Dog Breed Classification

Introduction

Kaggle provided a subset of ImageNet in order to practice fine-grained image categorization.with 120 breeds of dogs and a limited number training images per class. For image categorization, there is a need of little much knowledge of image processing and Convolution Neural Networks. All the operations are done after resizing the image to 64x64 pixels. After building an architecture of convolution neural network, classifier needed to be fit CNN to images by using shortcut process which is very practical for process called Image argumentation that basically consists of pre-processing of images to prevent overfitting. If don't use image argumentation process classifier get more accuracy of train set but for test set it will be worst that is called overfitting.

Given

Train set- Contains images of dogs. Each image has a file name that is unique id number.

Test set- Contains images of dog. Each image has a file name that is unique id number.

Label.csv - Contains id number of train set images of with their breed of dog.

Submission.csv- Contains only id number of test set images.

Breeds - According to problem, 120 different breeds of dog name had given to analyse the data.

Output

Classifier predicts the breed's name of test images.

Libraries Used

1. NumPy-

Creating indexing comparison and filtering reshaping and combining of arrays. Reading Text Files

2. Panda-

Aggregation, transformations, slicing, indexing ,subsetting merging and joining of datasets

3. OpenCv-

Used for Image Processing. Manipulating with images like reading, resizing of images.

4. TensorFlow-

TensorFlow is an open source library for fast numerical computing.It was created and is maintained by Google and released under the Apache 2.0 open source license. The API is nominally for the Python programming language, although there is access to the underlying C++ API.

● Algorithm Used

Convolution Neural Network(CNN)

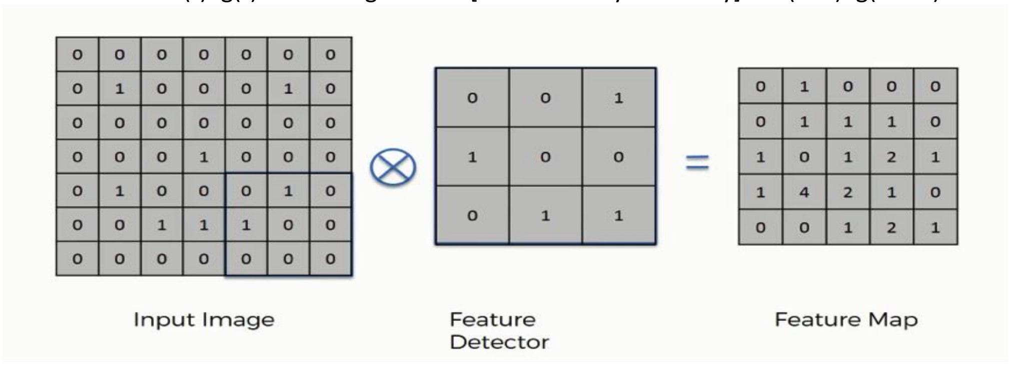
Convoution neural networkl have wide applications in image and video recognition, recommender systems and natural language processing. Convolutional neural networks (CNN, ConvNet) is a class of deep, feed-forward (not recurrent) artificial neural networks that are applied to analyzing visual imagery.

Steps:-

1. Convolution

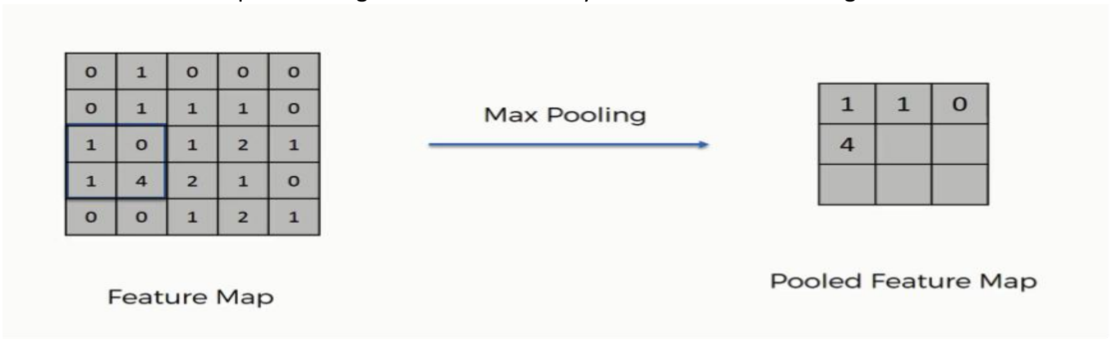
It reduces the image size by using feature detector by finding a feature of image.

