Digital Image Processing
Assignment 5

Tejash More
M.Tech SP

Packages used: NumPy, Pillow, Matplotlib, sklearn, torch, torchvision, tqdm, copy

1. Feature Extraction with a pre-trained model

a. Extracting features from a pre-trained model

- Pre-trained model: ResNet50 (trained on ImageNet)
- Batch size: 64
- Preprocessing: Upsample to (224, 224) from (32, 32), which is standard for ImageNet
- Features are the vectors that are the output of the CNN. This vector contains all the essential features of the image. This vector is then passed to the Fully Connected layer and is used to classify the image.
- What is done here is that this feature vector has been extracted for all the images in the CIFAR-10 dataset.

b. Training a simple classifier on these extracted features

- Loss Function: Cross-Entropy
- Optimizer: Adam
- Learning Rate: 0.001
- Epoch: 30
- ResNet50 has its own learned Fully Connected Layer, which is trained on the ImageNet dataset. Since we are using the CIFAR-10 Dataset. We removed this FC layer and trained our own FC layer. This is called Transfer Learning.

```
100%|██████████| 782/782 [00:01<00:00, 470.59it/s]
Epoch: 1/30 | Train Loss: 0.835 | Train Acc: 0.737 | Test Acc: 0.781
100%|██████████| 782/782 [00:01<00:00, 565.99it/s]
Epoch: 2/30 | Train Loss: 0.595 | Train Acc: 0.803 | Test Acc: 0.795
100%|██████████| 782/782 [00:01<00:00, 531.41it/s]
Epoch: 3/30 | Train Loss: 0.534 | Train Acc: 0.821 | Test Acc: 0.801
100%|██████████| 782/782 [00:01<00:00, 508.24it/s]
Epoch: 4/30 | Train Loss: 0.498 | Train Acc: 0.833 | Test Acc: 0.804
100%|██████████| 782/782 [00:01<00:00, 477.37it/s]
Epoch: 5/30 | Train Loss: 0.472 | Train Acc: 0.840 | Test Acc: 0.807
100%|██████████| 782/782 [00:01<00:00, 484.94it/s]
Epoch: 6/30 | Train Loss: 0.453 | Train Acc: 0.847 | Test Acc: 0.803
100%|██████████| 782/782 [00:01<00:00, 508.38it/s]
Epoch: 7/30 | Train Loss: 0.437 | Train Acc: 0.853 | Test Acc: 0.807
100%|██████████| 782/782 [00:01<00:00, 510.80it/s]
Epoch: 8/30 | Train Loss: 0.425 | Train Acc: 0.856 | Test Acc: 0.802
100%|██████████| 782/782 [00:01<00:00, 470.85it/s]
Epoch: 9/30 | Train Loss: 0.414 | Train Acc: 0.858 | Test Acc: 0.797
100%|██████████| 782/782 [00:01<00:00, 484.35it/s]
Epoch: 10/30 | Train Loss: 0.405 | Train Acc: 0.862 | Test Acc: 0.802
100%|██████████| 782/782 [00:01<00:00, 508.54it/s]
Epoch: 11/30 | Train Loss: 0.397 | Train Acc: 0.864 | Test Acc: 0.804
100%|██████████| 782/782 [00:01<00:00, 504.18it/s]
Epoch: 12/30 | Train Loss: 0.390 | Train Acc: 0.866 | Test Acc: 0.800
100%|██████████| 782/782 [00:01<00:00, 504.83it/s]
Epoch: 13/30 | Train Loss: 0.383 | Train Acc: 0.870 | Test Acc: 0.799
100%|██████████| 782/782 [00:01<00:00, 518.77it/s]
Epoch: 14/30 | Train Loss: 0.377 | Train Acc: 0.871 | Test Acc: 0.799
100%|██████████| 782/782 [00:01<00:00, 470.80it/s]
Epoch: 15/30 | Train Loss: 0.373 | Train Acc: 0.873 | Test Acc: 0.799
```

