**IMAGE CLASSIFIER USING CAT AND DOG DATASET**

**Data preprocessing**

Created two directories Training and Testing with subfolders Cat and Dog

Splitted the dataset with 0.9 of dataset to get the training and testing sets as follows

Training🡪Cat🡪11250 Training🡪Dog🡪11250

Testing🡪Cat🡪1250 Testing🡪Dog🡪1250

**Model**

1.Created model with keras API consisting of

CNN layers

First layer - Conv2D with 3x3 16 nodes, maxpooling2D 2x2, Dropout =0.25

Second layer - Conv2D with 3x3 32 nodes, maxpooling2D 2x2, Dropout =0.25

Third layer -Conv2D with 3x3 64 nodes, maxpooling2D 2x2, Dropout = 0.25

Activation =’relu’

Flatten layer

Dense layer with 512 nodes, relu activation, dropout layer=0.5

Dense layer with 1 node, sigmoid activation

Train and validation generator with image size (100,100), batch size=250

binary class

2. ResNet50 model with weight ‘imagenet’

Train and validation generator with data augmentation

image size (100,100),

batch size= 250,

binary class,

rotation\_range=40,

 width\_shift\_range=0.2,

       height\_shift\_range=0.2,

       shear\_range=0.2,

       zoom\_range=0.2,

        horizontal\_flip=True,

        fill\_mode='nearest'

**OPTIMIZER –** adam and RMSprop with learning rate 0.001

Training model with epochs=15 and epoch per step 90 since batch size 250

**PERFOMANCE OF THE MODELS**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **KERAS** | | **ResNet50** | |
| ADAM | RMSprop | ADAM | RMSprop |
| Training accuracy | 0.9402 | 0.8563 | 0.9692 | 0.9542 |
| Training loss | 0.1543 | 0.3436 | 0.0421 | 0.1163 |
| Validation accuracy | 0.8500 | 0.7287 | 0.9086 | 0.8580 |
| Validation loss | 0.3979 | 0.5924 | 0.2847 | 0.4359 |
| Size (in bytes) | 39675040 | 26466472 | 308855184 | 206238784 |
| Inference time | 0.083838. | 0.10808… | 0.05400. | 0.075400. |

Keras model created with API gives and adam optimizer achieves 94% accuracy while with RMSprop achieves 85% accuracy

ResNet 50 with weight imagenet and adam optimizer achieves 97% accuracy while with RMSprop achieves 95% accuracy