CMT 307
Applied Machine leaning

Fine-grained image classification of Dogs

Content

INTRODUCTION

DESCRIPTIVE ANALYSIS

DATA PREPARATION

IMPLEMENTATION OF MODEL

RESULTS

ERROR ANALYSIS

CONCLUSION

Introduction

- Fine-grained image recognition is the task of distinguishing between very similar objects such as identifying the species of a bird, the breed of a dog or the model of an aircraft.

- Outline the steps taken to develop an end-to-end machine learning pipeline used to develop a machine learning model which can categorise fine grained images of dogs into their subcategories (breeds).

- The Stanford Dog Dataset was used for this analysis.

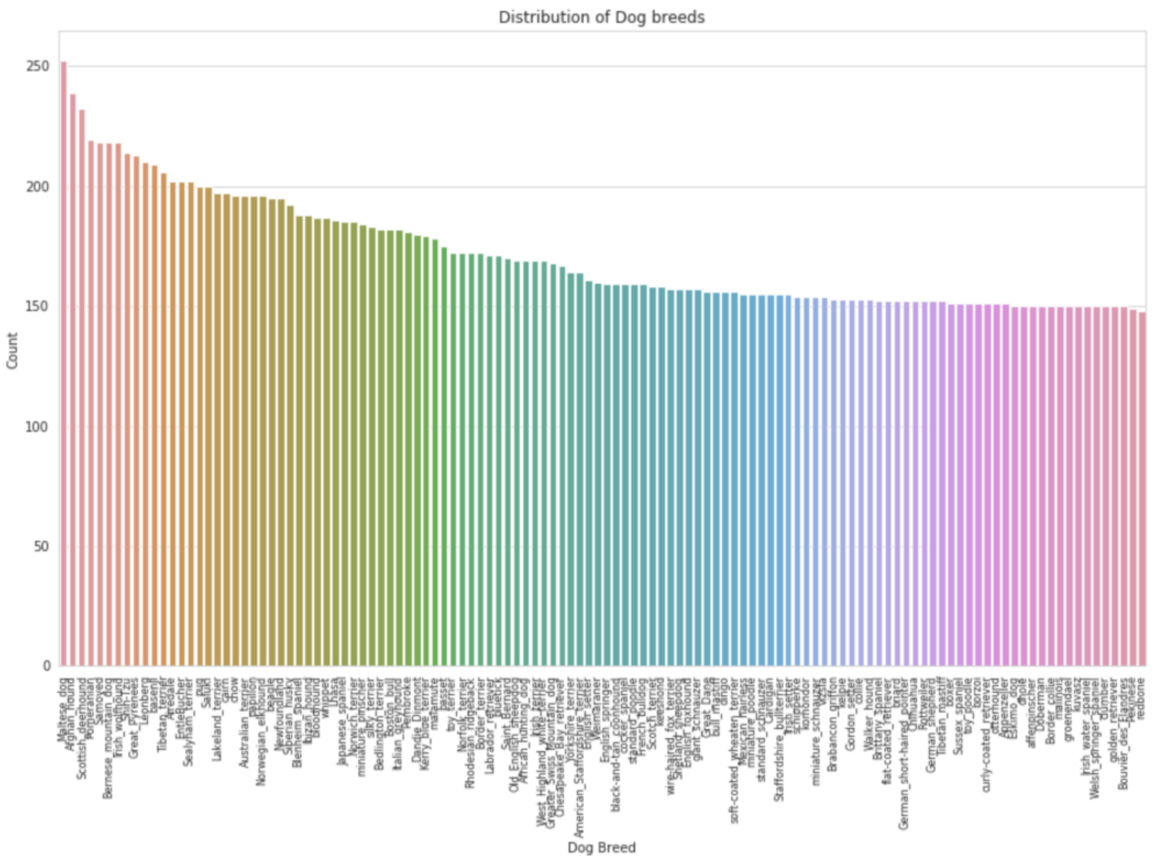
Source: https://medium.com/nanonets/how-to-easily-build-a-dog-breed-image-classification-model-2fd214419cde