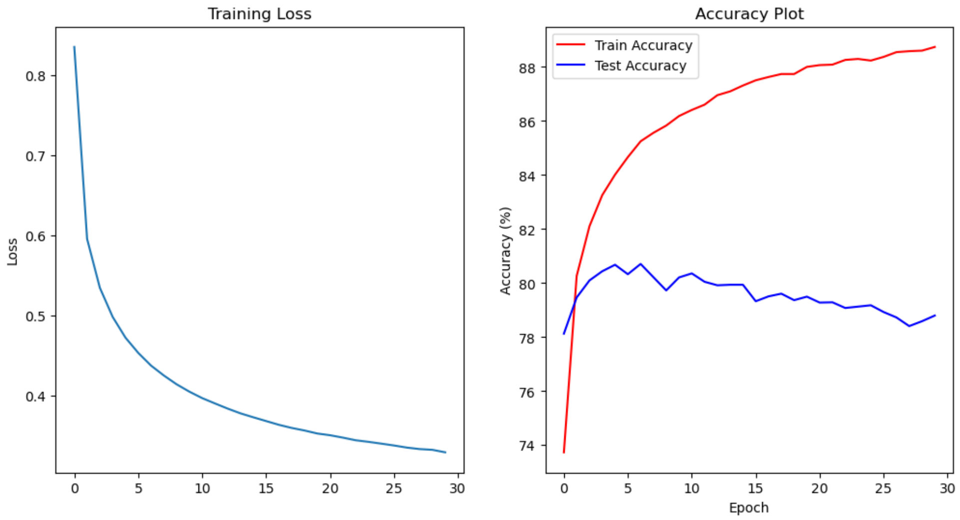
```
100%|██████████| 782/782 [00:01<00:00, 470.80it/s]
Epoch: 15/30 | Train Loss: 0.373 | Train Acc: 0.873 | Test Acc: 0.799
100%|██████████| 782/782 [00:01<00:00, 510.12it/s]
Epoch: 16/30 | Train Loss: 0.368 | Train Acc: 0.875 | Test Acc: 0.793
100%|██████████| 782/782 [00:01<00:00, 476.83it/s]
Epoch: 17/30 | Train Loss: 0.363 | Train Acc: 0.876 | Test Acc: 0.795
100%|██████████| 782/782 [00:01<00:00, 500.93it/s]
Epoch: 18/30 | Train Loss: 0.359 | Train Acc: 0.877 | Test Acc: 0.796
100%|██████████| 782/782 [00:01<00:00, 483.80it/s]
Epoch: 19/30 | Train Loss: 0.356 | Train Acc: 0.877 | Test Acc: 0.794
100%|██████████| 782/782 [00:01<00:00, 526.73it/s]
Epoch: 20/30 | Train Loss: 0.352 | Train Acc: 0.880 | Test Acc: 0.795
100%|██████████| 782/782 [00:01<00:00, 501.66it/s]
Epoch: 21/30 | Train Loss: 0.350 | Train Acc: 0.881 | Test Acc: 0.793
100%|██████████| 782/782 [00:01<00:00, 468.77it/s]
Epoch: 22/30 | Train Loss: 0.347 | Train Acc: 0.881 | Test Acc: 0.793
100%|██████████| 782/782 [00:01<00:00, 521.21it/s]
Epoch: 23/30 | Train Loss: 0.344 | Train Acc: 0.883 | Test Acc: 0.791
100%|██████████| 782/782 [00:01<00:00, 448.49it/s]
Epoch: 24/30 | Train Loss: 0.342 | Train Acc: 0.883 | Test Acc: 0.791
100%|██████████| 782/782 [00:01<00:00, 489.11it/s]
Epoch: 25/30 | Train Loss: 0.340 | Train Acc: 0.882 | Test Acc: 0.792
100%|██████████| 782/782 [00:01<00:00, 537.82it/s]
Epoch: 26/30 | Train Loss: 0.337 | Train Acc: 0.884 | Test Acc: 0.789
100%|██████████| 782/782 [00:01<00:00, 524.52it/s]
Epoch: 27/30 | Train Loss: 0.335 | Train Acc: 0.885 | Test Acc: 0.787
100%|██████████| 782/782 [00:01<00:00, 514.44it/s]
Epoch: 28/30 | Train Loss: 0.333 | Train Acc: 0.886 | Test Acc: 0.784
100%|██████████| 782/782 [00:01<00:00, 508.91it/s]
Epoch: 29/30 | Train Loss: 0.332 | Train Acc: 0.886 | Test Acc: 0.786
100%|██████████| 782/782 [00:01<00:00, 465.40it/s]
Epoch: 30/30 | Train Loss: 0.329 | Train Acc: 0.887 | Test Acc: 0.788
```

c. Report Classification Accuracy and Plot the Loss Curve

- After 30 Epochs
  - Classification Accuracy on Test Data was 78.8%
  - Train Loss: 0.329

