Springboard – DSC

Capstone Project III

Establishing Images Classification System by Convolutional Neural Network (CNN)

Final Report

Author: Wong Lok Hang Ronald
Mentor: Wayne Ang

Content

1. Introduction

Image classification is one of the hottest topics in Machine Learning's world, it sits at the intersection of many academic subjects, such as Computer Science (Graphics, Algorithms, Theory, Systems, Architecture), Mathematics (Information Retrieval, Machine Learning), Engineering (Robotics, Speech, NLP, Image Processing), Physics (Optics), Biology (Neuroscience), and Psychology (Cognitive Science).

In the scope of image classification problems, a set of images process that are all labeled with a single category is given, then ML methods are applied to predict these categories for a novel set of test images and measure the accuracy of the predictions.

As a powerful image classification system can benefit different research topics, this project aim at exploring and establishing a baseline model of image classification by Convolutional Neural Network (CNN)

1.1 Objective

The main objective of this project is building a CNN model which can classify 20 types of dog breeds from the labelled images. Although this model can only apply in one daily life problem which is helping people recognizing different breeds of dog. The ultimate goal of the project is building up a baseline model which can apply for different daily life problems such as classifying different kinds of vehicles, animals, face recognition and even in medical use i.e. helping doctor to classify X-ray. Hence, classifying dog breeds is only the first step into the world of image classification.

2. Dataset

The original data source is found on http://vision.stanford.edu/aditya86/ImageNetDogs/ and contains additional information on the train/test splits and baseline results.

The Stanford Dogs dataset contains images of 120 breeds of dogs from around the world. This dataset has been built using images and annotation from ImageNet for the task of fine-grained image categorization. Contents of this dataset:

- Number of categories: 120
- Number of images: 20,580
- Annotations: Class labels, Bounding boxes

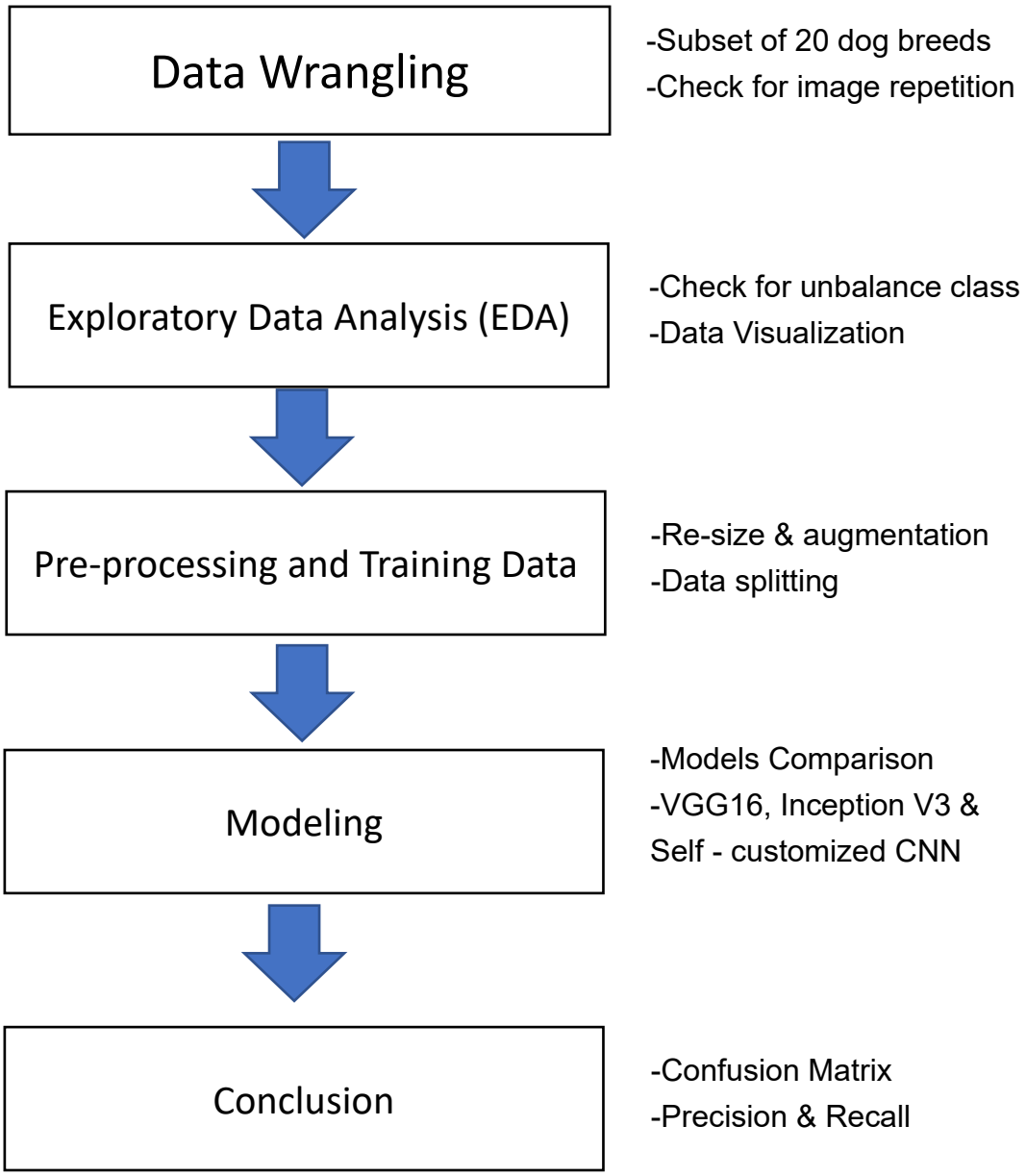
Due to the limit of computation power and constraint of time, this project adopts subset of the Stanford Dogs dataset as following:

- Number of categories: 20
- Number of images: 3,629

The above figure shows the preview of data and the data contains images and labels (dog breeds).