Descriptive Analysis

Stanford Dog Breed Dataset

- 20580 Images
- 120 Dog Breeds
- Descriptive analysis of dog breed images:

	count	mean	std	min	25%	50%	75%	max
Count	120.0	171.5	23.220898	148.0	152.75	159.5	186.25	252.0



- Initial training set: 12000 Images
- Initial test set: 8580 Images

Image sizes:

- We have a large distribution of image sizes
- Majority of images are between:
    - A width of 361 and 500
    - A height of 333 to 453

- Huge differences between the maximum and minimum

- Images will have to be normalised

Image Size - Descriptive Statistics	***Width***	***Height***
mean	442.5	385.9
std	142.8	124.9
min	97	4
25%	361	333
50%	500	375
75%	500	453
max	3264	2562

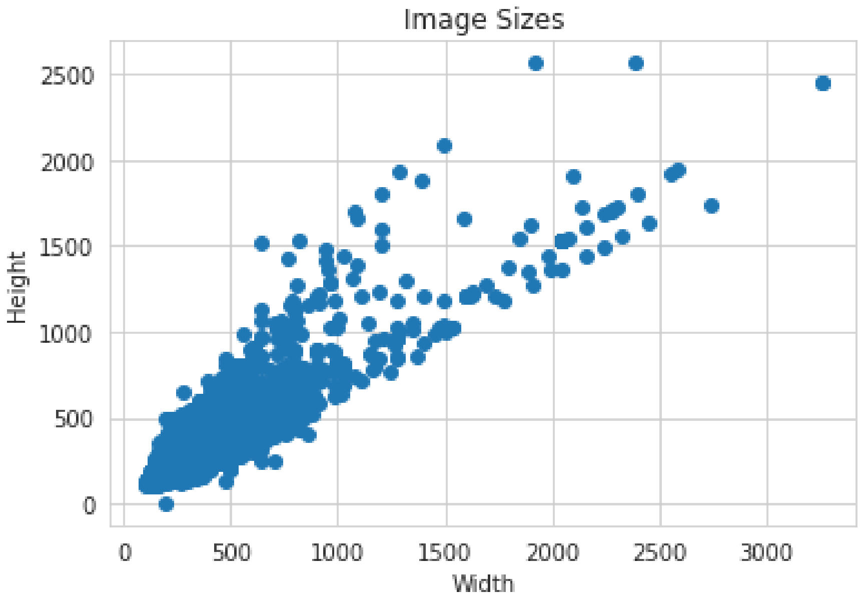
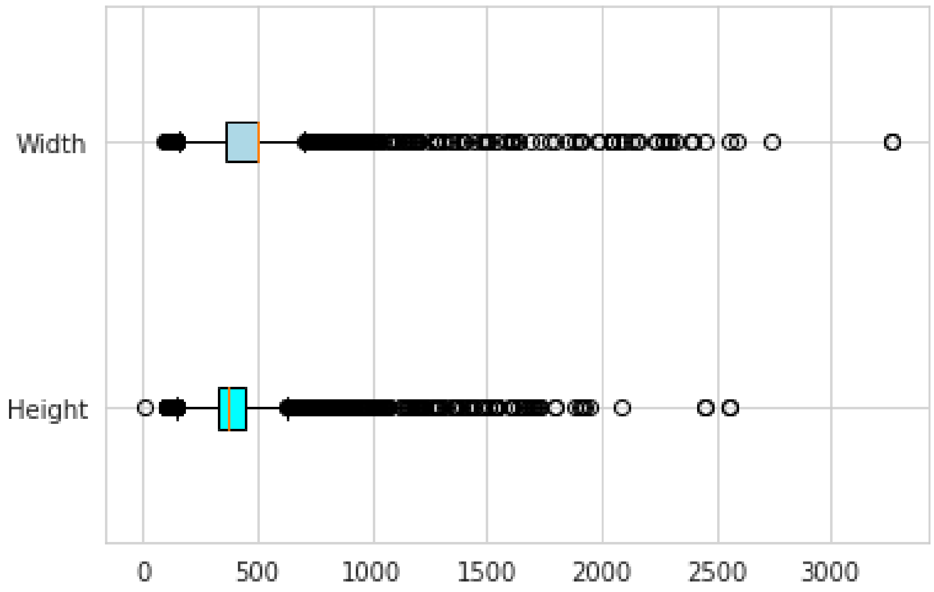


