

16/09/2025

LAB-7

Image classification
using CNN.

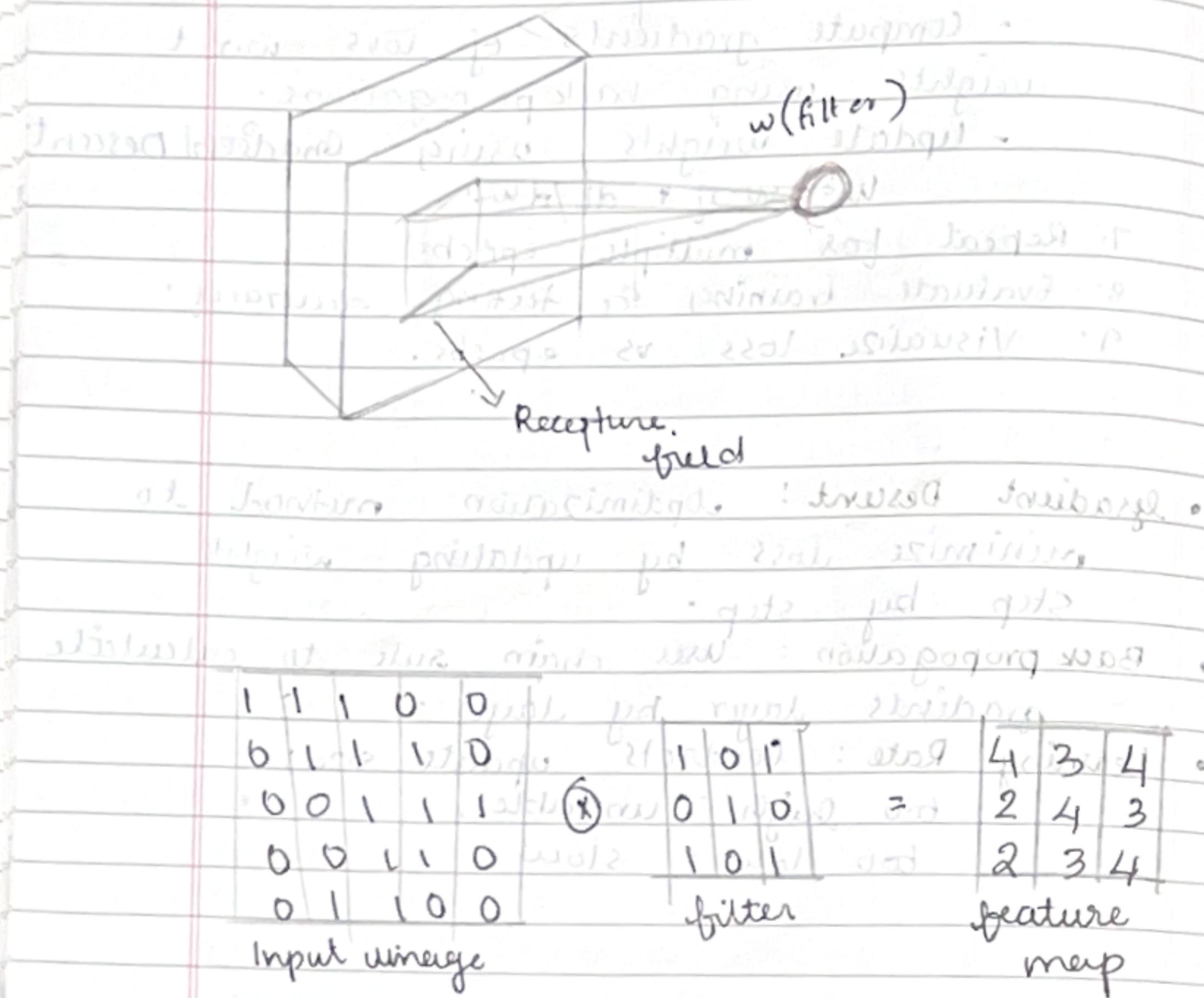
AIM:-

To implement multi-class image classification system using CNN.

OBJECTIVES:-

- > Understand the structures and working of CNN.
- > Build a custom CNN architecture for image classification.
- > Train the CNN on a multi-class dataset (CIFAR-10).
- > Evaluate the model's performance using loss and accuracy metrics.
- > Visualize feature maps to interpret the features learned at each convolutional layer.
- > Plot training loss and accuracy curves to monitor learning.

O.P. Convolutional Neural Network is a "deep learning" model that automatically learns and extracts features from images using convolutional convolution and pooling layers for tasks like classification and recognition.

