

RA2311047010016

S.No	Date	Title	Signature
9.	30.09.2025	Building a RNN	30/9 write neatly
8	09.10.2025	Long Short Term Memory	9/10/25
10.	17.10.2025	perform compression using airist dataset using auto encoders	
11.	17.10.2025	Experiment using Variations (VAE)	
12	27.10.2025	Implement a Deep convolutional GATN to generate color Images	copy
13	27.10.2025	Understanding the Architecture of a pre-trained model	
14	27.10.2025	Implement a pre-trained CNN model as a feature extractor	
15	27.10.2025	Implement a Yolo model object detection	copy

Completed
3/11/25

27.10.2025

1+ Implement a pre-trained CNN model as a feature extractor.

Aim:

To use a pre-trained CNN model (e.g. ResNet 50, VGG16) as a feature extractor for a custom dataset

Objectives

* To understand transfer learning

* To reuse learned convolutional

filter from large datasets.

* To train only the final classification layer with minimal

computations.

- * Accuracy improved due to effective feature reuse
- * freezing the base layers prevented overfitting on small datasets

Result:

Transfer learning using a pre-trained CNN model successfully extracted features and improved classification performance efficiently.

Output

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

Test accuracy = 98.4 %.

Sample prediction:

Input : Airplane

Output : Airplane

pseudocode

- * Start
- * Load pre-trained conv model with weights = 'imagent' and exclude top layers.
- * Freeze all convolution layers.
- * Compile model (Adam + cross Entropy).
- * Train only the new to layers on custom dataset.
- * Evaluate model accuracy
- * End /.

Observation:

- * Training was significantly faster compared to training from scratch