Image Size - Boxplot



Data Preparation

File preparation from ImageNet

- Images (757MB)
- Annotations (21MB)
- Lists, with train/test splits (0.5MB)
- Train Features (1.2GB), Test Features (850MB)

Dataset Split

Initial Split：

- A 60:40 ratio of *test* and *train* sets

References obtained：

- If the data-set is large enough one can consider ratios like 70:30 or 75:25, while in the case of small data-sets it is recommended to use ratios such as 90 : 10.

Final Split：

- Train, validation and test samples with proportions of 80%, 10% and 10% respectively

Image Pre-processing

Normalisation of the images：

It makes sure that each pixel has a similar distribution which will allow faster convergence when training the model.

Image augmentation:

It allows us to generate more training data by using our existing training data sets by transforming the original images providing more data for each class

Image Generator

- Rotation of 45-degrees to generate the images of the dog at a 45-degree angle.
- Width and height shift that randomly shifts the images left and right and up and down.
- Slanting the image.
- Zooming out
- Horizontally and vertically flipping the image
- Setting the fill mode to nearest which autofill's any empty pixels with the nearest pixel value.

ImageNet

- ImageNet is a project focused on labeling and categorising objects into 22,000 separate categories for the purpose of computer vision research.
- The models are trained on 1.2 million images, validated on 50,000 more and tested on 100,000 images.
- Image categories correspond to daily life object classes, such as dogs, cats, vehicle types etc.

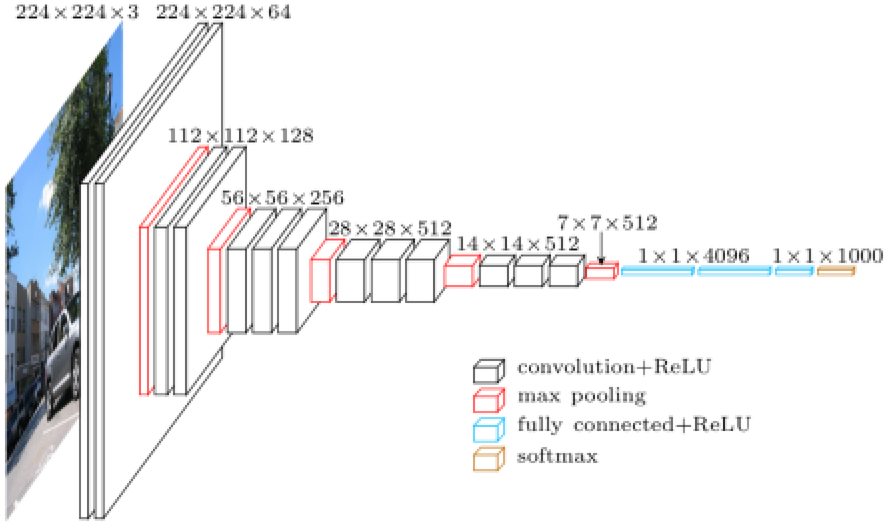
Overall

- Total number of non-empty synsets: 21841
- Total number of images: 14,197,122
- Number of images with bounding box annotations: 1,034,908
- Number of synsets with SIFT features: 1000
- Number of images with SIFT features: 1.2 million

Source: https://medium.com/syncedreview/sensetime-trains-imagenet-alexnet-in-record-1-5-minutes-e944ab049b2c

Vgg16

- It uses 3x3 convolutional layers stack on top of each other in increasing depth.
- Max pooling is responsible for reducing the volume size.
- There are 13 convolutional layers, 5 Max Pooling layers and 3 Dense layers (with 4,096 nodes and ReLu activation) which sums up to 21 layers
- At the end it has 2 fully-connected layers by a softmax classifier.
- It has 16 weight layers.