Digital Image Processing
Assignment 5

Tejash More
M.Tech SP



- **Observation:**
  - Loss Curve: From the plot, it can be seen that each epoch model successfully learned to decrease the overall loss, minimizing the error.
  - Accuracy Plot: Train Accuracy continued to increase each epoch, whereas test accuracy started decreasing from epoch five onwards. This indicates that although the model continued to learn the training data pattern, it was unable to generalize the pattern to classify the test images correctly. This is a sign of overfitting.

The reason could be that the CNN block of ResNet50 is trained on high-resolution images of ImageNet (224, 224), but the CIFAR-10 dataset has images of shape (32, 32). Upsampling these small images to (224, 224) will not retain many high-textured details, which is why the model is unable to extract the textured details of the CIFAR-10 dataset as effectively.

2. Domain Shift Evaluation and Fine-Tuning

a. Evaluating the previously trained model on the CIFAR-10-C dataset

- CIFAR-10-C dataset has 19 types of corruptions added to the image, each corruption is a separate numpy file.

Brightness	Contrast	Defocus_blur	Elastic_transform
Fog	Frost	Gaussian_blur	Gaussian_noise
Glass_blur	Impulse_noise	Jpeg_compression	Motion_blur
Pixelate	Saturate	Shot_noise	Snow
Spatter	Speckle_noise	Zoom_blur	NA

- Each file has 50,000 images with 10,000 unique images, but corrupted on five scales (level 1 to 5). A separate file ‘labels.npy’ contains the labels of these 50,000 images.
- Corruption used here: gaussian_blur
- Batch size: 64
- Accuracy obtained from our model: 37.64%

The reason could be that, even before, the CNN block of ResNet50 was unable to extract the highly textured details due to the images being so small. Now, even those small images are Gaussian blurred. Upsampling these images will not retain almost any of the highly textured details. Due to this, Accuracy dropped so dramatically.