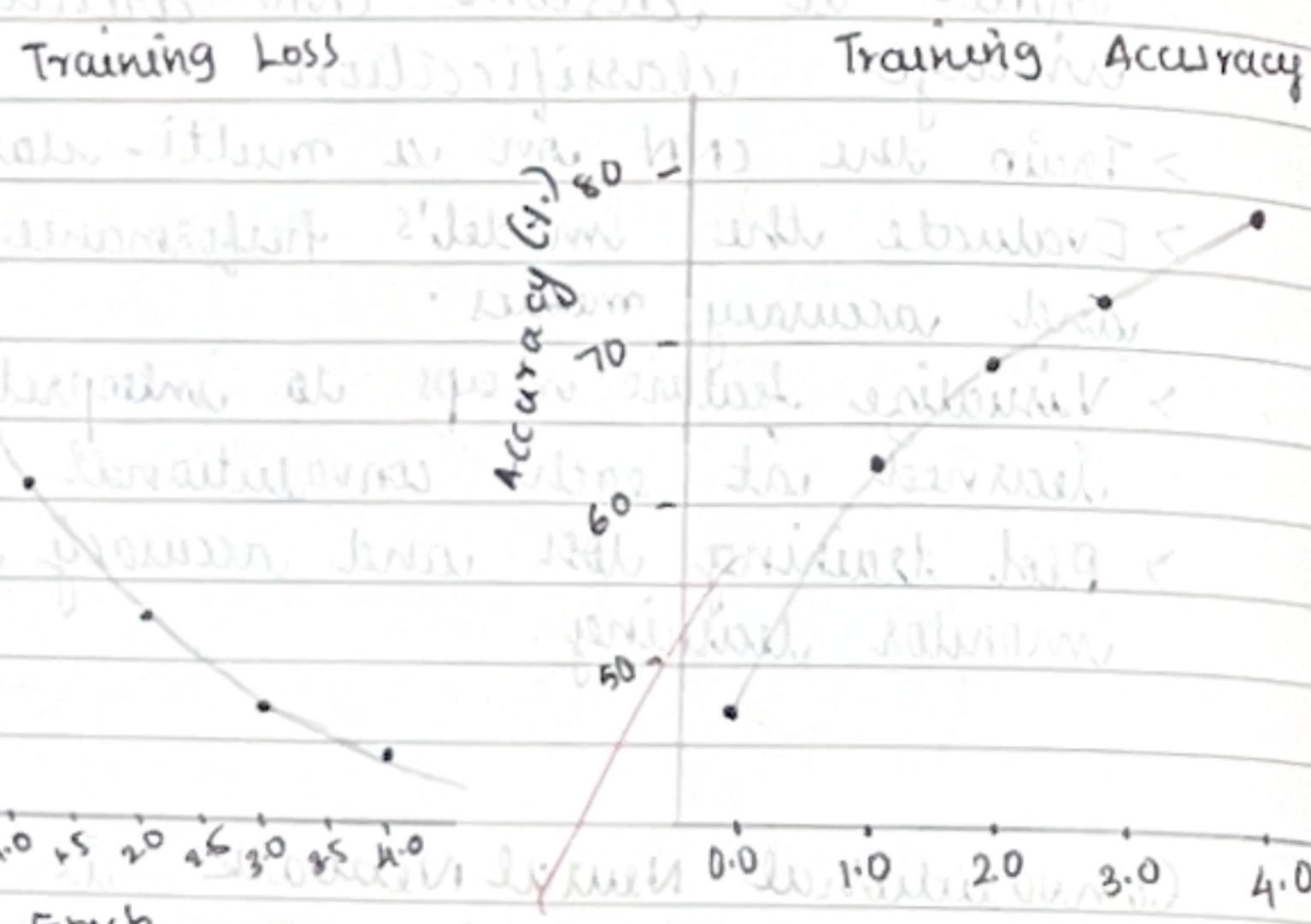


$$\text{size of feature map} = \frac{N-F+1}{S} + 1$$

where $N \Rightarrow$ original size of image
 $F \Rightarrow$ filter size
 $S \Rightarrow$ stride

OUTPUT:-

classes: ['airplane', 'automobile', 'bird', 'cat', 'deer',
'dog', 'frog', 'horse', 'ship', 'truck']
Epoch [1/5] Loss: 1.4277 Accuracy: 48.24%
Epoch [2/5] Loss: 1.0336 Accuracy: 63.44%
Epoch [3/5] Loss: 0.8669 Accuracy: 69.30%
Epoch [4/5] Loss: 0.7643 Accuracy: 73.09%
Epoch [5/5] Loss: 0.6843 Accuracy: 76.08%



Filter: A small matrix that slides over the input to detect specific patterns like edges or textures.

Feature Map: The output produced after applying filters that highlights detected features.
Stride: The step size with which the filter moves across the input image.

Receptive Field: The region of the input image that a neuron in a feature map is influenced by.

PSEUDOCODE:-

START

1. Import libraries and load data set (CIFAR-10)
2. Define CNN: Conv \rightarrow ReLU \rightarrow Pool \rightarrow Output layer
3. Loss Function & Optimizer
4. Train the model: Forward pass \rightarrow Loss \rightarrow Backpropagate \rightarrow update weights \rightarrow Track Accuracy.
5. Plot training loss and accuracy curves.

RESULT:-

> Loss decreasing over epoch; Accuracy increasing
> Feature Map:-

- Conv1: Detects edges and basic patterns
- Conv2: Detects textures and simple shapes
- Conv3: Detects complex patterns and object parts.

> Sample Prediction:- correctly classified images from test set.

CNN Architecture



RESULT:

- > Loss decrease over epoch
- > Accuracy increases
- > Correctly classified images from data set
- > Multiclass image classification system using CNN implemented successfully

10/12/2018