Formula- F(t)*g(t) = integration of[limits -infinity to infinity] of F(Tau)*g(t-Tau)



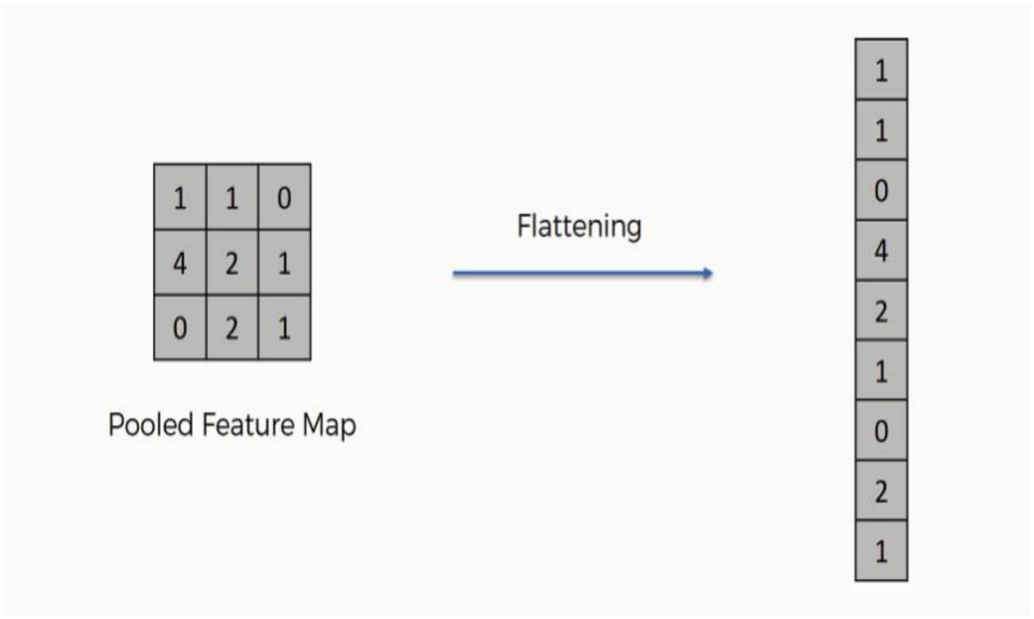
2. Max Pooling

It uses the feature map for finding the closest similarity and removes overfitting



