

Build a CNN Model to classify cat or dog image.

AIM: To implement CNN to build a CNN model to classify a cat or dog image.

Pseudo Code:-

- Initialize parameters.

- Define input of shape = $[64, 64, 3]$ → image resolution chosen for uniformity.

- Define no. of classes = 2

- Set batch size, L.R.

- Load dataset

- Import cat & dog image from dataset.

- Preprocess - resize all images to same dimension.

- Normalize pixel values $[0, 1]$ to stabilize training

- Split into training & validation set.

- Data Augmentation

- Apply random flips, rotation, zoom.

- Construct CNN architecture

- Convolution layer

- Activation ReLU

- Pooling layer

- Repeat convolution + Pooling.

- Compare Model

- Optimize

- Loss function

- Evaluation metric = Accuracy.

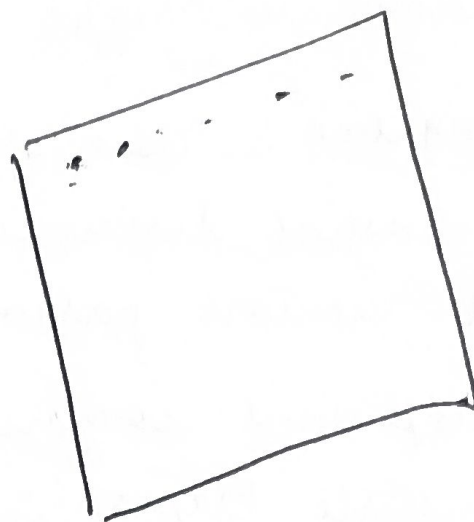
- Train Model
- Evaluate Model
- Calculate accuracy & loss on validation set
- Prediction
 - Get a new image
 - Resize to (64, 64)
 - Normalize pixel
 - Passing
 - If output $< 0.5 \rightarrow$ cat else \rightarrow dog

Observation:

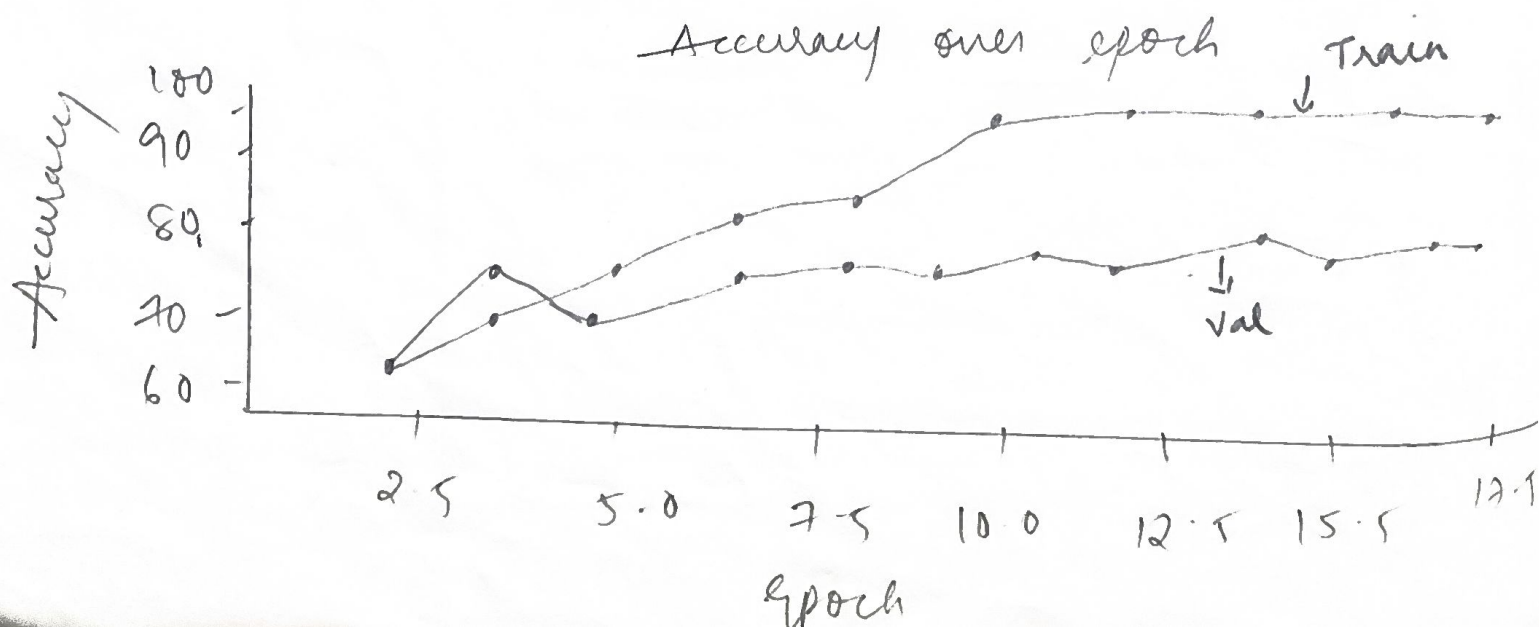
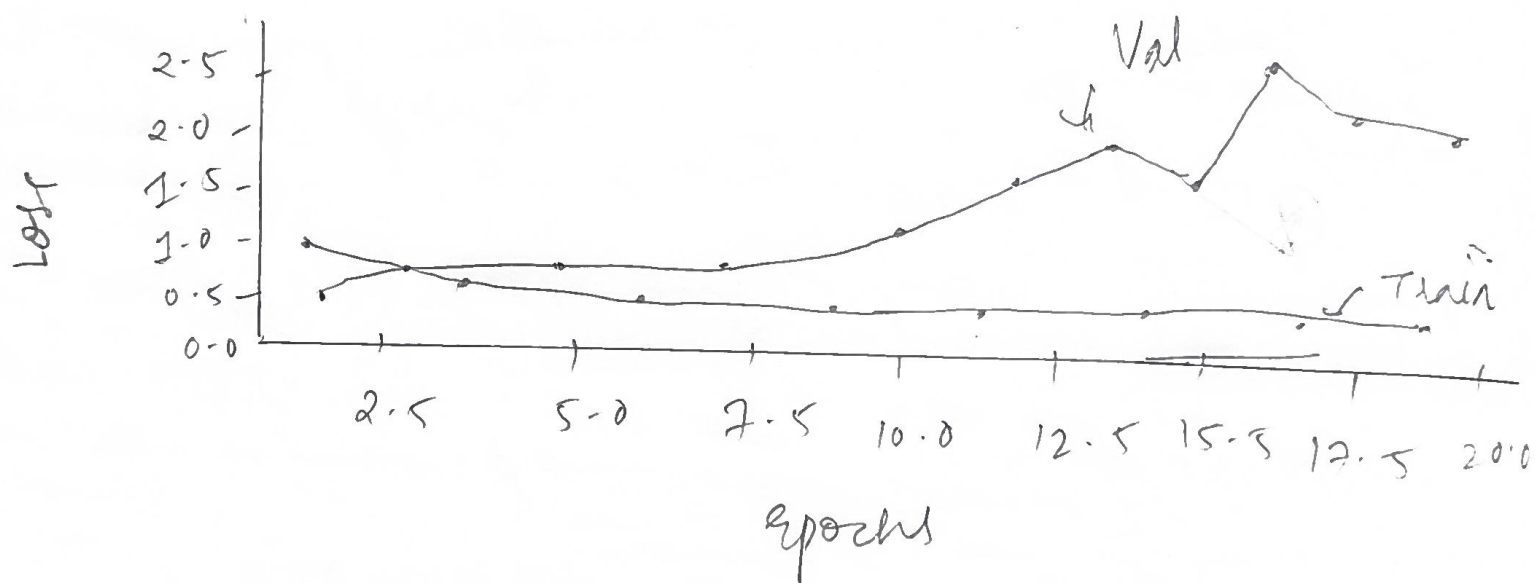
- \rightarrow Training accuracy improved reading 95%
- \rightarrow Validation accuracy peaked 80-85%
- \rightarrow Loss decreased significantly confirming effective feature learning from CNN.

Result:

The experiment was successfully executed.



Conv ?



O/P:

epoch 1 - Loss : 208.7533 - Acc : 0.64

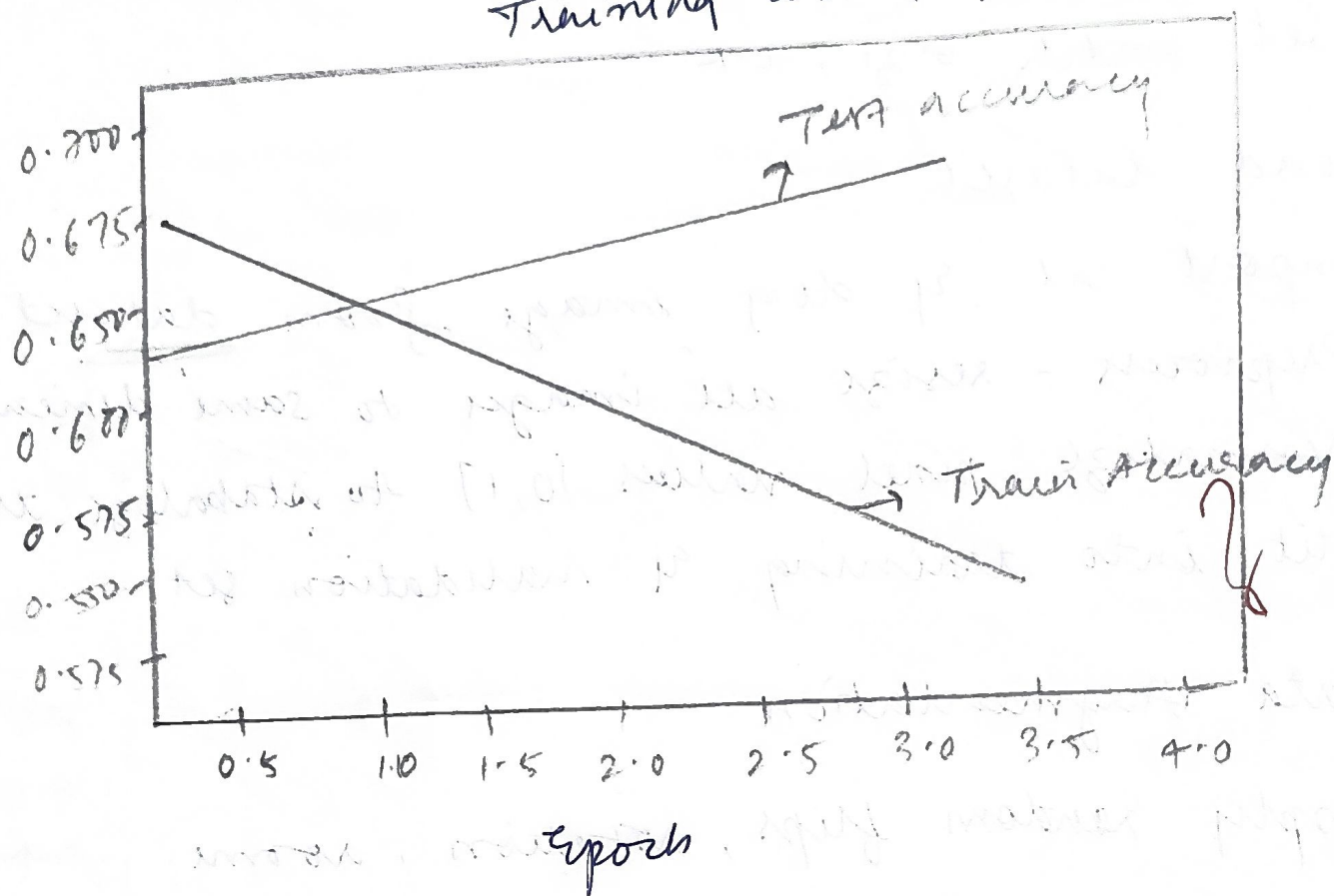
epoch 2 - Loss : 195.6160 - Acc : 0.67

epoch 3 - Loss : 181.9068 - Acc : 0.69

epoch 4 - Loss : 171.0776 - Acc : 0.69

epoch 5 - Loss : 180.1082 - Acc : 0.70

Training Loss & Test Accuracy

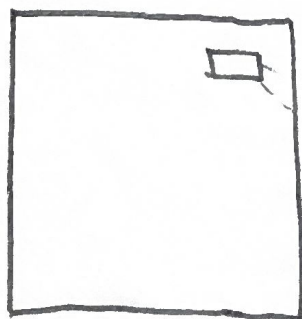


Loss over epoch

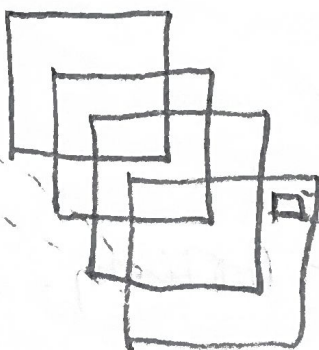
Fully

Connected

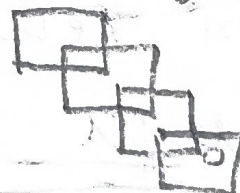
Input



Convolution



Pooling



Output

Feature extraction

Classification