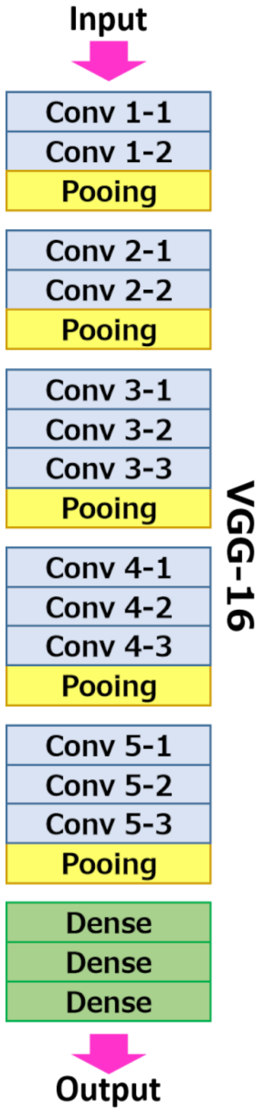


Advantages

- The Vgg16 network architecture its characterised by its simplicity compared to others.
- The model achieves 92.7% accuracy on the test set of ImageNet dataset.

Disadvantages

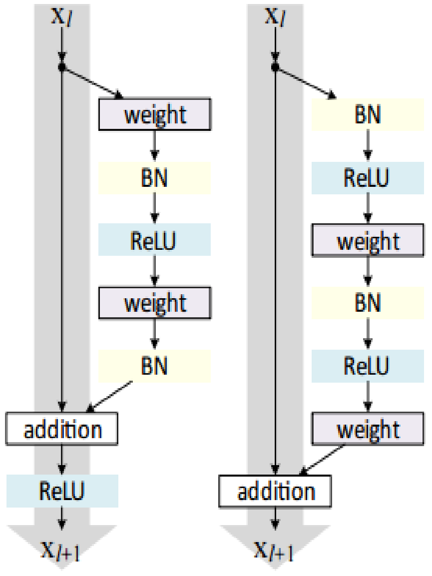
- Training and deploying the network is time consuming (approximately 138 million parameters).
- The architecture weights themselves are quite large in terms of disk/bandwidth.



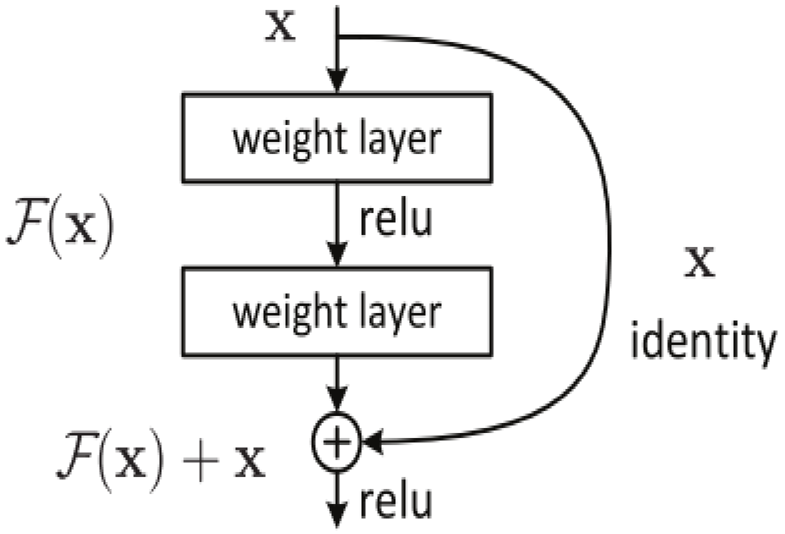
Source: https://neurohive.io/en/popular-networks/vgg16/

ResNet50

- ResNet-50 is a convolutional neural network that is 50 layers deep.
- Implements skip connections that do not simply go to the next layer, but instead propagate features from previous layers ahead in time.
- The network has an image input size of 224-by-224.
- Uses Global Average Pooling.
- Batch normalisation used after each convolutional and before activation.
- Batch size of 256.
- Learning rate starts from 0.1 and is divided by 10 when error plateaus.
- Trained for 60 x 10^4 iterations.
- Weight decay of 0.0001, momentum of 0.9.
- Does not use Dropout.
- Test-time augmentation: 10-crop testing.



