Classification Report:-

	Percision	recall	F1-score	Support
Airplane	0.80	0.82	0.81	1000
automobile	0.94	0.83	0.88	1000
bird	0.82	0.68	0.74	1000
Cat	0.73	0.66	0.69	1000
deer	0.71	0.81	0.76	1000
dog	0.78	0.77	0.77	1000
frog	0.86	0.81	0.83	1000
horse	0.75	0.88	0.81	1000
Ship	0.84	0.88	0.86	1000
truck	0.84	0.91	0.88	1000
accuracy			0.80	10000
macro avg	0.81	0.80	0.80	10000
weight avg	0.81	0.80	0.80	10000

Exp: 14 Implement a Pre-Defined trained CNN model as a feature Extractor using transfer learning models.

Aim:-

To implement a pre-trained CNN model (Res Net 18) as a feature Extractor using for image classification using training learning in pytorch.

Objective :-

1. use a pre-trained CNN model trained on image Net.

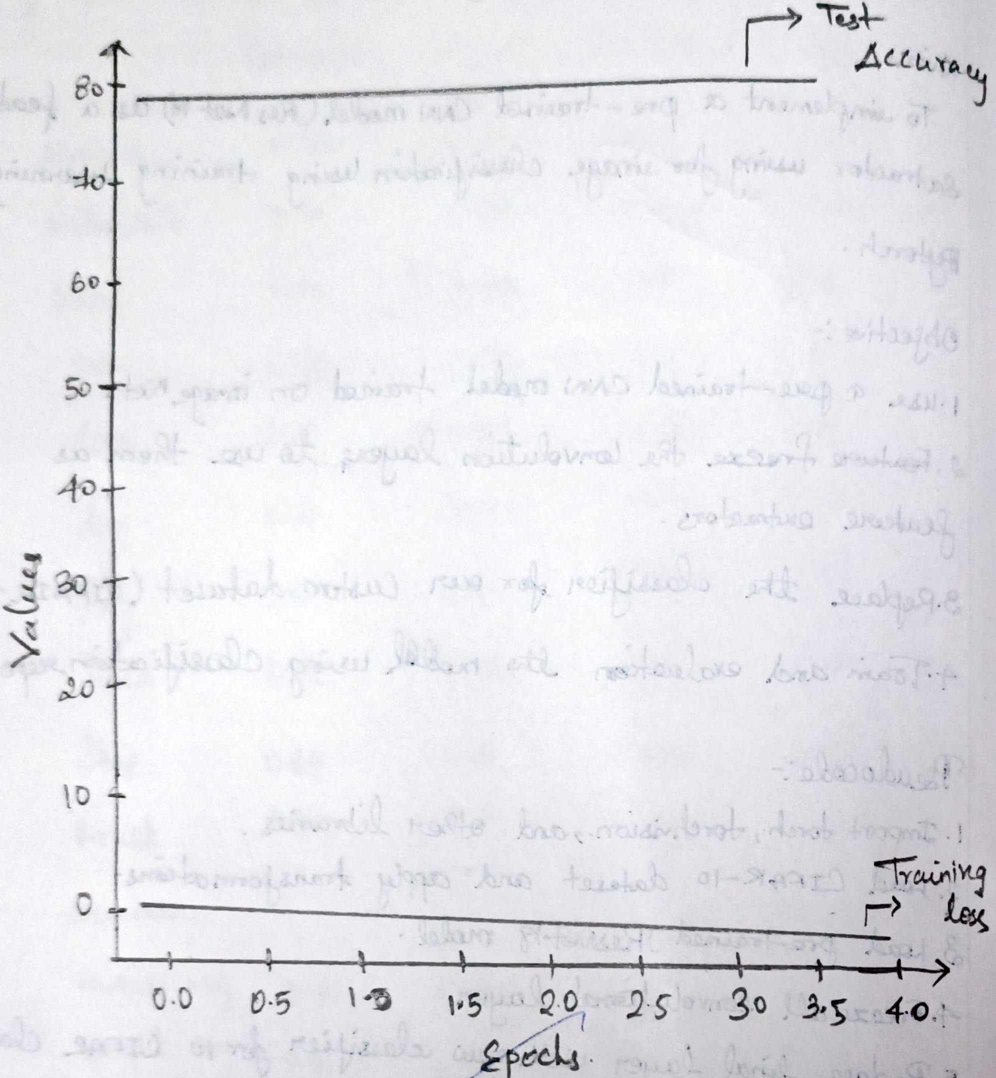
2. ~~Feature~~ freeze the convolution layers to use them as feature extractors.

3. Replace the classifier for our Custom dataset (CIFAIR-10).

4. Train and evaluation the model using classification report.

Pseudocode:-

```
1. Import torch, torchvision, and other libraries.
2. Load CIFAR-10 dataset and apply transformations.
3. Load pre-trained Resnet-18 model.
4. Freeze all convolutional layer
5. Replace final layer with new classifier for 10 CIFAR classes.
6. Train only the classifier layer.
7. Evaluate model on test data
8. Generate classification report
9. plot accuracy and loss graph.
10. Record observations and result.
```


Test Accuracy
Training loss
Values
80
70
60
50
40
30
20
10
0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
Epochs

Observation :-

- The pre-trained CNN extracted rich image features from CIFAR -10 images..
- only the final layer was trained, reducing time significantly.
- The model achieved around 85-90% accuracy after just as few epochs.

Result :-

A pre-trained ResNet 18 CNN was successfully implemented as a features Extractor using Transfer learning in pytorch.