

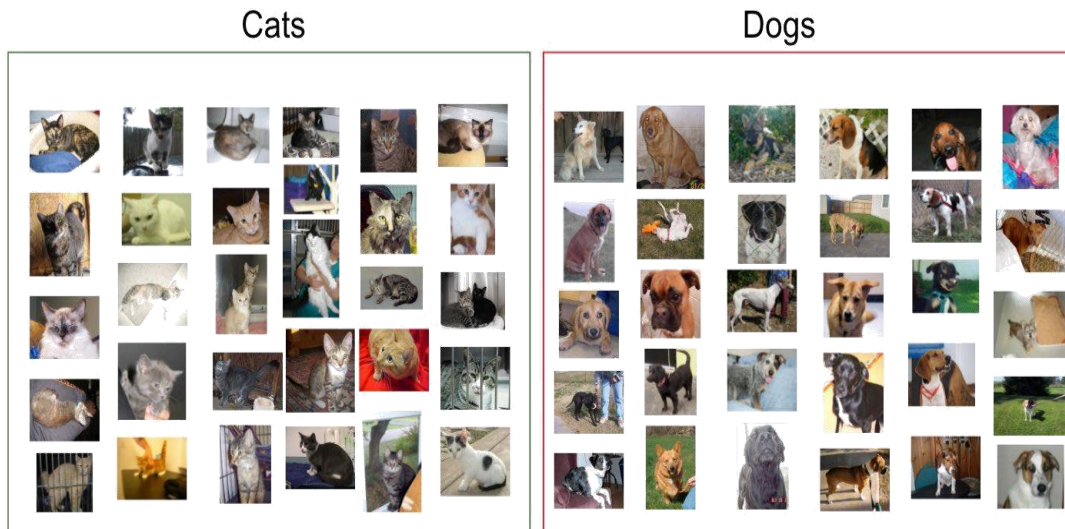


CNN.

Submitted by – Kartik Nagras

Dataset Explanation.

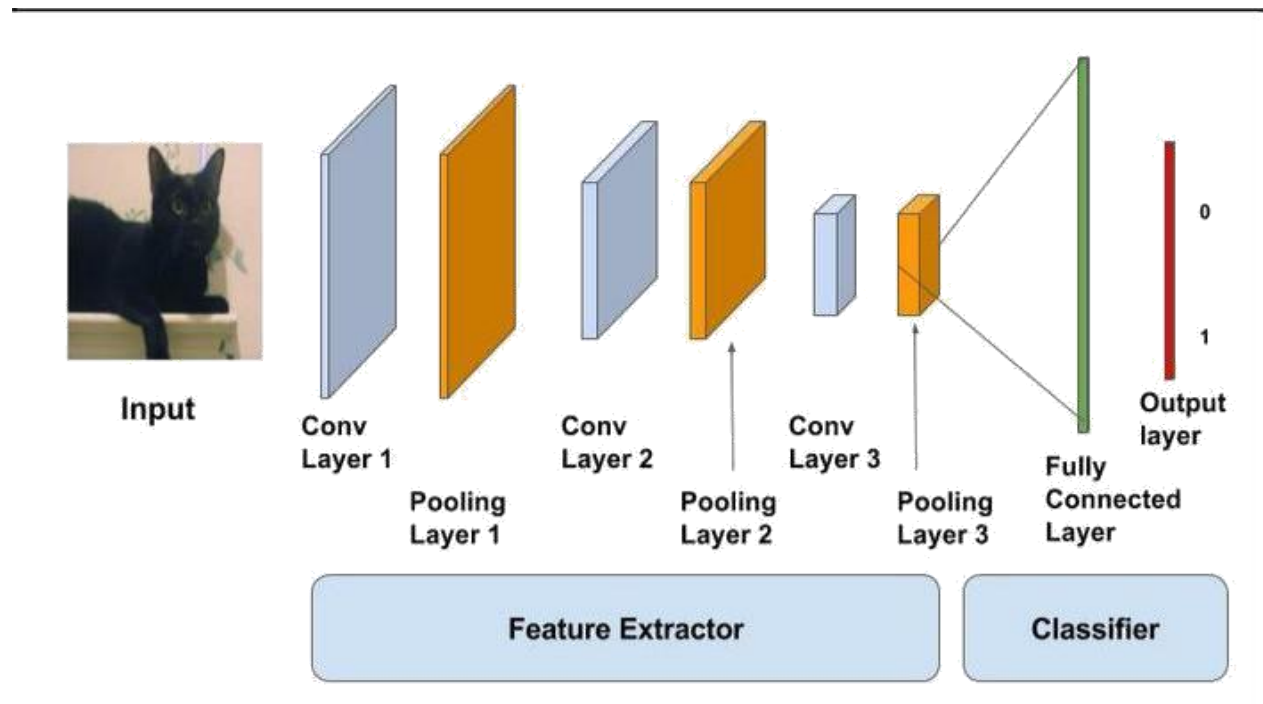
The Dataset is of 10,000 different images of **Dogs** and **Cats** and is taken from Kaggle <https://www.kaggle.com/c/dogs-vs-cats>.