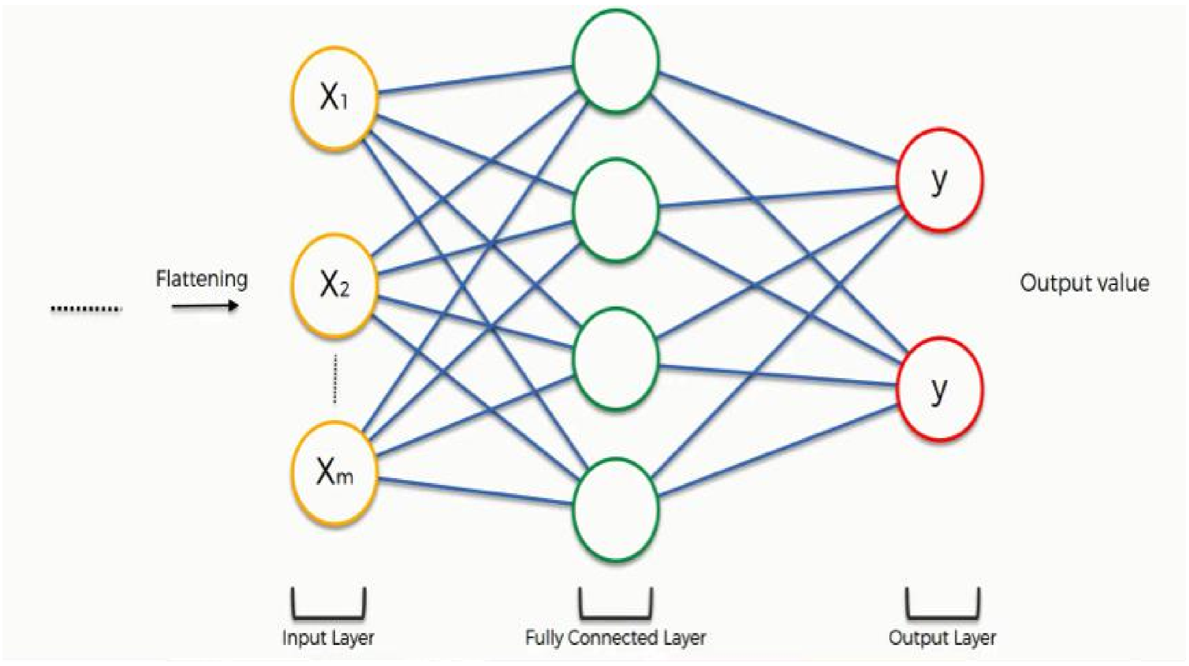
3. Flattening

After finishing the previous two steps, we're supposed to have a pooled feature map by now. As the name of this step implies, we are literally going to flatten our pooled feature map into a column like in the image below



4. Full Connection

As i said in the previous step, the input layer contains the vector of data that was created in the flattening step. The features that we distilled throughout the previous steps are encoded in this vector. At this point, they are already sufficient for a fair degree of accuracy in recognizing classes.



- **Keywords**

1. Softmax

For example, there is classification problem of two breeds(Bull Dog, Rottweilerr). Our aim is to find a class of a image.

Class	**Probability**
Bull Dog	65%
Rottweiler	80%

Its not good process to analyse because it gives a worst solution. This situation have a solution which named Softmax.

Formula -

Fj(z) =(e(exp(z))*j) / (Sum[till k](e(exp(z))*k))

Class	**Fj**
Bull Dog	0.15
Rottweiler	0.85
Total	1

2. Cross entropy

In convolution neural network, MSE can be used but better option in convolution network after apply softmax function turns out to be in Cross entropy function These functions are called lost funtion in convolution.

Lost funtion is something that we want to minimize in order to maximize optimization of a network.

Formula-

H(p,q) = -sum[till](x(p(x)*logq(x))

According to above example,

Class	Fj	Prediction
Bull Dog	0.1	0
Rottweiler	0.9	1

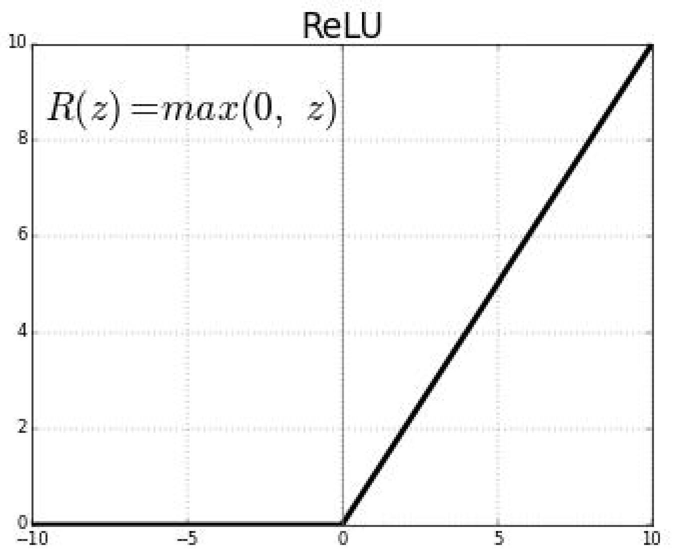
H(p,q) = -sum[till](x(p(x)*logq(x))