3. Project Outline

This project followed the following workflow to establish three image CNN models and select the best one:


Data Wrangling
-Subset of 20 dog breeds
-Check for image repetition
Exploratory Data Analysis (EDA)
-Check for unbalance class
-Data Visualization
Pre-processing and Training Data
-Re-size & augmentation
-Data splitting
Modeling
-Models Comparison
-VGG16, Inception V3 &
Self - customized CNN
Conclusion
-Confusion Matrix
-Precision & Recall

4. Data Wrangling

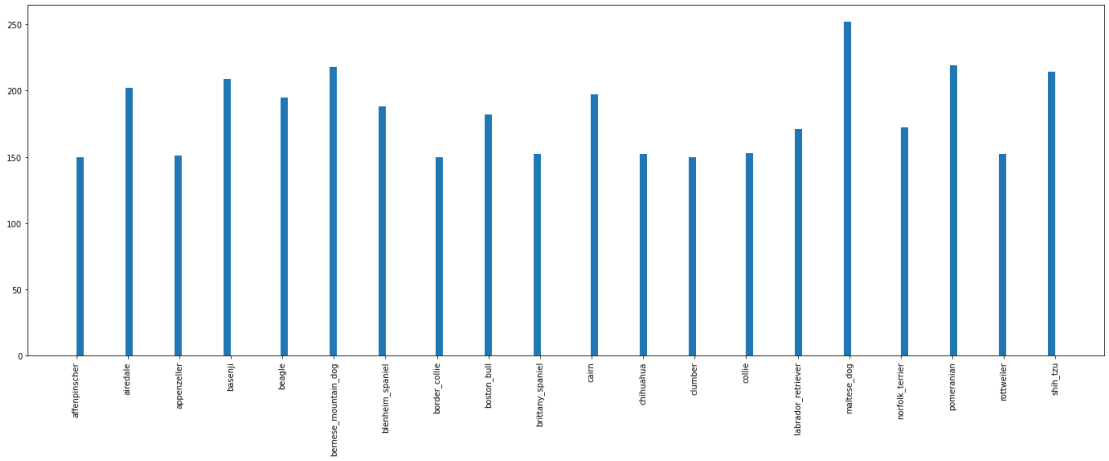
As the original dataset has a good quality. Data wrangling of this project consists of only 2 steps as below:

1) Create subset of 20 dog breeds data from the original Stanford Dog Dataset.

2) Check for any repetition of images in each categorical.

5. Exploratory Data Analysis (EDA)

The objective of EDA is to explore any unbalance classes among the 20 dog breeds data. The plot showing number of images across the 20 dog breeds.



From above plot, the data set has no significant unbalance classes problem. Hence, the data set will be directly adopted into the next step of analysis.

6. Preprocessing and Training Data Development

6.1 Preprocessing

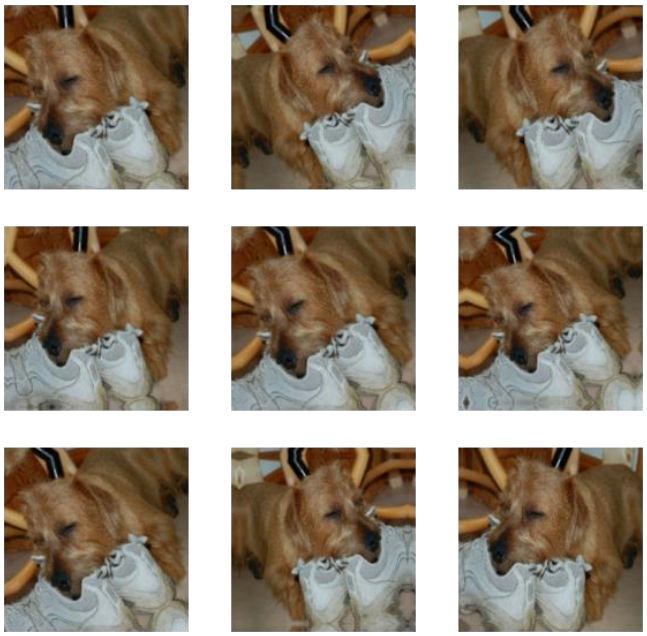
The preprocessing of data consists of image resize and augmentation:

Image resize

In order to ensure all image inputs are having the same image height and weight, all images are resized to 224 x 224. It is important to make sure all image size are the same before inputting the data to CNN model otherwise there may be problems for the model to convert all input into node of the neural network.

