NAME: Aarush Talukdar Reg: RA2311047010014.

Serial. No.	Topic.	Date.	Signature.
1)	Exploring the Deep Learning Platforms & Frameworks	31/07/2025.	
2)	Implement a Classifier using an open-source dataset	7/8/2025	
3)	Study of Classifiers with respect to Statistical Parameter	7/8/2025	
4)	Build a simple feed forward network to recognize handwritten character	14/8/2025	
5)	Study of Activation Functions and its role	9/9/2025	9/9
6)	Implement gradient descent and backpropagation in deep neural network.	13/9/2025	
7)	Build a CNN model to classify Cat & dog image	13/9/2025	
8)	Experiment using LSTM	13/9/2025	
9)	Build a Recurrent Neural Network	13/9/2025.	
10)	Perform compression on MNIST		
11)	Experiment using VAE		
12)	Implement a DCGAN	02/11/25	
13)	Understand pre-trained model		
14)	Transfer Learning		
15	YOLO Model		

Completed

Expt-7 (Redo) 17/09

Build a CNN Model to classify Cat & Dog Image

Aim:

To build and train a CNN ~~using~~ for classifying cat & dog images.

Objective:

To build and train a Convolutional Neural Network (CNN) from scratch using PyTorch for binary image classification (Cats vs Dogs).

Pseudocode:-

1) Import Libraries (torch, torchvision)

2) Define image transformations

3) Load dataset from directory and split into training & validation

4) Create DataLoader for mini-batch processing.

5) Define CNN architecture:-

- Conv + ReLU + Maxpool
- Flatten
- Fully connected layer
- Sigmoid output

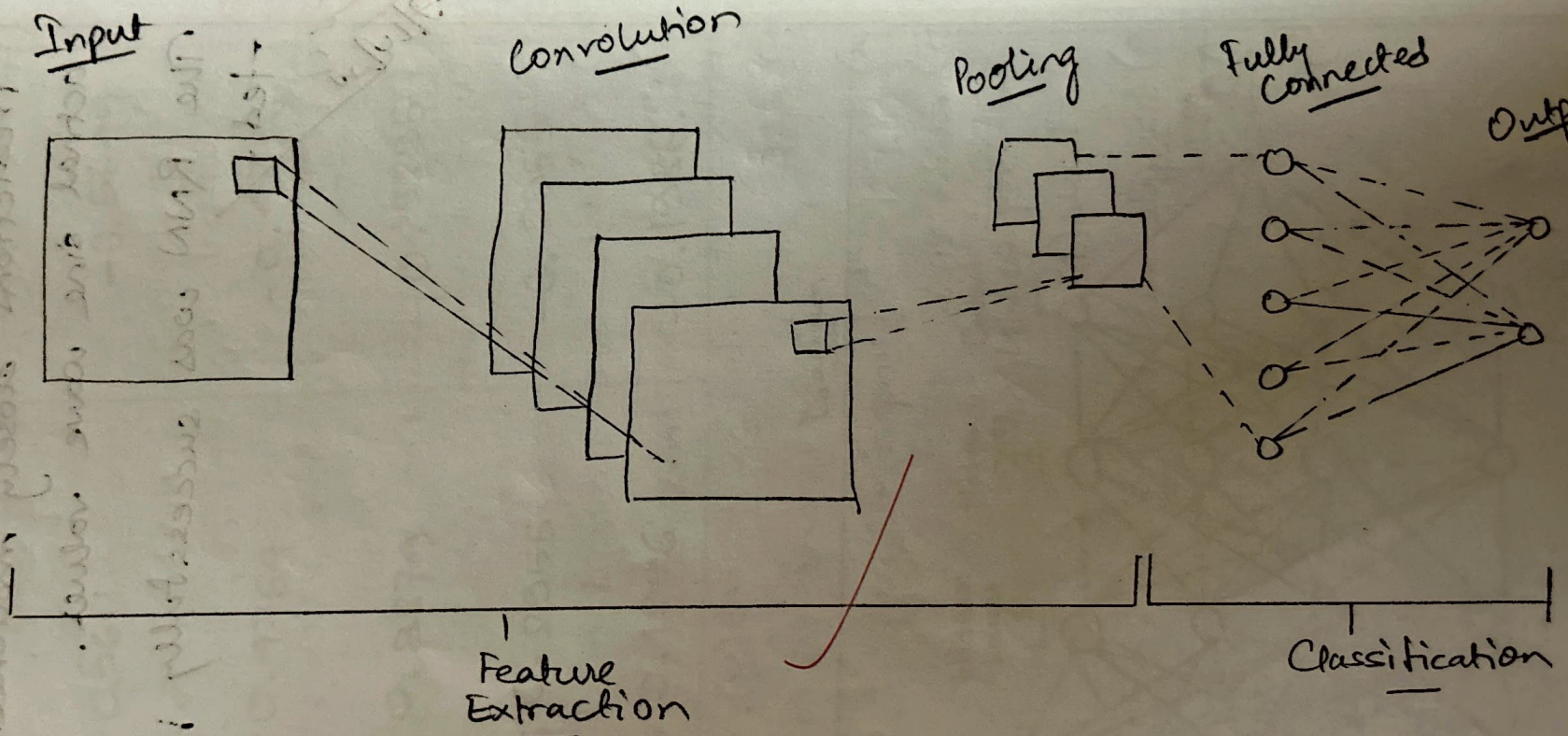
6) Set loss function = Cross-Entropy loss

7) Set optimizer = Adam

8) for each epoch:

a) Train model on training data.

CNN Architecture



b) Compute training loss & accuracy.

c) Evaluate on validation data.

d) Compute validation loss & accuracy.

9) Save trained model.

Observations:

- Model was successfully trained for 10 epochs.
- Training & validation gradually improved with each epoch.
- Data augmentation helped prevent overfitting
- The model achieved good accuracy.

Result:

The following experiment was carried out successfully.

Epoch	Loss	Accuracy
1/10	0.7750	56.63%
2/10	0.6851	56.85%
3/10	0.6692	57.98%
4/10	0.6440	61.8%
5/10	0.6394	63.82%
6/10	0.6380	64.49%
7/10	0.6404	65.62%
8/10	0.6365	65.62%
9/10	0.6246	67.42%
10/10	0.6006	68.76%