

Lab: 7 Build A CNN Model To Classify CAT AND DOG Image.

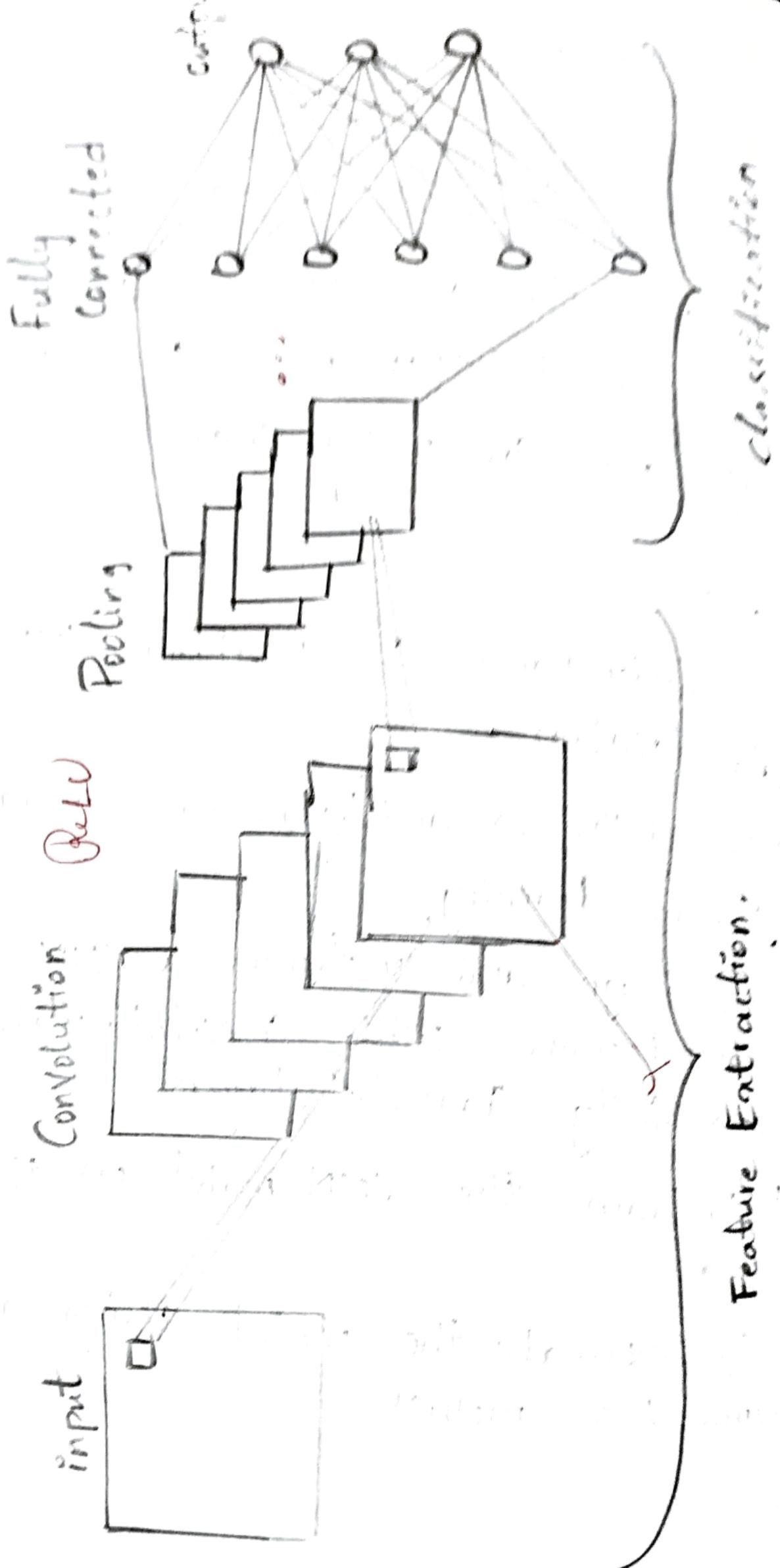
Aim:-

To build and train a convolutional neural network (CNN) model that classifies images into Cat or Dog categories

Objective:-

- 1, To Understand the Working CNN for image classification
- 2, To Preprocess and normalize image data for efficient learning
- 3, To design and implement a CNN architecture with Convolutional, Pooling and fully Connected layers.
- 4, To train the CNN model on the cat & Dog
- 5, To evaluate the model using accuracy and loss metric.

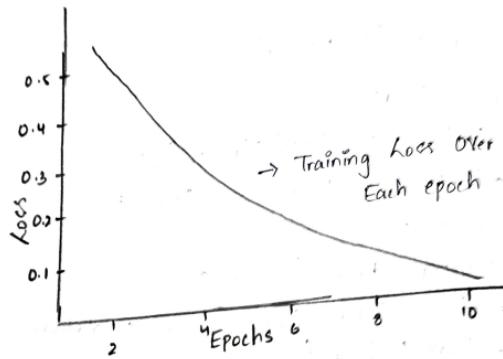
Convolution Neural Network



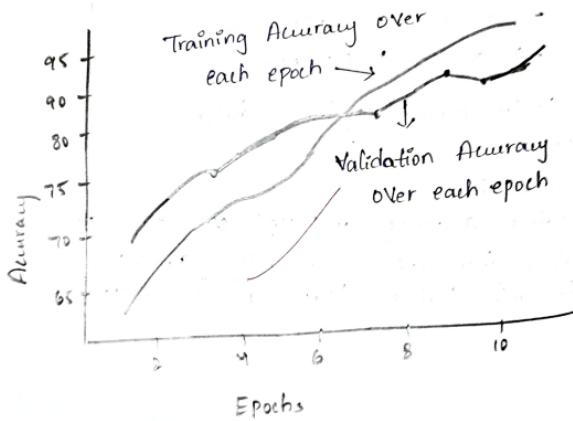
Section 5:

3. Import required libraries (TensorFlow (tf),
Numpy, matplotlib)
 4. Load dataset (Cat Vs dog)
 5. Preprocess data:
 - Normalize Pixel Values (0-1)
 - Split into train and test sets
 6. Define Model:
 - import layer (image size)
 - Convolutional layer + ReLU activation
 - Max Pooling Layer
 - Conv+pooling layer to extract
 7. Compile model With:
 - loss function = Binary Cross entropy
 - optimizer = adam
 - Metrics : Accuracy
 8. Train model on traing data.
- J, END.
- Observation:-
- Training accuracy improved over each epoch
 - Loss decreased significantly confirming efficient feature learning from CNN

Loss Graph



Accuracy Graph



Result:

The CNN model To classify CAT and Dog Image was Successfully executed.

Epochs over Training & Validation

Epoch [1/10] → Average Training Loss: 0.5688

Training Accuracy: 63.61%

Validation Accuracy: 71.95%

Epoch [2/10] → Average Training Loss: 0.5183

Training Accuracy: 78.76

Validation Accuracy: 77.08%

Epoch [4/10] → Average Training Loss: 0.5936

Training Accuracy: 81.81%

Validation Accuracy: 82.36%

Epoch [6/10] → Average Training Loss: 0.3921

Training Accuracy: 87.59%

Validation Accuracy: 84.42%

Epoch [10/10] → Average Training Loss: 0.0367

Training Accuracy: 98.56%

Validation Accuracy: 93.21%

~~11/23/23~~