(a) original (b) proposed

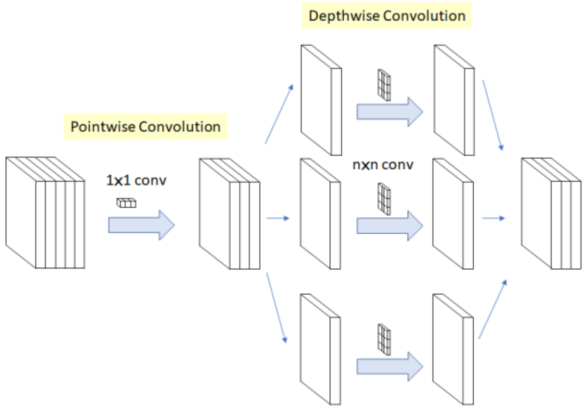


Advantages

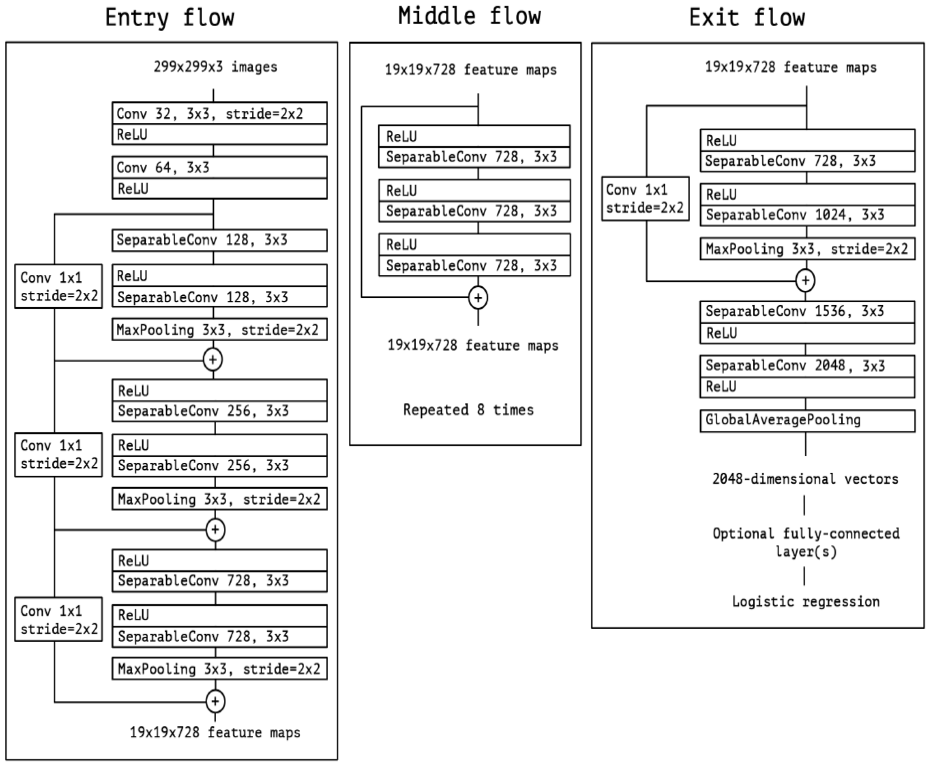
- Deeper than VggNet, with less computation.
- The model achieves 93.3% accuracy on the test set of ImageNet dataset.

Source: https://towardsdatascience.com/an-overview-of-resnet-and-its-variants-5281e2f56035

Xception



- Xception stands for the Extreme version of Inception and replaces the standard Inception modules with depth-wise separable convolutions.
- More efficient use of Inception's parameters.
- Xception is a convolutional neural network that is 71 layers deep.
- Implements residual skip connections that propagate features from previous layers ahead in time.
- The network has an image input size of 299-by-299.
- The initial number of channels is translated to n×n spatial convolutions.
- The 3×3 convolutional is done first, before any spatial convolutions.
- Uses intermediate ReLu activation function.



