**31.87**

Inference

After fine-tuning, the model `model_finetuned` achieved an average accuracy of **88.17%** on `CIFAR-10-C`, marking an improvement of $\Delta = \mathbf{88.17\%} - \mathbf{56.25\%} = \mathbf{31.87\%}$ over the frozen feature extractor model. This demonstrates that **partial fine-tuning** enables the model to adapt to new corruptions by updating higher-level features, thereby improving robustness while retaining generalization.

3 Feature Representation Analysis

3.1 Feature Extraction from Transfer Learned Model and Finetuned Model

Objective

To extract and compare the feature representations from the penultimate layer (before the final fully-connected classifier) of the pretrained **ResNet-50** model both before and after fine-tuning.

Implementation Summary

- Features were extracted from the **penultimate layer** (average pooling layer) of the `model.clean` (trained on clean *CIFAR-10*) and `model.finetuned` (fine-tuned on *CIFAR-10-C*).
- The extracted features are 2048-dimensional vectors representing high-level semantic embeddings for each image.
- Feature extraction was performed on both **source (CIFAR-10)** and **domain-shifted (CIFAR-10-C)** datasets.

Results

The extracted features were used to analyze how the model's internal representation of images changes after fine-tuning on the domain-shifted dataset.

3.2 Visualization of Feature Distribution using PCA

Objective

To visualize and compare the feature distributions of the transfer-learned and fine-tuned models using **Principal Component Analysis (PCA)**.

Implementation Summary

- PCA was applied to reduce the 2048-dimensional feature vectors to two principal components for visualization.
- Two sets of PCA plots were generated:
    1. For **source images (CIFAR-10)**.
    2. For **domain-shifted images (CIFAR-10-C)**.
- Each point in the PCA plot represents a single image, color-coded by its true class label.

Results

PCA visualizations revealed distinct clusters corresponding to individual classes. The distribution of features became more compact and well-separated after fine-tuning, particularly for corrupted images, indicating better feature discrimination under domain shift.

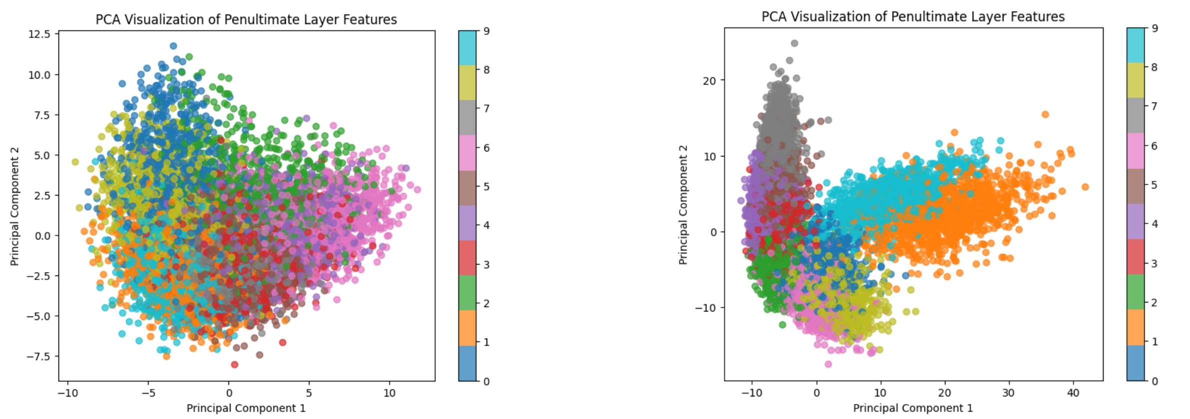
3.3 Interpretation of Feature Distribution

Objective

To interpret how fine-tuning affects the organization of the learned feature space under domain shift.

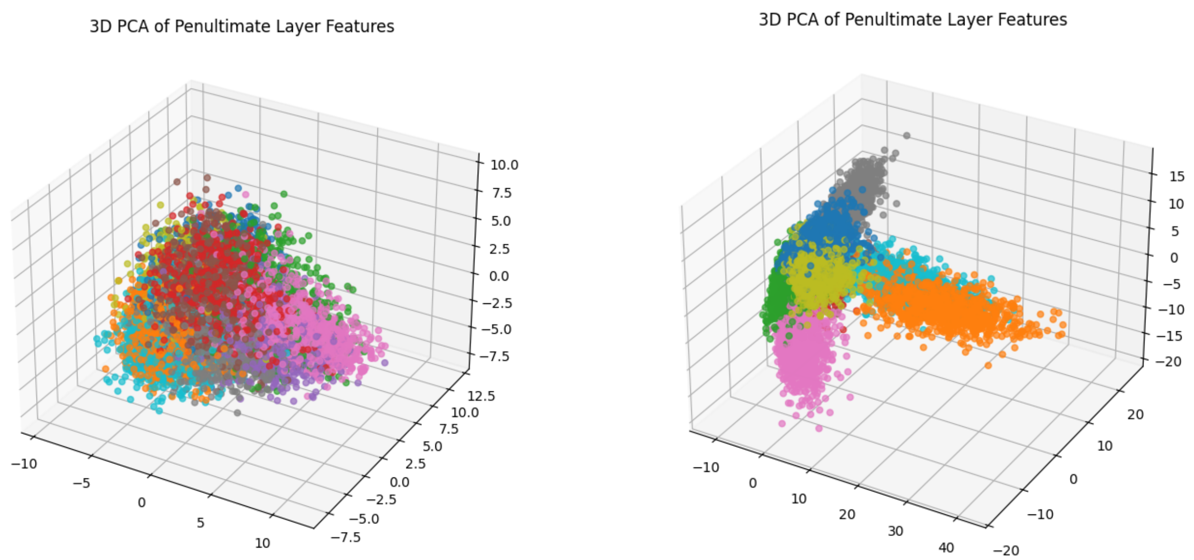
Observations

- In the **transfer-learned model**, the feature clusters for domain-shifted images (CIFAR-10-C) were more dispersed and overlapped between classes, indicating sensitivity to corruption.
- After **fine-tuning**, the clusters became tighter and more distinct, suggesting that the model adapted its high-level representations to better separate corrupted samples.
- The PCA plots show that feature variance between classes increased, while intra-class variance decreased after fine-tuning.



(a) Feature distribution from `model_clean`. **(b)** Feature distribution from `model_finetuned`.

Figure 3.1: Comparison of PCA visualizations in 2-D before and after fine-tuning on CIFAR-10-C.



(a) Feature distribution from `model_clean`. **(b)** Feature distribution from `model_finetuned`.

Figure 3.2: Comparison of PCA visualizations in 3-D before and after fine-tuning on CIFAR-10-C.

Conclusion

Fine-tuning helped align the feature distributions of source and corrupted domains, leading to more robust representations.

3.4 Inference of Fine-tuning

Objective

To summarize the effect of fine-tuning on the model's robustness and feature stability.

Inference

- Fine-tuning improved robustness to domain shift by adapting the last convolutional block to better capture corrupted visual patterns.
- This adaptation resulted in more stable and separable feature embeddings for corrupted images, as reflected in the PCA visualizations.
- The model's improved performance on *CIFAR-10-C* confirms that limited fine-tuning of higher layers can significantly mitigate the effects of domain shift while preserving generalization from the source domain.