If we have to calculate cross entropy for bull dog then p=0 and q=0.1

If we have to calculate cross entropy for bull dog then p=1 and q=0.9

3.Relu(rectified linear unit)

It is the rectifier layer which is used after convolution to increase non-linearity in a image. It turns in non-negative values



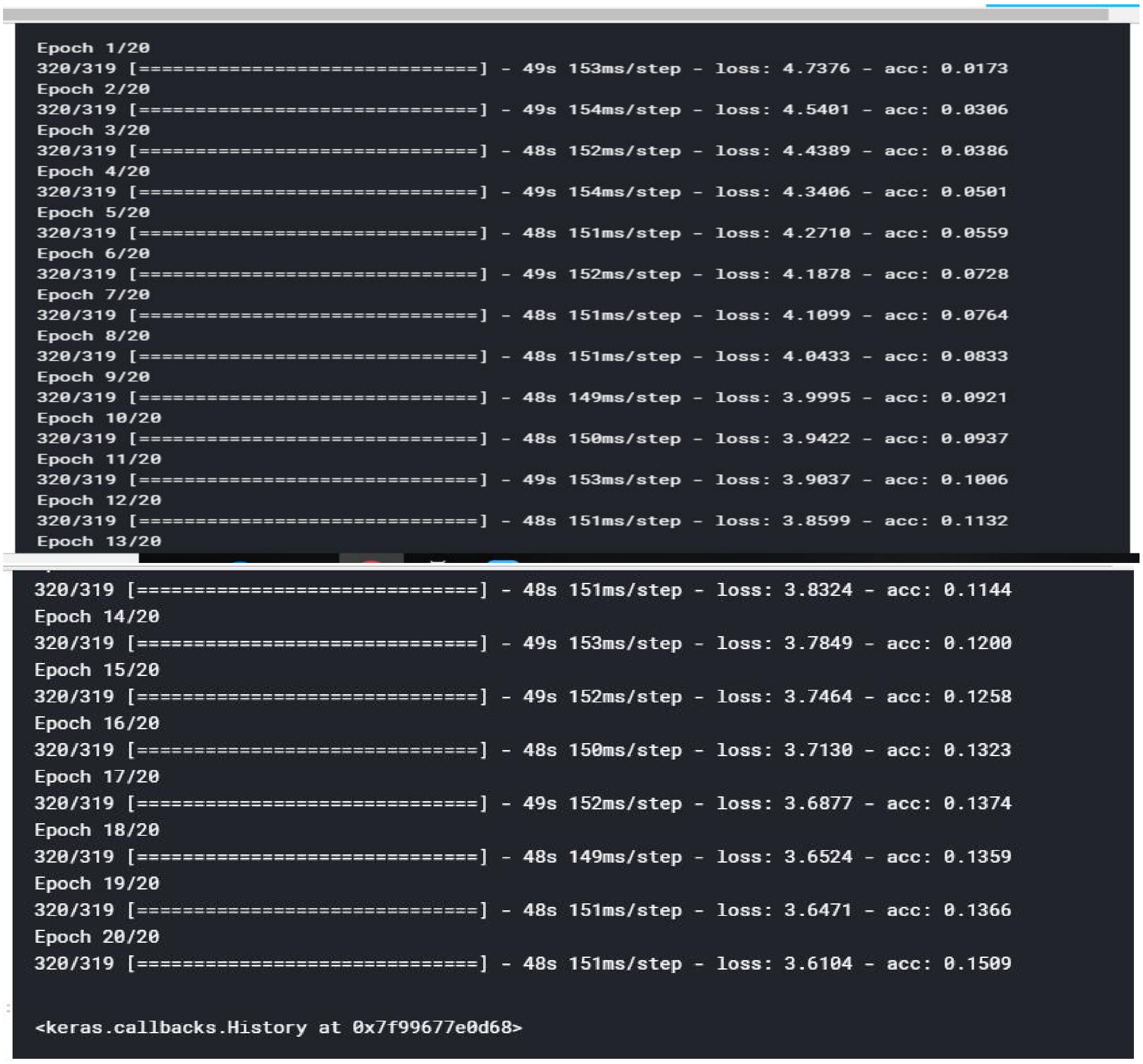
.