Image Augmentation

Data augmentation is a strategy to significantly increase the diversity of data available for training models, without actually collecting new data. Data augmentation techniques such as cropping, padding, and horizontal flipping are commonly used to train large neural networks.



The above figure demonstrates the idea of image augmentation. By this technique of nine different images (to CNN model) are generated from one

image source. Image augmentation can widen the database and reduce chance of overfitting.

6.2 Training Data Development

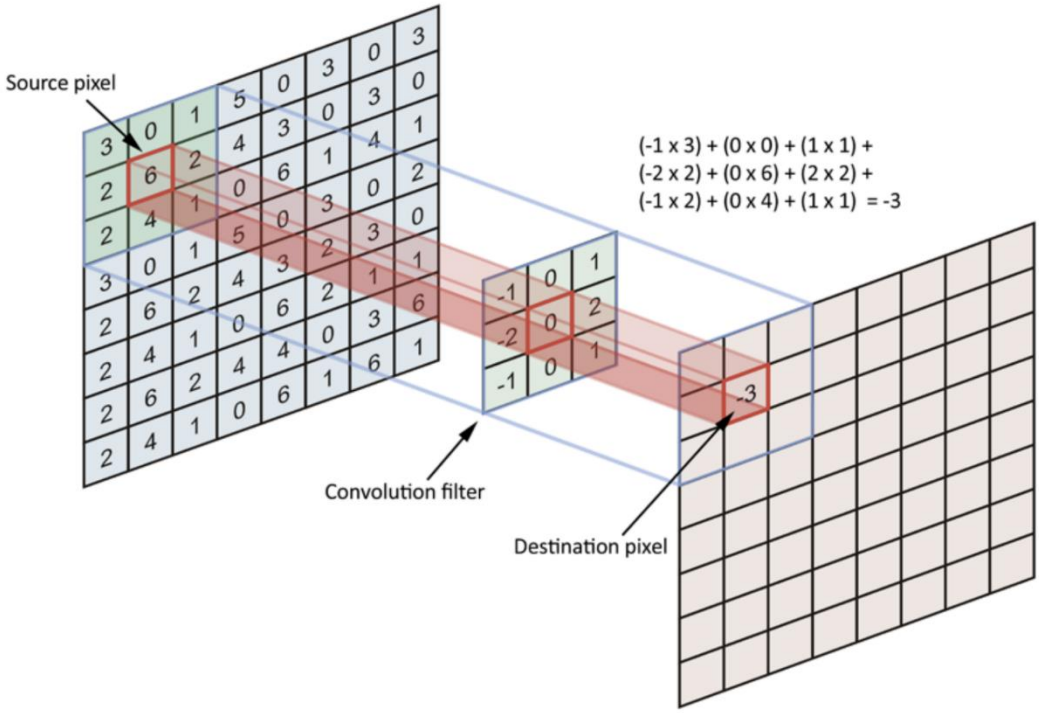
The last step before establishing the model is splitting the data into train set and test set. The train-test split is a technique for evaluating the performance of a machine learning algorithm. It can be used for classification or regression problems and can be used for any supervised learning algorithm. The procedure involves taking a dataset and dividing it into two subsets.

In this project, the dataset has been split to 90% of training data and 20% of testing data.

7. Modelling

The basic concept of Convolution Neural Network (CNN) model is as followed:

Step 1 – Apply filter and activation function to input image



Source: https://towardsdatascience.com/simple-introduction-to-convolutional-neural-networks-cdf8d3077bac

Convolution filters which contain features important for classification are mapped with each of the input images and produce feature maps. One filter will produce one feature map. After that, the feature maps will map with activation function to generate new feature maps that can be readable and optimized by the neural network model. The concept of neural network is plugging bias and weighted node to activation functions to generate and optimize a non-linear relationship to solve the classification problems. Each node in the neural network would have one bias, one weight and one activation function.

At the beginning, all filters are randomly selected and then they will be updated and optimized by backpropagation method in each epoch.

Step 2 – Pooling

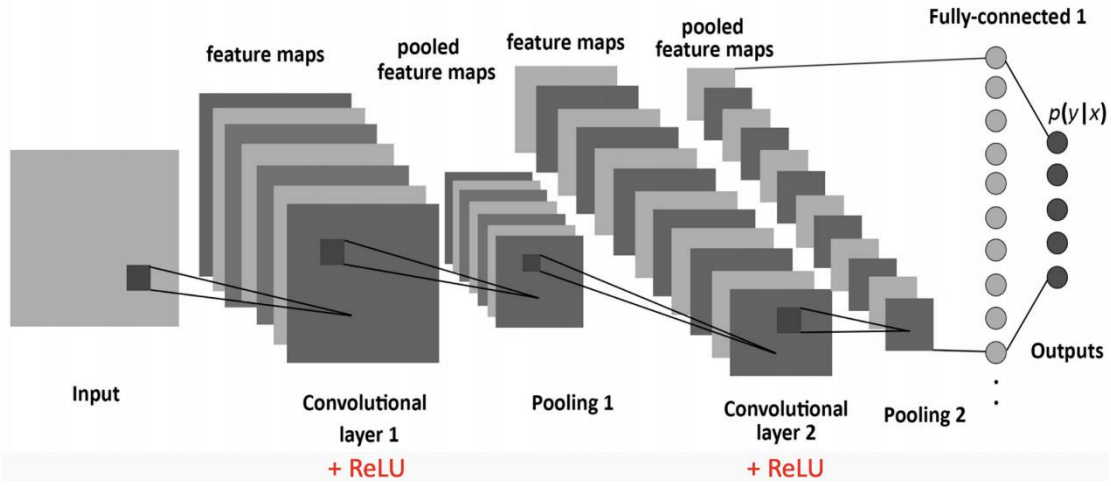
Pooling layers are similar to convolutional layers, but they perform a specific function such as max pooling, which takes the maximum value in a certain filter region, or average pooling, which takes the average value in a filter region. These are typically used to reduce the dimensionality of the network.

