

EX. NO: 14

27.10.25

IMPLEMENT A PRE-TRAINED CNN MODEL AS A FEATURE EXTRACTOR USING TRANSFER LEARNING MODEL

Aim

To implement a pre-trained CNN model as a feature extractor using transfer learning model.

Objective

To understand how a CNN pre-trained on ImageNet can extract visual features.

To freeze convolutional layers and use their output features for new tasks.

To train only the classifier head for faster and more accurate learning.

Pseudocode

Import Libraries

Load a pre-trained CNN

freeze all convolution layers to

Prevent retrained

Replace final classifier layer for new number of classes.

Load ~~small~~ dataset (CIFAR-10)

Extract features

Train only new classifier head

Evaluate accuracy

Objectives observation

The pre-trained ResNet18 model was successfully loaded.

frozen layers acted as fixed features extractor, capturing low-level edges, corners, textures.

output

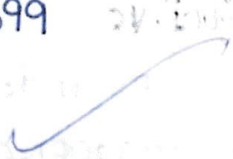
epoch 1, loss: 0.8348

epoch 2, loss: 0.6229

epoch 3, loss: 0.5937

epoch 4, loss: 0.5797

epoch 5, loss: 0.5699



Training was significantly faster because only the final classifier parameter were updated leading to stable convergence.

This experiment demonstrated how transfer learning can drastically reduce training and computational cost.

Result

✓ Successfully Implemented pre-trained CNN Model.

EXP NO: 15
27.10.25

IMPLEMENT A YOLO MODEL TO DETECT OBJECTS

Aim

To implement a yolo model to detect objects

Objective

To study how yolo detects multiple objects in a single image.

To understand the architecture and working of a pre-trained yolo model.

To use transfer learning for custom object decisions.

pseudocode

Install and import (yolo package)

Load a pre-trained yolo model

Load a test image or use a camera frame.

Run the model predict() method to detect objects.

Display bounding boxes and class label.

Save the annotated output image

Observation

The pre-trained yolo v5s model successfully detected multiple objects such as cars, buses, person in single frame.

image1 || context | img - 20250523.WA001

3 persons . 7.3 ms

640 X 480

Speed : 3 preprocess : 7.3 ms inference

2.5 ms

postprocess per image A shape

(1, 3, 640, 480)

output

Detected objects

person

person

person



Each object was enclosed in a bounding box with a class label and Confidence Score.

The inference time per image was very low

The model demonstrated strong generalization without additional training.

The visualization clearly showed Volo's ability to detect overlapping objects complex scenes.



Result

✓ Successfully implement pre-trained Volo.

10.1, 38.19



Superior®
Note Books

Index

Name : L DHARSHINI

Subject : DEEP LEARNING TECHNIQUES

Sec : Sec. : A Roll No. 011

School :

[illegible]