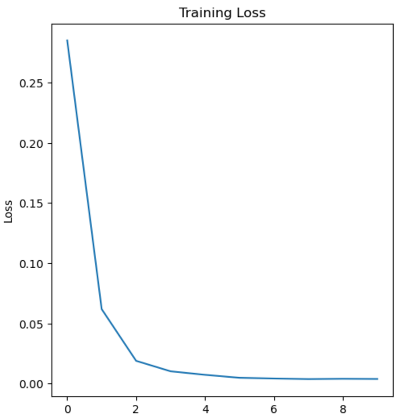
Digital Image Processing
Assignment 5

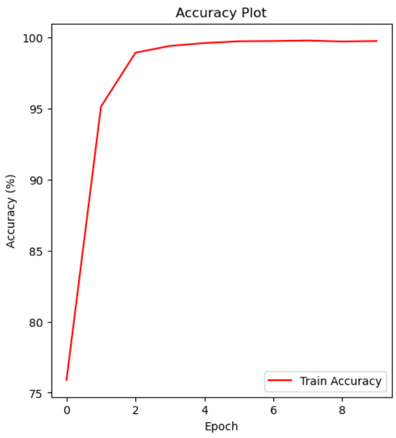
Tejash More
M.Tech SP

b. Fine-Tune the Pre-trained Model

- Fine-tuning has been done on the subset of the images from the CIFAR-10-C dataset, containing 20,000 images, randomly chosen from 50,000 images.
- Parameters will be updated for the Last layer (Layer 4) of the CNN Block and the FC layer.
- Learning Rate (layer 4): 0.0001
- Learning Rate (FC Layer): 0.001
- Loss Function: Cross-Entropy Loss
- Optimizer: Adam
- Epoch: 10

```
100%|██████████| 313/313 [01:39<00:00,  3.15it/s]
Epoch: 1/10 | Train Loss: 0.285 | Train Acc: 0.759
100%|██████████| 313/313 [01:41<00:00,  3.08it/s]
Epoch: 2/10 | Train Loss: 0.062 | Train Acc: 0.951
100%|██████████| 313/313 [01:42<00:00,  3.05it/s]
Epoch: 3/10 | Train Loss: 0.019 | Train Acc: 0.989
100%|██████████| 313/313 [01:39<00:00,  3.14it/s]
Epoch: 4/10 | Train Loss: 0.010 | Train Acc: 0.994
100%|██████████| 313/313 [01:41<00:00,  3.09it/s]
Epoch: 5/10 | Train Loss: 0.007 | Train Acc: 0.996
100%|██████████| 313/313 [01:41<00:00,  3.08it/s]
Epoch: 6/10 | Train Loss: 0.005 | Train Acc: 0.997
100%|██████████| 313/313 [01:38<00:00,  3.18it/s]
Epoch: 7/10 | Train Loss: 0.004 | Train Acc: 0.998
100%|██████████| 313/313 [01:40<00:00,  3.13it/s]
Epoch: 8/10 | Train Loss: 0.004 | Train Acc: 0.998
100%|██████████| 313/313 [01:39<00:00,  3.16it/s]
Epoch: 9/10 | Train Loss: 0.004 | Train Acc: 0.997
100%|██████████| 313/313 [01:39<00:00,  3.16it/s]
Epoch: 10/10 | Train Loss: 0.004 | Train Acc: 0.998
```





- After 10 Epochs:
  - Loss: 0.004
  - Train Accuracy: 99.8%

c. Evaluate the fine-tuned Model again on the CIFAR-10-C dataset

- This fine-tuned model is evaluated on both the CIFAR-10 and CIFAR-10-C datasets
  - Accuracy on CIFAR-10-C: 94.25% (jump from 37.64%)
  - Accuracy on CIFAR-10: 90.79% (jump from 78.8%)

Observation:

Accuracy on both datasets has increased significantly. This confirmed our assumption that the ResNet50's CNN blocks were proficient in extracting high-textured details from images, which are absent in our dataset, resulting in poor performance. By modifying the parameters of the last layer of the CNN block with the corrupted images of CIFAR-10-C. The model learned not to rely on the high-textured details but to rely on the overall smooth structure of the image. This made the model more robust, and as a result, it also performed better on the clean images of the CIFAR-10 Dataset.