The pooling method adopted in this project is maximum pooling since this method can capture outliers from input images.

The pooling method can significantly reduce the number of nodes in the neural network model

Step 3 – Flattening (Fully Connected)

This step flattens all the processed image input to N rows x 1 columns and then all the N rows will become the first layer of node in the Neural Network Model.



Source: https://towardsdatascience.com/simple-introduction-to-convolutional-neural-networks-cdf8d3077bac

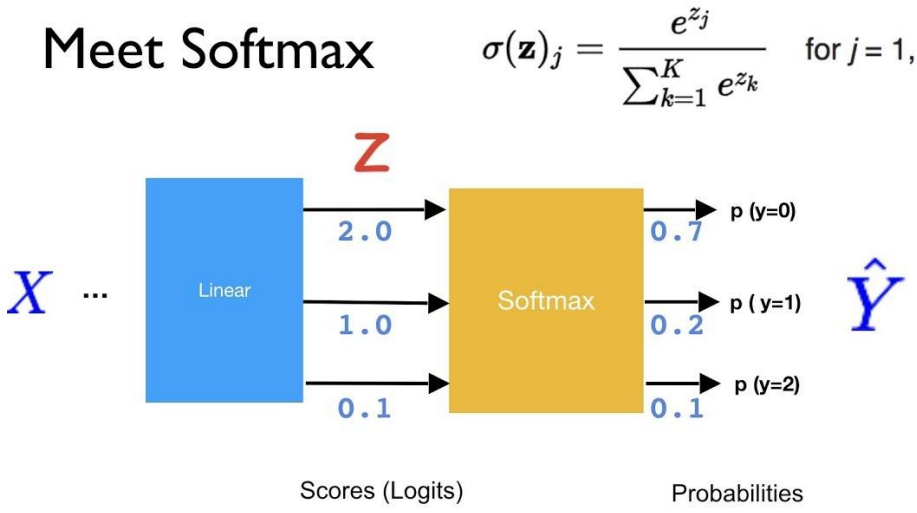
The above diagram shows an example of an CNN model. Usually,different combination of steps 1 and 2 will be adopted for extracting important feature from the image.

Step 4 – Model training and Backpropagation

After step 3, the structures of CNN has been completed and the next step is fitting in the data as well as training the model by updating the bias and weighting of each node in the network.

The concepts of training neural network is backpropagation which the model will try to reduce error by updating bias and weighting of each node from the result of previous epoch.

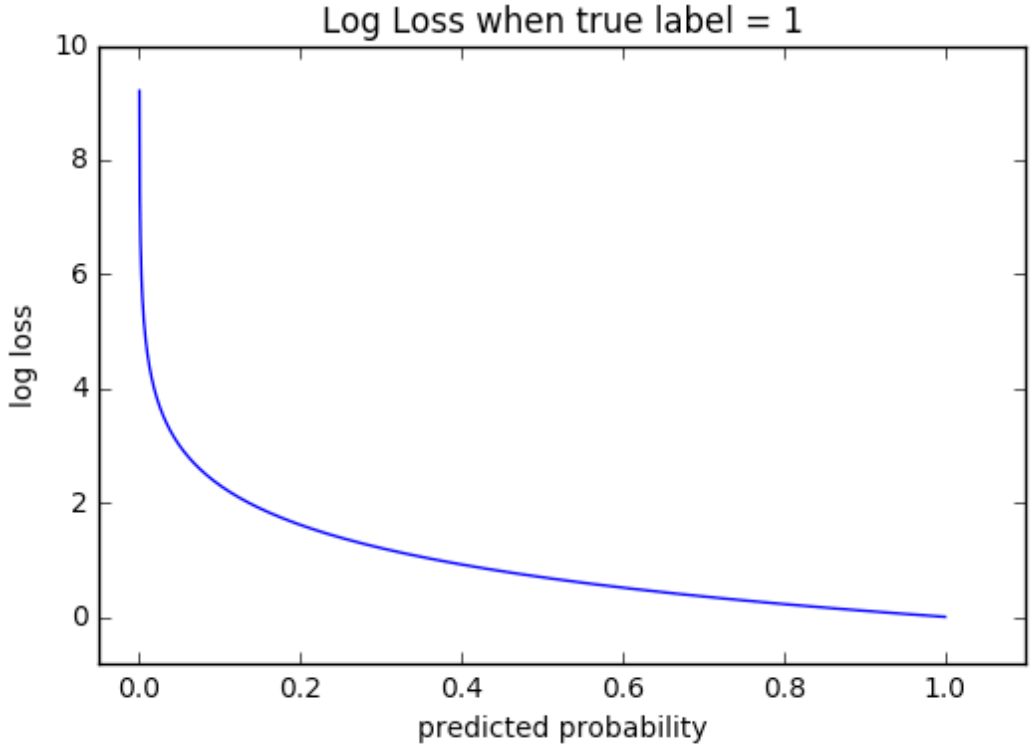
For classification problem, the concept of softmax and crossentropy are usually adopted for backpropagation of the model. The softmax function is used to map raw model output into probability of each class. It is important to highlight that, the softmax probability is sensitive to initial random guess of bias and weighting and the probability is only an approximation but not absolute.



