Computer Vision

Project Report

Distinguish Images of Dog from Cat

Rahul Kumar (160001047)
Gaurav Naukudkar (160001040)
Harsh Vardhan (16001025)

Index

1. Introduction

This project is aimed at building a classification system for images of dogs and cats using Convolutional Neural Network(CNN). This is an image classification problem where the dataset has images of dogs and cats(there are no such images where dog and cat both are present in the same picture) and CNN finally categorizes the data into either a dog or cat with a very high accuracy rate.

2. Background

We are using the Convolutional Neural Network(CNN) as our model for training the dataset. Below figures show the architecture which includes convolution and pooling layers.
The RGB image is fed into a Convolution layer whose output is filtered using a Max-Pool layer. Later follows 3 more pairs of convolution/max-pooling layers. Now the output is fed into a dropout layer, which assigns weights of some parameter to zero(this is done to reduce overfitting). The next layer flattens the previous output and outputs to a fully connected layer which further reduces the output size to 1. The final output is used to infer the class of the image.

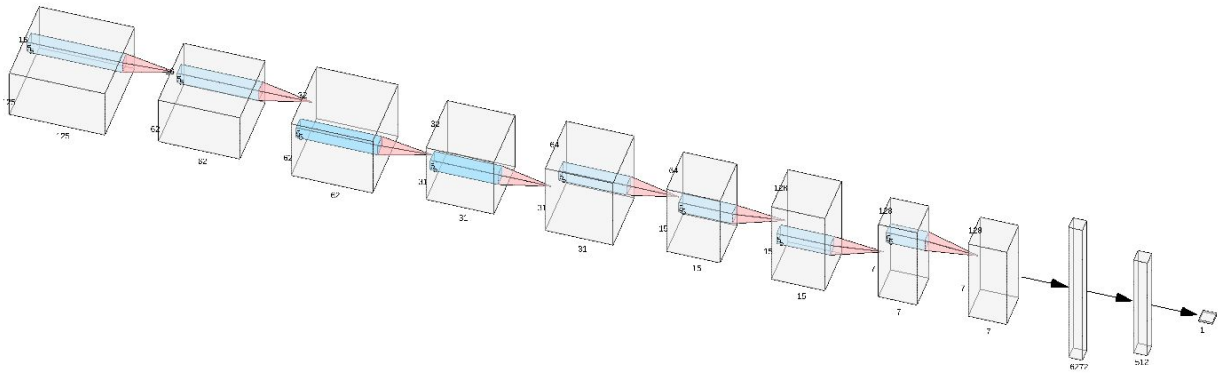

Fig:- Diagram of the CNN model

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 125, 125, 16)      1216
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 62, 62, 16)        0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 62, 62, 32)        12832
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 31, 31, 32)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 31, 31, 64)        51264
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 15, 15, 64)        0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 15, 15, 128)       204928
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 7, 7, 128)         0
_________________________________________________________________
dropout (Dropout)            (None, 7, 7, 128)         0
_________________________________________________________________
flatten (Flatten)            (None, 6272)              0
_________________________________________________________________
dense (Dense)                (None, 512)               3211776
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 513
=================================================================
Total params: 3,482,529
Trainable params: 3,482,529