- **Process Cycle**

Libraries and Datasets	•Importing libraries •Importing datasets and label as train set and test set.
Encoding	•Encode the categorical data, according to this classifier breed is a categorical data.
Reading and Resizing	•Resizing of train set images to 64 x 64 pixels. •Resizing of test set images to 64 x 64 pixels
Append	•Append new images which cames from previous step in new datasets of train and test sets.
Standardize	•Standardizing the train set and test set by dividing the array by 255.
Initialize	•Initializing the CNN.
Convoluton 2D and Max Pooling	•Adding convolution 2D and Max pooling. •Applying 3 layers of convolution.
Flattening	•Adding flatting method to classifier.
Full Connection	•Adding Dense method. •Using Relu function •Using Softmax Function
Compiling	•Compiling the classifier using optimizer adam and loss function as cross_entropy.
ImageDataGenerator	•Importing ImageDataGenerator library.
Fitting and Generating	•Fitting a train set and generating the results by using classifier.

- **Results**

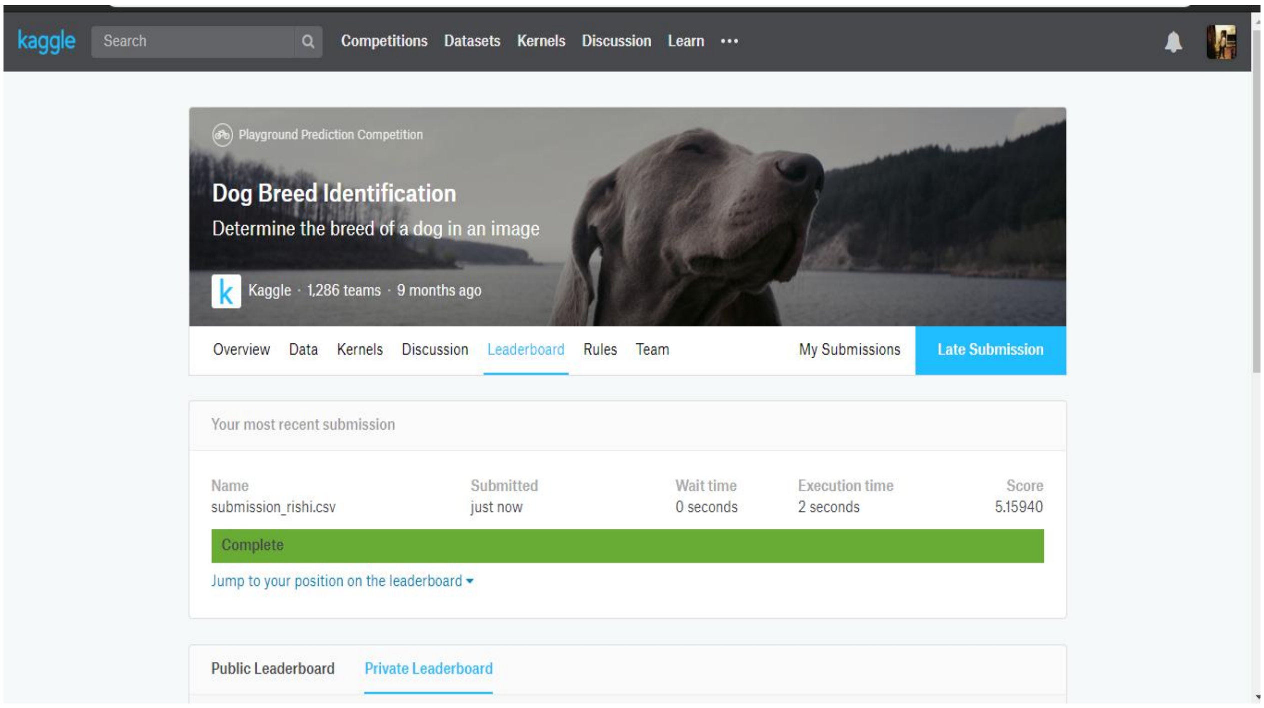
Accuracy and loss of train set



```
Epoch 1/20
320/319 [==============================] - 49s 153ms/step - loss: 4.7376 - acc: 0.0173
Epoch 2/20
320/319 [==============================] - 49s 154ms/step - loss: 4.5401 - acc: 0.0306
Epoch 3/20
320/319 [==============================] - 48s 152ms/step - loss: 4.4389 - acc: 0.0386
Epoch 4/20
320/319 [==============================] - 49s 154ms/step - loss: 4.3406 - acc: 0.0501
Epoch 5/20
320/319 [==============================] - 48s 151ms/step - loss: 4.2710 - acc: 0.0559
Epoch 6/20
320/319 [==============================] - 49s 152ms/step - loss: 4.1878 - acc: 0.0728
Epoch 7/20
320/319 [==============================] - 48s 151ms/step - loss: 4.1099 - acc: 0.0764
Epoch 8/20
320/319 [==============================] - 48s 151ms/step - loss: 4.0433 - acc: 0.0833
Epoch 9/20
320/319 [==============================] - 48s 149ms/step - loss: 3.9995 - acc: 0.0921