Business Question:

We want our model to classify the input image into **Dog** or **Cat**.

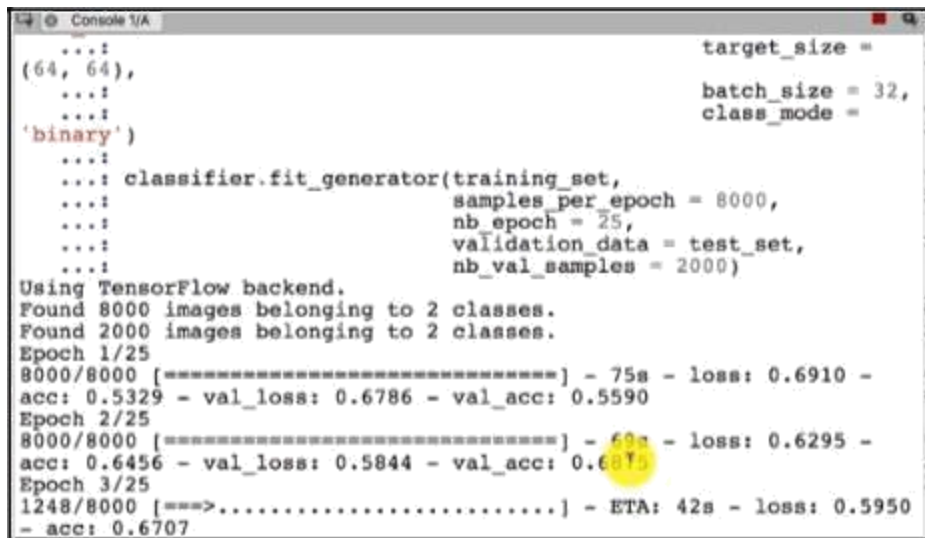
CNN Architecture.



Building CNN Steps:

1. Importing the Keras libraries and packages.
2. Initialising the CNN.
3. Adding Convolution Layer.
4. Pooling.
5. Adding a second convolutional layer (To improve accuracy of the model).
6. Flattening.
7. Full connection.
8. Compiling the CNN.
9. Fitting the CNN to the images.

Results –



```
...: target_size =
(64, 64),
...: batch_size = 32,
...: class_mode =
'binary')
...: classifier.fit_generator(training_set,
...:                         samples_per_epoch = 8000,
...:                         nb_epoch = 25,
...:                         validation_data = test_set,
...:                         nb_val_samples = 2000)
Using TensorFlow backend.
Found 8000 images belonging to 2 classes.
Found 2000 images belonging to 2 classes.
Epoch 1/25
8000/8000 [=====] - 75s - loss: 0.6910 -
acc: 0.5329 - val_loss: 0.6786 - val_acc: 0.5590
Epoch 2/25
8000/8000 [=====] - 60s - loss: 0.6295 -
acc: 0.6456 - val_loss: 0.5844 - val_acc: 0.6075
Epoch 3/25
1248/8000 [====>.....] - ETA: 42s - loss: 0.5950
- acc: 0.6707
```

```
Python console
Console 1/A
8000/8000 [=====] - 69s - loss: 0.3916 -
acc: 0.8245 - val_loss: 0.4658 - val_acc: 0.8005
Epoch 20/25
8000/8000 [=====] - 69s - loss: 0.3845 -
acc: 0.8280 - val_loss: 0.4415 - val_acc: 0.8045
Epoch 21/25
8000/8000 [=====] - 69s - loss: 0.3774 -
acc: 0.8281 - val_loss: 0.4406 - val_acc: 0.8130
Epoch 22/25
8000/8000 [=====] - 69s - loss: 0.3597 -
acc: 0.8382 - val_loss: 0.4444 - val_acc: 0.8105
Epoch 23/25
8000/8000 [=====] - 69s - loss: 0.3613 -
acc: 0.8404 - val_loss: 0.4327 - val_acc: 0.8120
Epoch 24/25
8000/8000 [=====] - 69s - loss: 0.3477 -
acc: 0.8444 - val_loss: 0.4628 - val_acc: 0.8025
Epoch 25/25
8000/8000 [=====] - 69s - loss: 0.3381 -
acc: 0.8516 - val_loss: 0.4486 - val_acc: 0.8180
Out[1]: <keras.callbacks.History at 0x122b2cbe0>
In [2]:
```

Total Time for execution = 20 mins.

Using training dataset to test our CNN model we get accuracy of **53.29%** after **1 epoch cycle**, and our accuracy increases to **85.16%** after **25 epoch cycle**.

On testing our CNN model on our testing dataset we get accuracy of **55.90%** after **1 epoch cycle**, and our accuracy increases to **81.80%** after **25 epoch cycle**.

Conclusion-

When we want to classify a given image into **Dog or Cat** we can do that using our **CNN model** at an accuracy of **81.80 %**.