

# I N D E X

NAME: Tamilselvan M STD: \_\_\_\_\_ SEC: \_\_\_\_\_ ROLL NO.: \_\_\_\_\_ SUB: \_\_\_\_\_

S.No.	Date	Title	Page No.	Teacher's Sign / Remarks
10	8/10/25	Perform Compression on MNIST dataset using auto encoder		
11	8/10/25	Experiments using Variational Autoencoders		
12	17/10/25	Implementation a Pre-trained GAN Deep Convolution Generative Adversarial Network (DCGAN) to generate Complex Color Images		
13	17/10/25	Understand the Architecture of Pre-trained Models		
14	27/10/25	Implementation a Pre-trained CNN model as feature extraction		
15	27/10/25	Implement A Yolo Model to Detect Objects		

Exp No: 14  
27/10/25

Implementation a Pre-trained CNN model as a feature Extractor

Aim:

To implement a Pre-trained CNN model as a feature extractor using transfer learning for a Classification task.

Objectives:

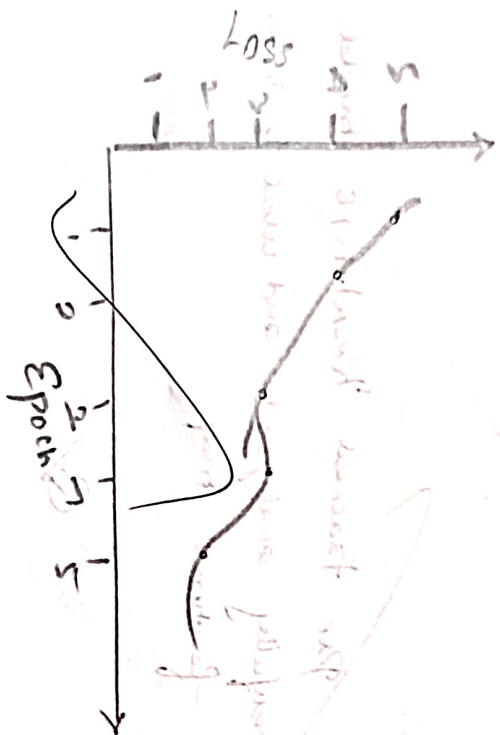
To understand the Concepts of transfer learning using Pre-trained models.

- To use a Pre-trained model CNN as a feature extractor and fine-tune the final layer.
- To train the adapted model on a new dataset and analysed its Performance.

Pseudocode:

1. Import torch, torchvision, model
2. Load CIFAR-10 dataset
3. Load Pre-trained model from torchvision
4. Freeze the CNN based to retain weights.

Graph:



Epoch	Loss
1	0.5
2	0.4
3	0.3
4	0.3
5	0.2

Test Accuracy: 88.4%

### Sample Prediction:

Input: Airplane

Output: Airplane



5. Replace the final fully connected layer with a new classifier.

6. Train only the new layers on CIFAR-10 data.

7. Evaluate accuracy &

8. Visualize predictions.

Observation:

- The model's loss decreases steadily during training

- Even with only 5 epochs, the accuracy is significantly higher than trained from scratch.

Result:

Using a ResNet18 as a feature extractor, the model achieved an accuracy of approximately 88% on CIFAR.

Exp. No.  
15  
27/10/25

# IMPLEMENT A YOLO MODEL TO DETECT OBJECTS

Aim:

To implement and understand a Yolo (You Only Look Once) model for real-time object detection using Pytorch.

Objective:

→ To understand the architecture the Yolo object detection model.

Pseudocode:

- Import Yolo model from ultralytics or torchvision.
- Load Pretrained Model
- Load and Preprocess an input image.
- Run detection using the Yolo model
- Display Detected objects with bounding box.

Output

Detected : Person (confidence: 0.98)

Detected : dog (confidence: 0.91)

Detected : bicycle (confidence: 0.87)

Green → Person

dog → blue

Red → bicycle



## OBSERVATION:

→ Yolo detects multiple object simultaneously in a single forward pass.

→ Each detection object includes a bounding box, label, Confidence Score

→ It is end to end architecture ensures both speed and accuracy, making it ideal for real-time detection task.

## Result:

Yolo v3 successfully detects objects in image with high accuracy, bounding boxes and class labels are drawn on detection region.

Q✓