Non-trainable params: 0
```

Fig:- Description of all layers of the model with their output tensor and number of parameter

3. Work Done

The code to run the model was created in the python programming language using the TensorFlow library.
The code contains two files, one for training and another for testing. The model has already been trained for 50 epochs(took around 12 hrs) and is saved. For testing purposes, the model is loaded from the saved file and runs on input images.
For training the model please download these files in a folder named “class_data” in the same directory as the code. Then run the python code for training.
The requirement for running the project is Python 3 and TensorFlow 2.2.
For testing, simply run the python code for the test.

4. Details of data

Training data was collected from Kaggle. It contains 25000 labeled images of dogs and cats and 12500 unlabelled test images.
For our purpose, we divided the data in ratio 6:2:2 in training, validation, and testing sets.

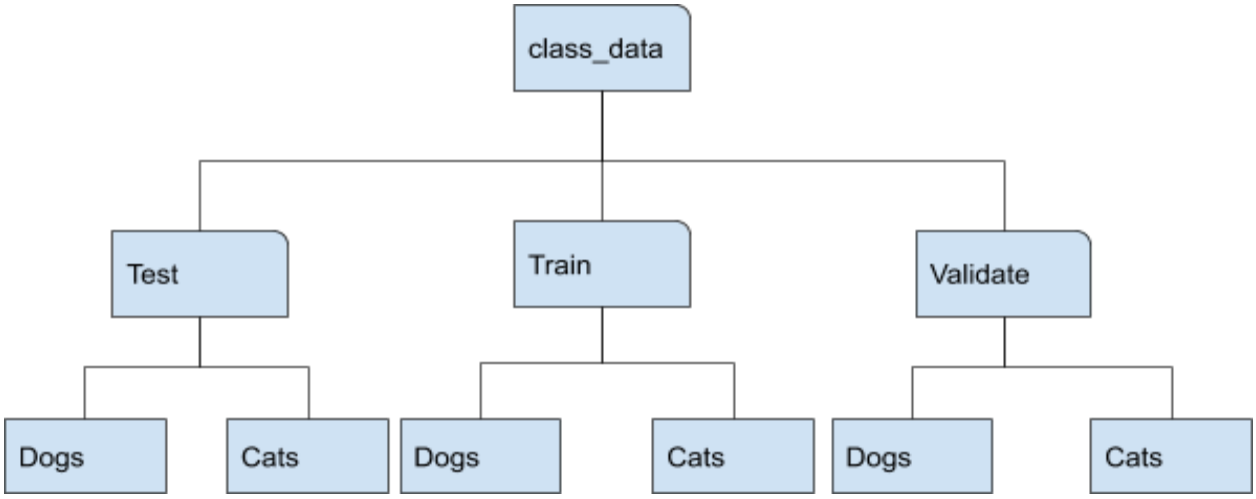

Fig:- final directory structure of the data

The above figure shows the directory structure of the data. This directory structure helps easy loading of data for training in TensorFlow.

For training purposes, we will be using RGB images of size 125*125 (images will be resized in the program).

5. Results and Analysis

We have trained our model with 15000 images for 50 epochs in batch size of 50.

The accuracy of the model is 90.22%.

The below figures show the accuracy and loss of each epoch for the training and validation set.

It can be inferred that the accuracy of the validation set has increased continuously and started to converge.

Some more results for individual images with labeled output is shown below.

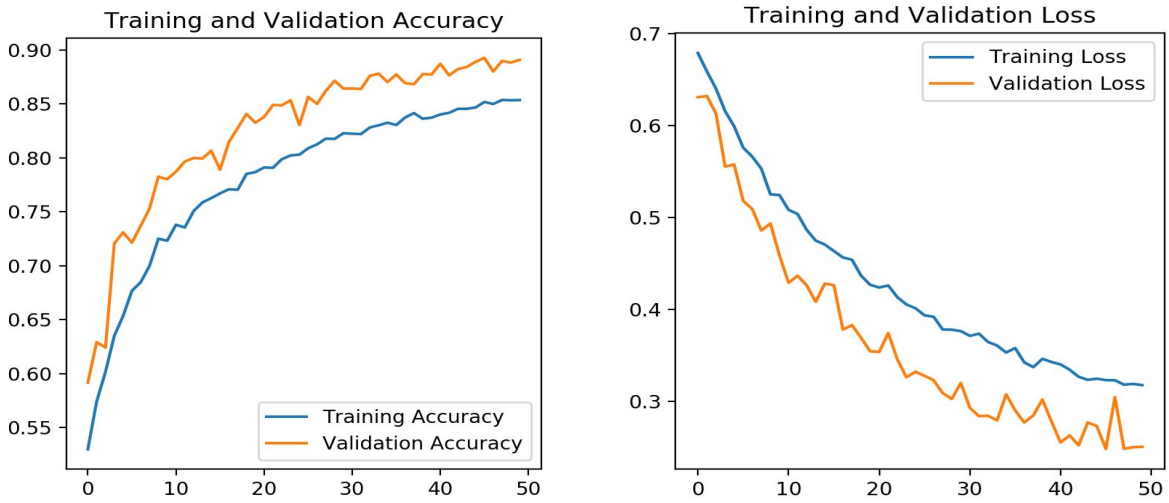

Fig:- Graph of accuracy and loss for training and validation set

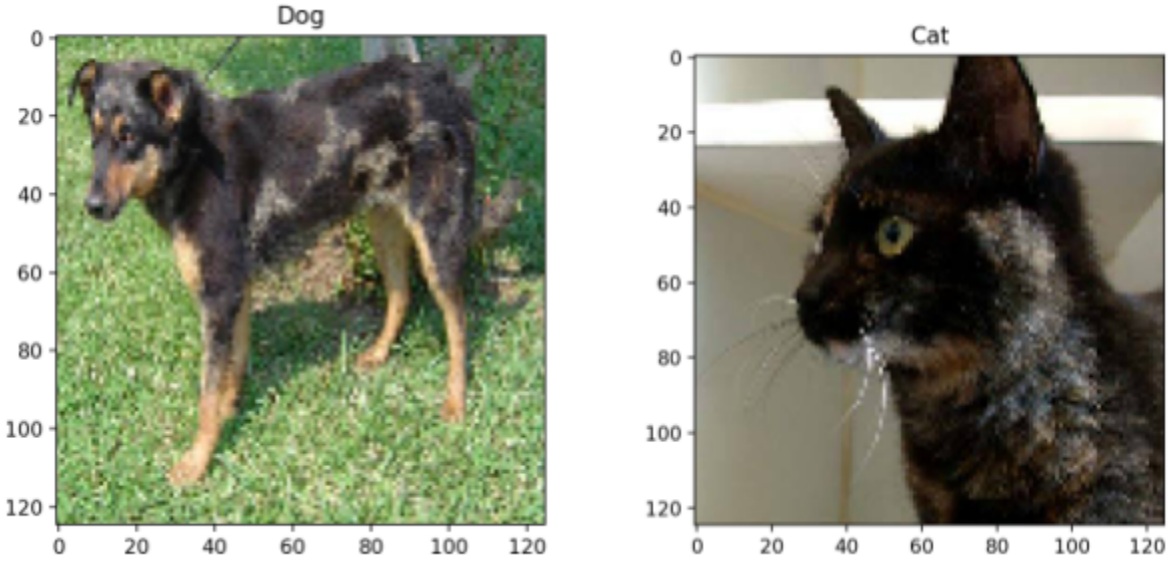

Fig:- Correctly labeled output images of cat and dog by the model

To prevent the model from overfitting, two approaches have been used. The first one is the introduction of the Dropout layer in the model. The dropout layer randomly drops some of the output.

Another approach is to introduce the training data augmentation where input images are flipped, rotated by some angle.

6. Conclusion

Our designed system is able to correctly distinguish between images of dogs and cats with an accuracy of 90.22%.