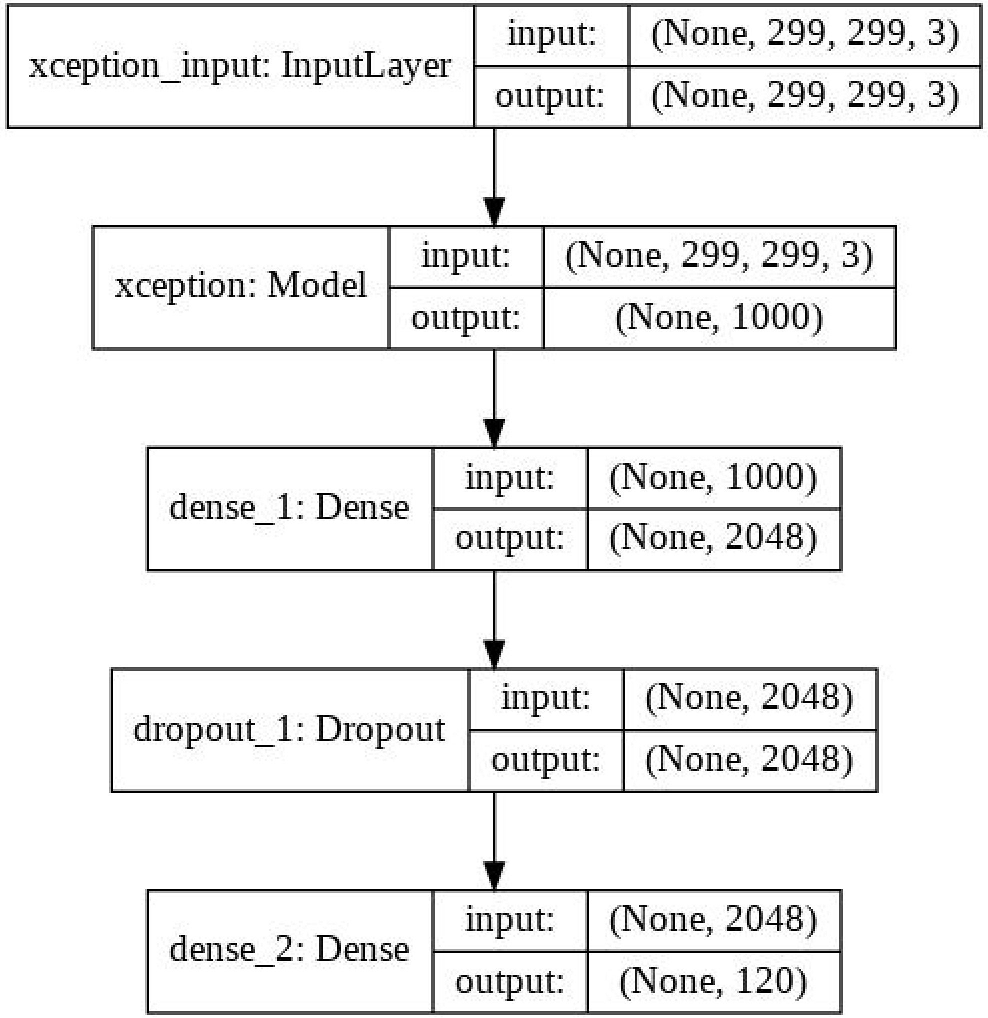
Advantages

- Slightly outperforms Inception V2 and significantly outperforms Inception V3 on the ImageNet dataset.
- The model achieves 94.5% accuracy on the test set of ImageNet dataset.
- The number of connections is smaller which leads to lighter model.

Source: https://towardsdatascience.com/review-xception-with-depthwise-separable-convolution-better-than-inception-v3-image-dc967dd42568

Final Model

- At the end of the Xception network, we added two Dense and one Dropout layer.