Source: https://github.com/hunkim/PyTorchZeroToAll

The figure above shows the softmax equation and showing the process how this equation map all raw outputs to probabilities. The softmax equation can be applied as an activation function to the raw output and new layer of "probabilities" will be calculated.

Crossentropy is the subsequent step of softmax and the concept of it is to map all probabilities into the curve below:



From the above logit curve, the curve is flat when the probability is close to one and log loss rise significantly when the predicted probability go downs. In other words, the model will punish more heavily on worse prediction. With this curve, the model can effectively optimize the bias and weighting in each node.

Step 5 – Tunning of Hyperparameter

As the CNN mode is complicated, there are a lot of hyperparameters can be fine tined. The project enumerates some common tunning of hyperparameter as follow:

Hyperparameter	**Description**
No. of Epoch	No. of times updating bias and weighting of CNN
Learning rate	Step size of each epoch
Activation Function	Fitting function mapping input and output
Dropout	%, no. and position of network dropout
No. of layer	Combination of Step 1 & Step 2
Filter	**Size & padding of filters**

7.1 Self-Customized CNN Model

The first CNN model adopted in this project is self-customized as followed:

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
sequential (Sequential)      (None, 224, 224, 3)       0
_________________________________________________________________
rescaling (Rescaling)        (None, 224, 224, 3)       0
_________________________________________________________________
conv2d (Conv2D)              (None, 224, 224, 16)      448
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 112, 112, 16)      0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 112, 112, 32)      4640
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 56, 56, 32)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 56, 56, 64)        18496
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 28, 28, 64)        0
_________________________________________________________________
dropout (Dropout)            (None, 28, 28, 64)        0