3. Feature Representation Analysis

a. Extract Features from the Fine-tuned Model

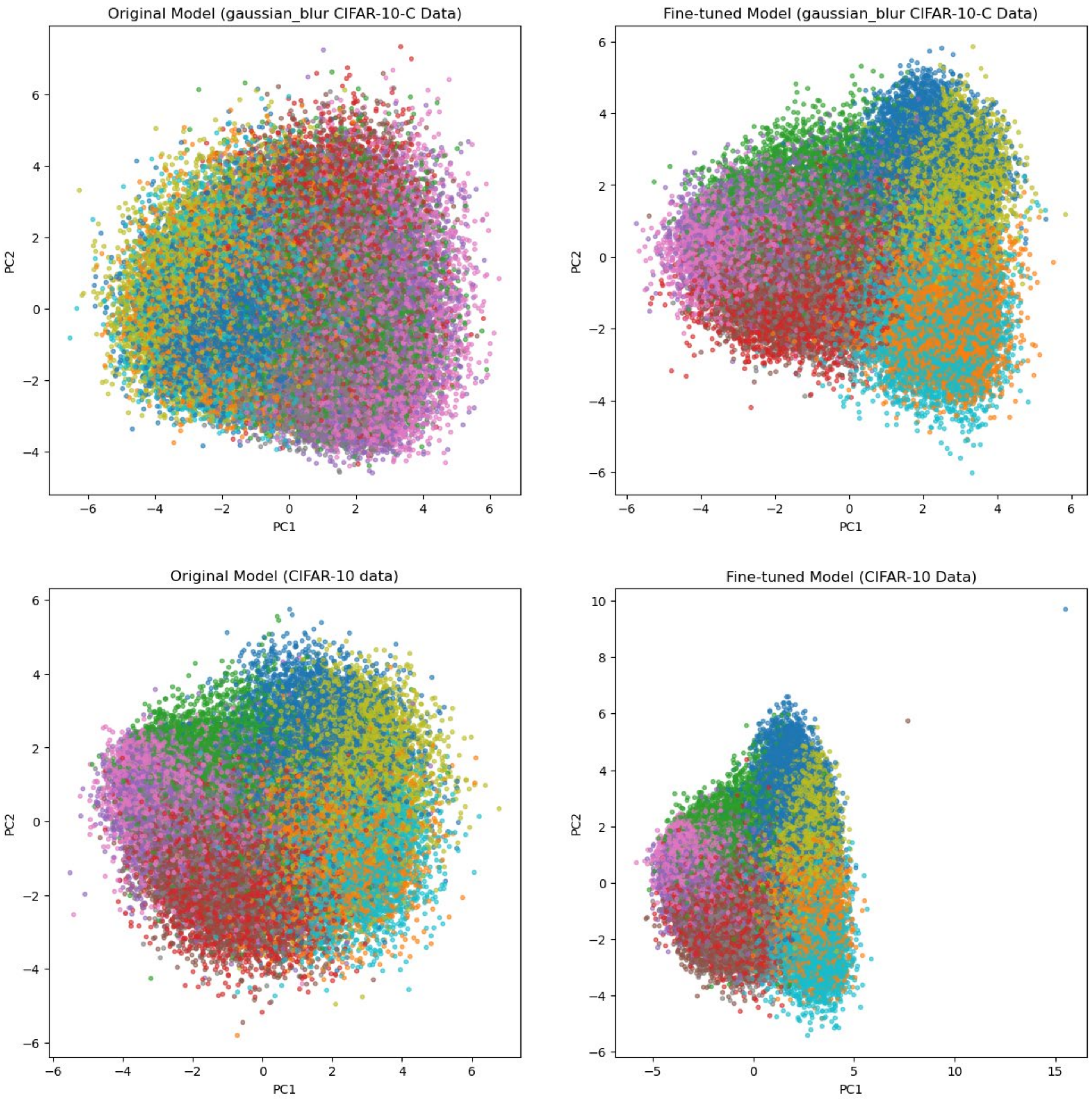
- Features have been extracted from this fine-tuned model in the same way as it was extracted from the Original ResNet50 model in Question 1.
- Four Features have been extracted:
  - Feature vectors of the CIFAR-10 dataset using the Original ResNet50 model
  - Feature vectors of the CIFAR-10 dataset using the Fine-tuned ResNet50 Model
  - Feature vectors of the CIFAR-10-C dataset using the Original ResNet50 Model

Digital Image Processing
Assignment 5

Tejash More
M.Tech SP

- Feature vectors of the CIFAR-10-C dataset using the Fine-tuned ResNet50 Model.

b. Use PCA to visualize the features



- Top-left plot (CIFAR-10-C dataset using Original ResNet50)
  - Feature vectors of different classes are heavily overlapped with each other. This explains that the CNN block of the original model was unable to correctly distinguish between the different classes, resulting in a low accuracy of 37%

Digital Image Processing
Assignment 5

Tejash More
M.Tech SP

- Top-right plot (CIFAR-10-C dataset using our Fine-tuned ResNet50)
  - Unlike the previous plot, this plot has distinct clusters with visible classes. This proves that the modified parameter of the last CNN block of our model is not relying on those high-textured details in the image anymore, and it learned to differentiate the classes based on the robust features of the image, therefore increasing the accuracy to 94%

- Bottom-left plot (CIFAR-10 dataset using the original ResNet50)
  - The clusters are seen segregated here, but their edges are not clearly defined. This indicates that the model, on average, will be able to distinguish between the classes; however, a significant number of images will be misclassified by the model as a neighboring class, especially for those images which has a feature vector near the center of this huge blob.

- Bottom-right plot (CIFAR-10 dataset using our fine-tuned ResNet50)
  - It has already been proven that our fine-tuned model relies on the robust features of the images to classify them. On the clean images of the CIFAR-10 dataset, it can classify them almost perfectly into their correct class.
  - The overall blob is very dense; that is, the intra-class points are very closely packed with each other. This means that if two feature vectors have even a little distance between them, it is more likely that they belong to different classes. This low margin of error made the model as accurate as possible in determining the class, increasing the accuracy to 90%
  - Another observation is that the center where there is the most overlap of the classes has less variance than the plot in the Bottom-left (Original Model). This lower variance means that not many images fall into that 'grey' zone, reducing the False predictions.