

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] \rightarrow 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.

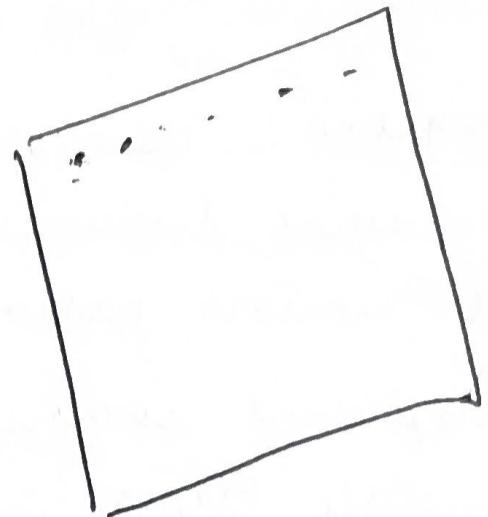
- Train Model
- Evaluate Model
- Calculate accuracy of loss on validation set
- Prediction
- Get a new image
- Range do (64, 64)
- Normalize pixel
- Parsing
- If output < 0.5 → cat else → dog

Observation:

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

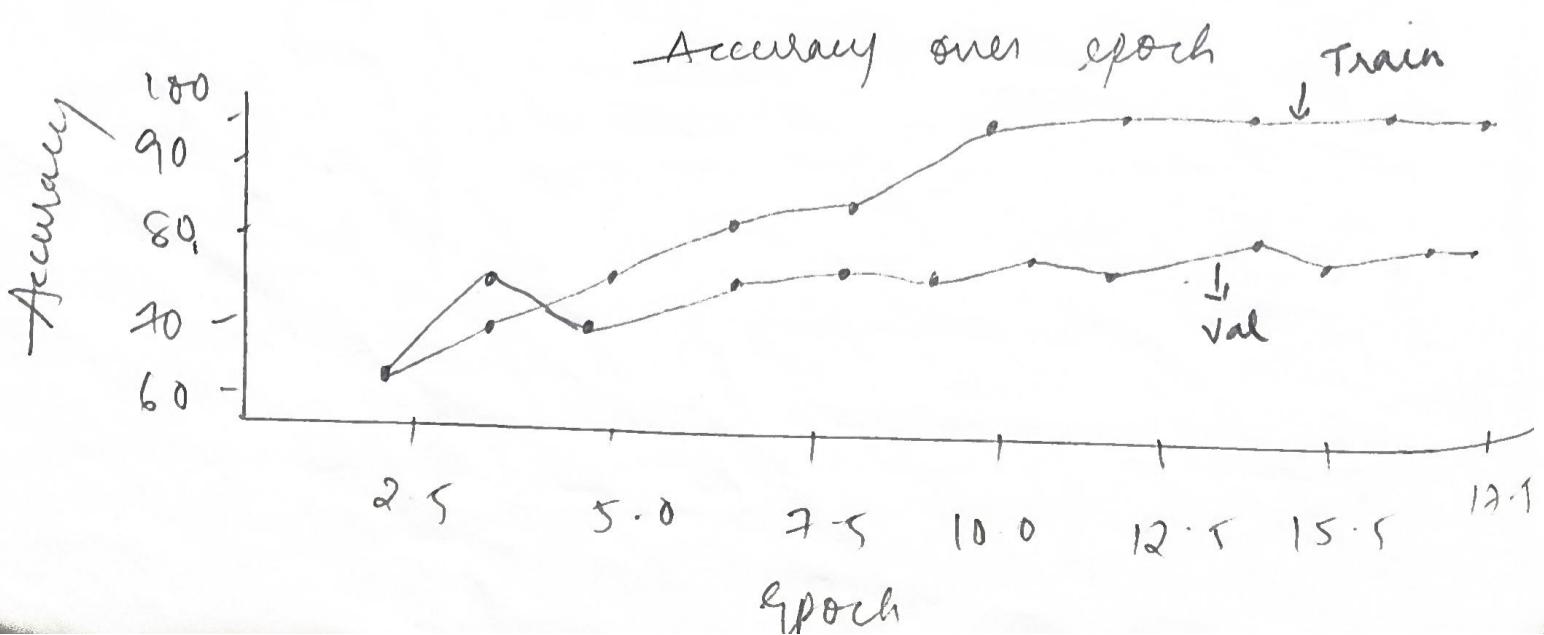
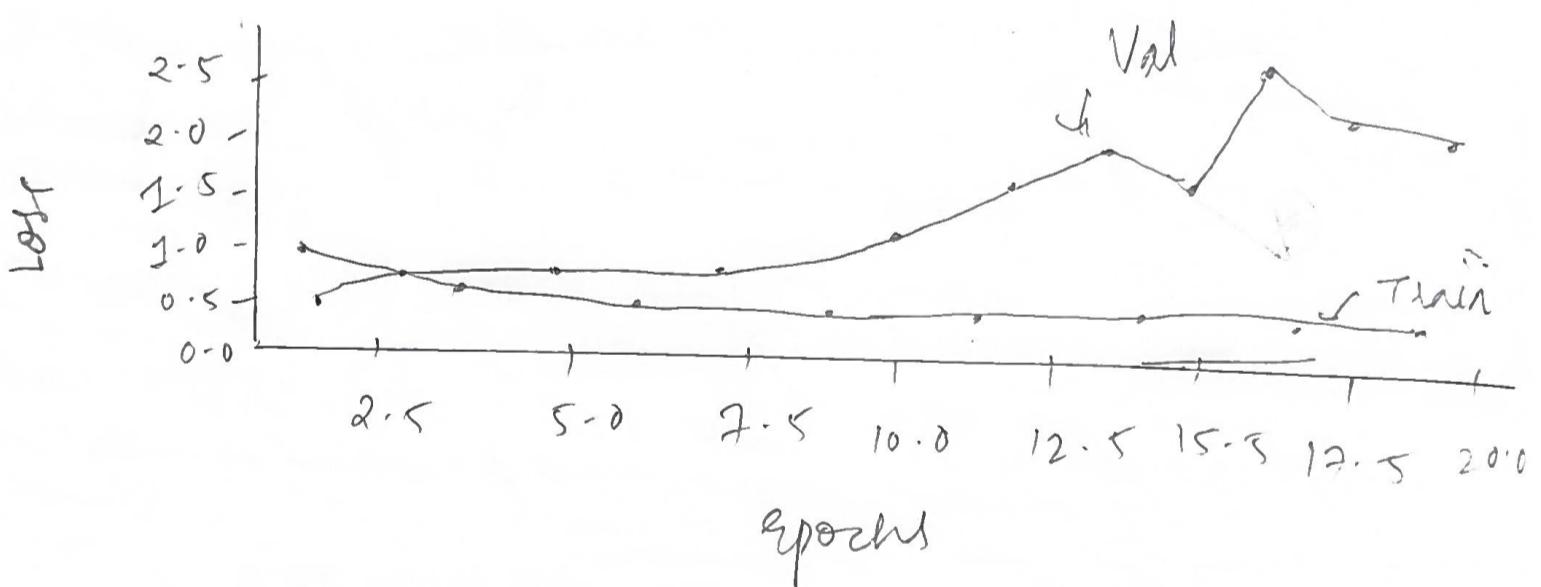
Result:

~~The experiment was successfully executed.~~



Cross

?



O/P:-

epoch 1 - loss : 208.7533 - Acc : 0.69

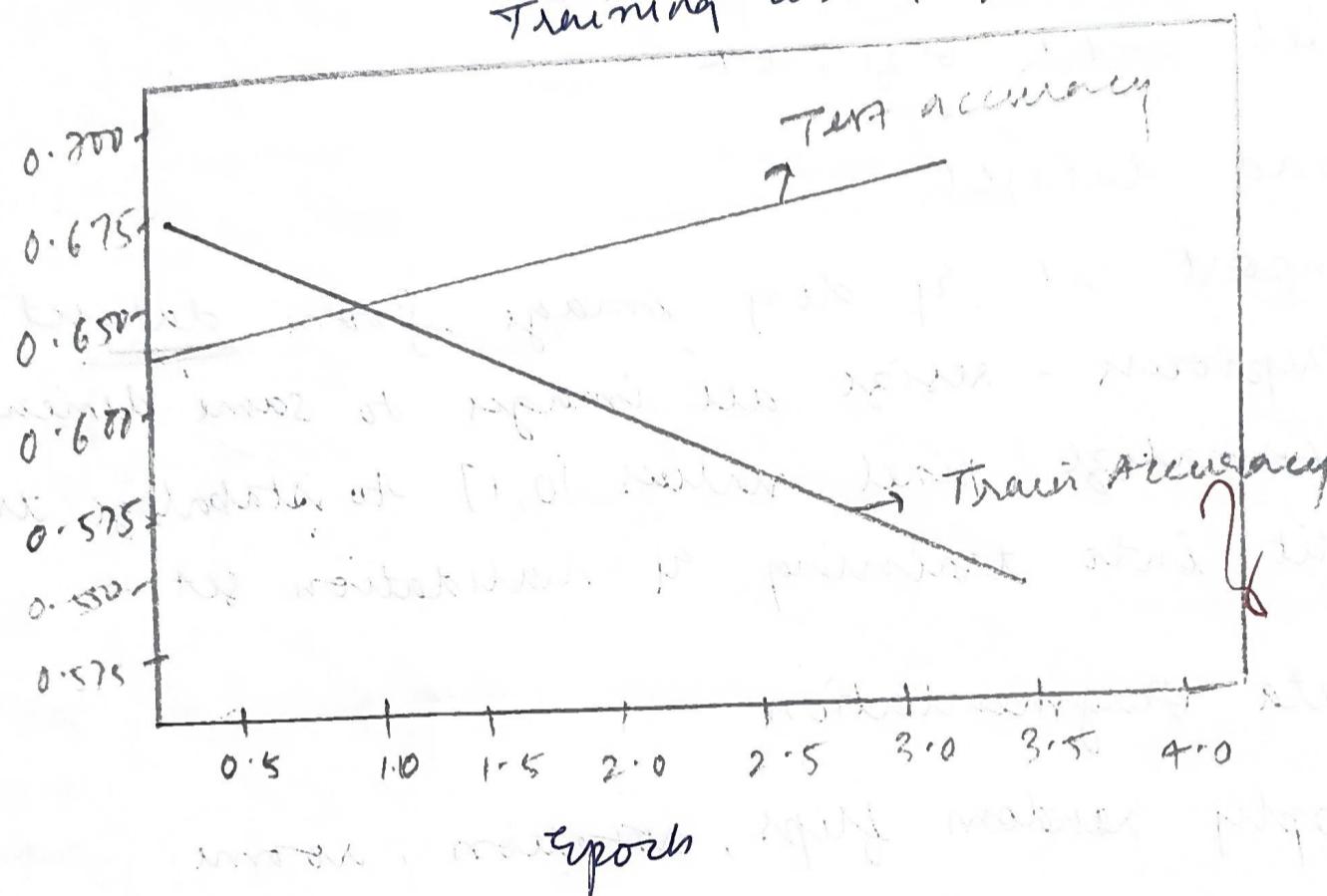
epoch 2 - loss : 195.6160 - Acc : 0.67

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