_________________________________________________________________
flatten (Flatten)            (None, 50176)             0
_________________________________________________________________
dense (Dense)                (None, 128)               6422656
_________________________________________________________________
dense_1 (Dense)              (None, 20)                2580
=================================================================
Total params: 6,448,820
Trainable params: 6,448,820
Non-trainable params: 0
_________________________________________________________________
```

This model is relatively simple which consist of only 2 convolution and pooling layers. This model serves as a baseline and will compare with other models.

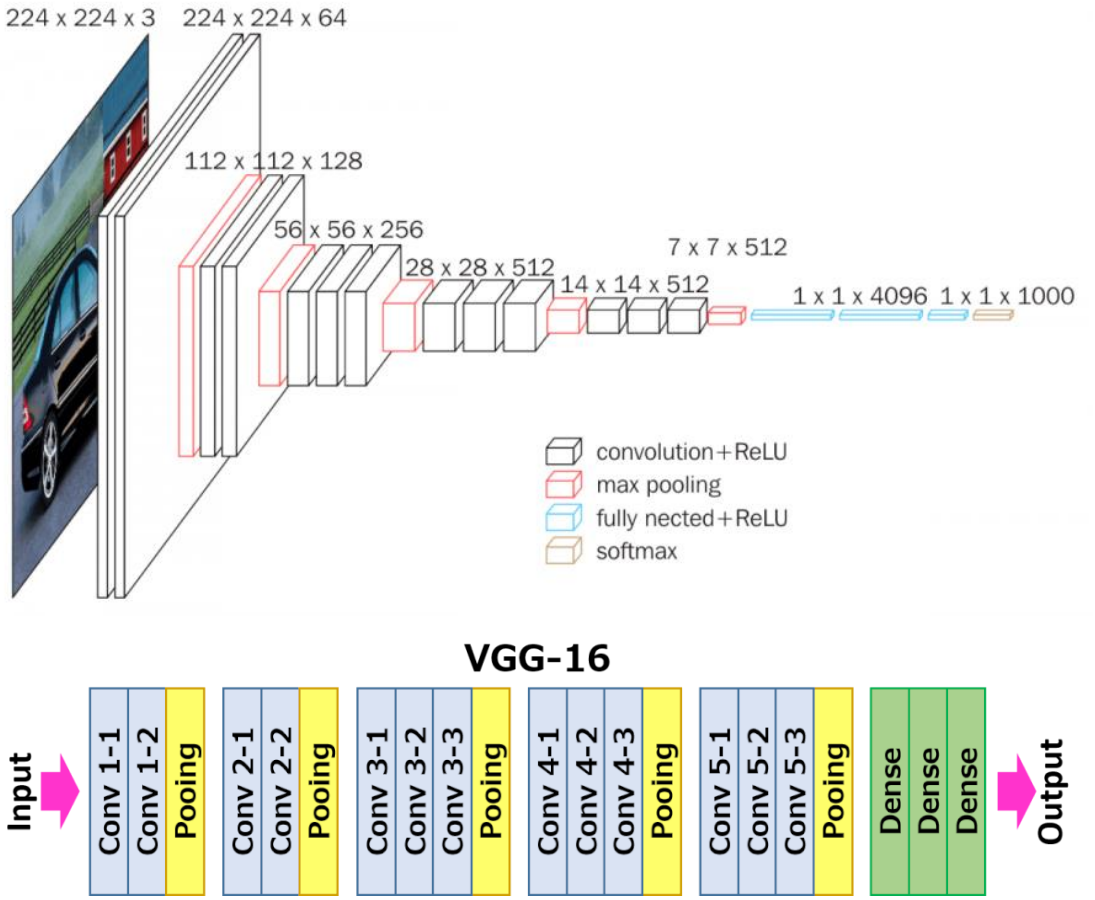
The result of the first model is as follow:

The accuracy of the model is 30.2% which performance is bad. However, this model can be used to compare the different between simple and more complex CNN model.

7.2 VGG16 Model

VGG16 is a convolutional neural network model proposed by K. Simonyan and A. Zisserman from the University of Oxford in the paper “Very Deep Convolutional Networks for Large-Scale Image Recognition”. The model achieves 92.7% top-5 test accuracy in ImageNet, which is a dataset of over 14 million images belonging to 1000 classes. It was one of the famous model submitted to ILSVRC-2014. It makes the improvement over AlexNet by replacing large kernel-sized filters (11 and 5 in the first and second

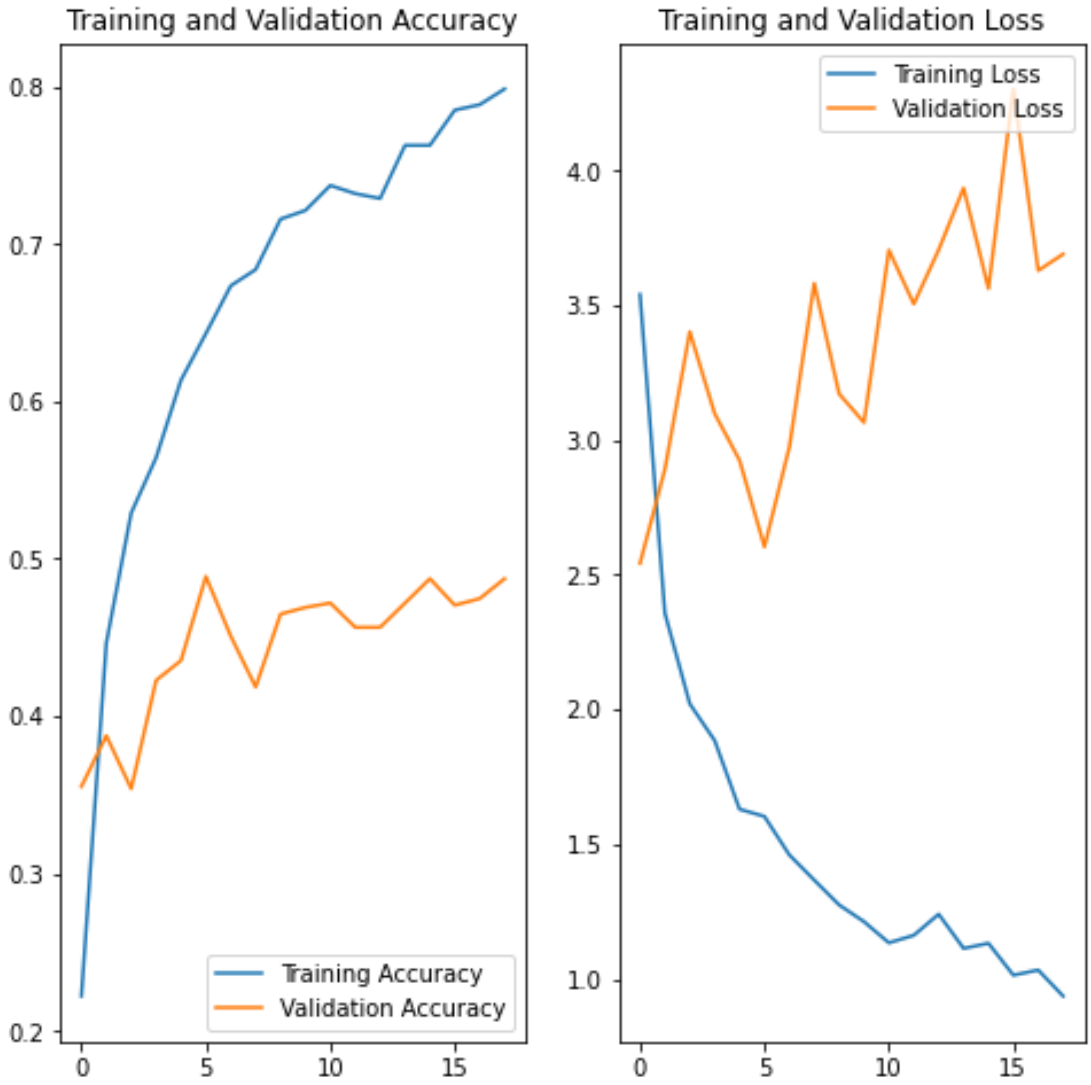
convolutional layer, respectively) with multiple 3×3 kernel-sized filters one after another.



Source: https://neurohive.io/en/popular-networks/vgg16/

The above figure demonstrates the structure of VGG16 model. The concept is the same as the first baseline model but the combination of convolution layers, pooling layers, filters are different.

The result is of the VGG16 models is as followed:

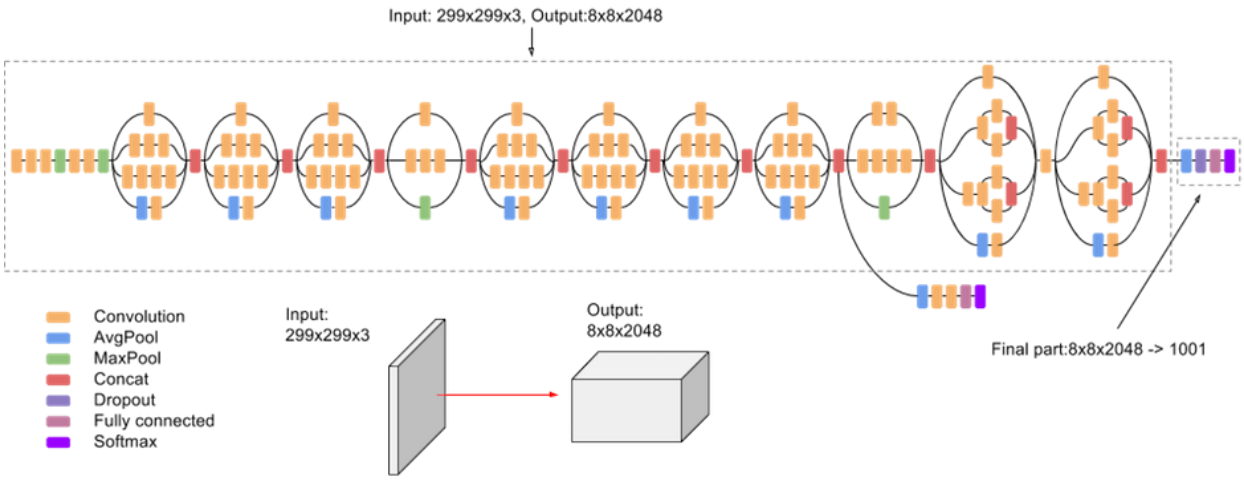


The accuracy of the model is 48.7% which has great increase from 30.2% of the first model. This indicates that with the same concept but different combination of layers, a CNN model will have great different.