Epoch 10/20
320/319 [==============================] - 48s 150ms/step - loss: 3.9422 - acc: 0.0937
Epoch 11/20
320/319 [==============================] - 49s 153ms/step - loss: 3.9037 - acc: 0.1006
Epoch 12/20
320/319 [==============================] - 48s 151ms/step - loss: 3.8599 - acc: 0.1132
Epoch 13/20
320/319 [==============================] - 48s 151ms/step - loss: 3.8324 - acc: 0.1144
Epoch 14/20
320/319 [==============================] - 49s 153ms/step - loss: 3.7849 - acc: 0.1200
Epoch 15/20
320/319 [==============================] - 49s 152ms/step - loss: 3.7464 - acc: 0.1258
Epoch 16/20
320/319 [==============================] - 48s 150ms/step - loss: 3.7130 - acc: 0.1323
Epoch 17/20
320/319 [==============================] - 49s 152ms/step - loss: 3.6877 - acc: 0.1374
Epoch 18/20
320/319 [==============================] - 48s 149ms/step - loss: 3.6524 - acc: 0.1359
Epoch 19/20
320/319 [==============================] - 48s 151ms/step - loss: 3.6471 - acc: 0.1366
Epoch 20/20
320/319 [==============================] - 48s 151ms/step - loss: 3.6104 - acc: 0.1509

<keras.callbacks.History at 0x7f99677e0d68>
```

Score of Assignment



Dog Breed Identification

Determine the breed of a dog in an image

Kaggle · 1,286 teams · 9 months ago

Your most recent submission

```
Name                 Submitted   Wait time   Execution time   Score
submission_rishi.csv just now    0 seconds   2 seconds        5.15940
Complete
Jump to your position on the leaderboard
```

- **Conclusion**

One of the important real-world classification problems is the **Dog Breed Identification**. In this assignment, systematic efforts are made in designing a system which results in the prediction of breeds of dog. During this work, one deep learning classification algorithm is studied and evaluated on various measures. Experiments are performed on Kaggle's datasets .In future, the designed system with the used deep learning classification algorithms can be used to predict other animals identification.