7.3 Inception V3 model

Inception v3 is a widely-used image recognition model that has been shown to attain greater than 78.1% accuracy on the ImageNet dataset. The model is the culmination of many ideas developed by multiple researchers over the years. It is based on the original paper: "Rethinking the Inception Architecture for Computer Vision" by Szegedy, et. al. It is the third edition of Google's Inception Convolutional Neural Network, originally introduced during the ImageNet Recognition Challenge.

The model itself is made up of symmetric and asymmetric building blocks, including convolutions, average pooling, max pooling, concats, dropouts, and fully connected layers. Batchnorm is used extensively throughout the model and applied to activation inputs. Loss is computed via Softmax.

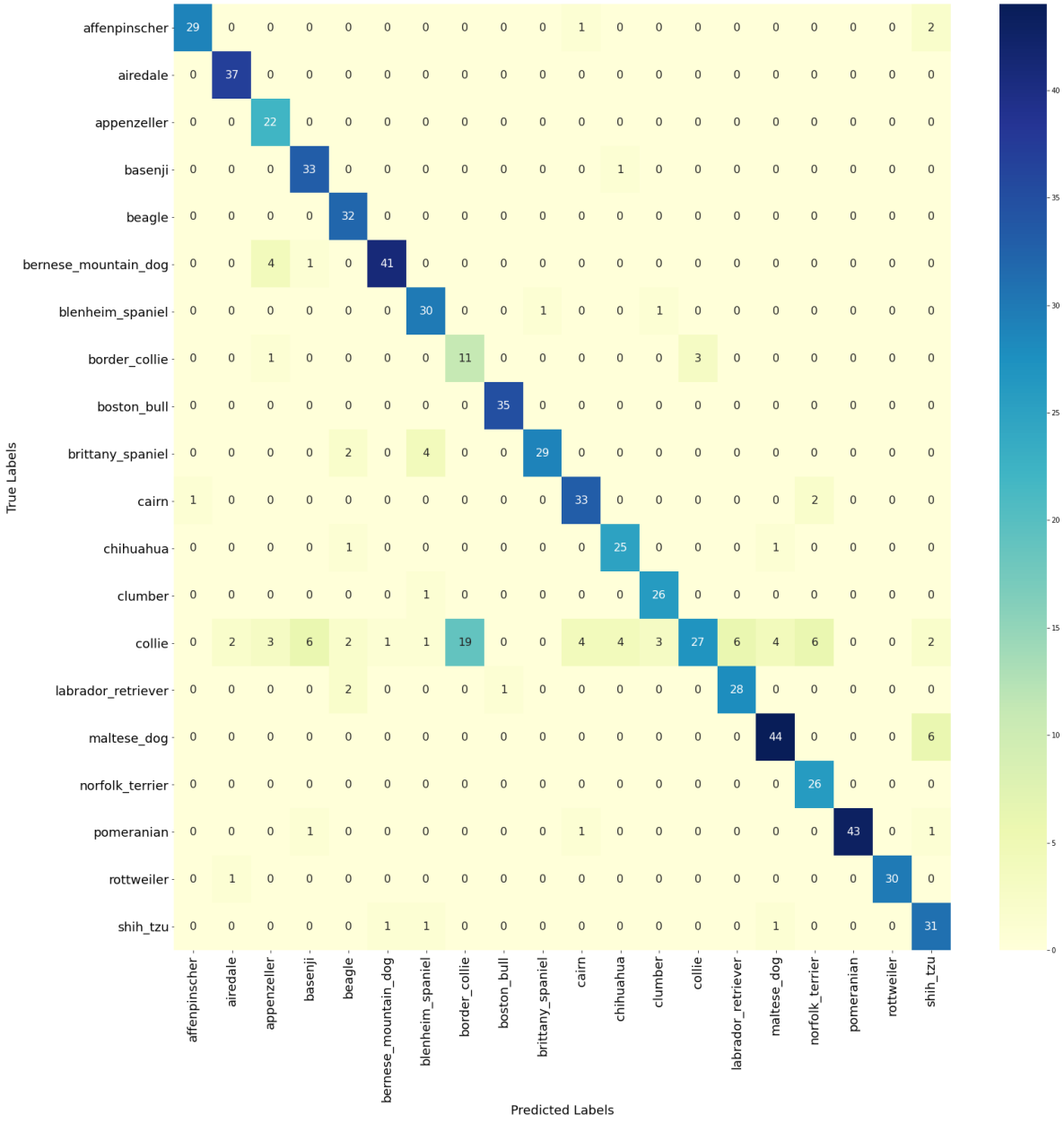


Source: https://cloud.google.com/tpu/docs/inception-v3-advanced

Inception V3 model is the most complex model in this project. Besides different combination of layers, Inception V3 model also adopts parallel block for computation.

The result of Inception V3 model is as followed:

The model has accuracy of 84.5% on the validation set of data which means with 84.5% of chance the Inception V3 models can correctly identify the dog breeds (for dogs among the 20 breeds in this project). This is the best model of the project and to further evaluate this model, confusion matrix and classification report can be explored.



From the confusion matrix, we can further evaluate the performance of the models and investigate any actions for improvement. For instance, from the confusion matrix, we can observe that the model is having problem to distinguish between collie and border collie.

	precision	recall	f1-score	support
0	0.97	0.91	0.94	32
1	0.93	1.00	0.96	37
2	0.73	1.00	0.85	22
3	0.80	0.97	0.88	34
4	0.82	1.00	0.90	32
5	0.95	0.89	0.92	46
6	0.81	0.94	0.87	32
7	0.37	0.73	0.49	15
8	0.97	1.00	0.99	35
9	0.97	0.83	0.89	35
10	0.85	0.92	0.88	36
11	0.83	0.93	0.88	27
12	0.87	0.96	0.91	27
13	0.90	0.30	0.45	90
14	0.82	0.90	0.86	31
15	0.88	0.88	0.88	50
16	0.76	1.00	0.87	26
17	1.00	0.93	0.97	46
18	1.00	0.97	0.98	31
19	0.74	0.91	0.82	34
accuracy			0.85	718
macro avg	0.85	0.90	0.86	718
weighted avg	0.87	0.85	0.84	718

The concept of confusion matrix is similar to confusion matrix. Precision and recall are adopted to evaluate the model. From the report, the recall of collie is only 0.3 which means the number of false negative is high for collie.

8. Result and discussion

The main findings and insights of the project are as follow:

1) This project adopts CNN model to identify 20 dog breeds from their images and obtains an accuracy of 84.5% on validation data.

2) The dog breeds classification is only one of many applications of the image classification model. This project aims to establish a baseline model and further transformation of the model can be performed to solve other daily life questions such as classifying different kinds of vehicles, animals, face recognition and even in medical use i.e. helping doctor to classify X-ray.

9. Future Studies

Due to constraints of time and resources, the project has both limitations and rooms for improvement. The directions of future studies can be as followed:

1) Due to limit of computation power, this project reduced the problem from 120 to 20 classes. Google Colab or other methods can be applied to solve the computation problems.

2) From confusion matrix and classification report, the classification are some classes such as collie and border collie are performed bad in the best mode. Further studies can be carried for further improvement.

3) More time and effort can be used for tuning of hyperparameter listed in Section 7 for further model improvement.

4) More problems can be solved by transforming models in this project. This project only provides baseline models for study and reference.

10. Conclusion

In conclusion, the project established a CNN model to classify 20 dog breeds from their images. The project demonstrates that with the same CNN concepts but different combination of layers and hyperparameters. The result of model will have huge different. Further models can be performed based on this project for improvement of classification accuracy or transforming this project to solve other classification problems