Results

Xception	Train set	Validation set	Test set
Accuracy	55%	90%	92%
Loss	2.1	0.4	0.23

Error Analysis

Misclassified Breeds:

- Some breeds have a lots of **similarities**, it's difficult to classify even by human
- Some images contains **more than one** dog breed

Whippet

Italian greyhound

Pitbull terrier and Irish Greyhound

Error Analysis

Bias and Variance

- Some dog breeds are initially having **much more images** than other

	Dog	Count
0	Maltese_dog	252
0	Afghan_hound	239
0	Scottish_deerhound	232
0	Pomeranian	219
0	Samoyed	218

	Dog	Count
0	clumber	150
0	golden_retriever	150
0	Bouvier_des_Flandres	150
0	Pekinese	149
0	redbone	148

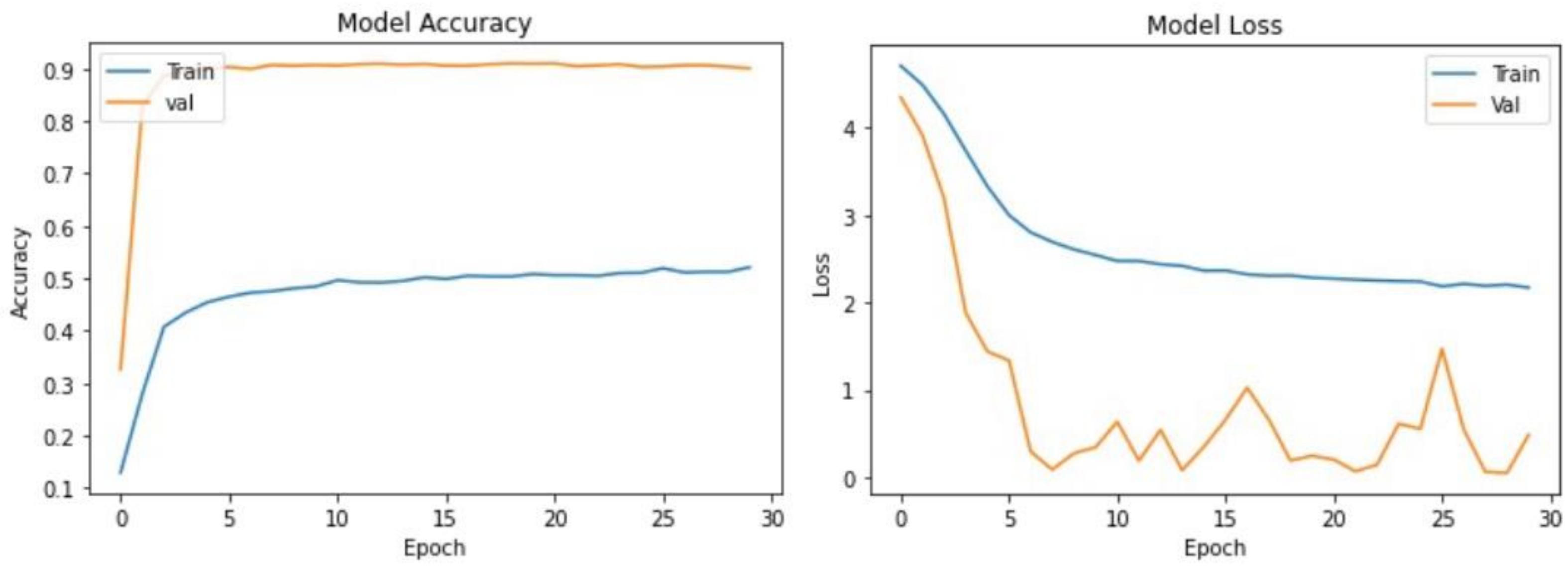
Mitigations:

- Make stable distribution for each breed
- Use image augmentation to **increase** the size of the training set
- Adopt sparse model

Error Analysis - Xception

Validation accuracy is much higher than training accuracy

- Because we adopt **drop-out layer**
- Better generalization and is less likely to overfit the training data
- More robust on the validation



Accuracy and Loss of Xception model in 30 epochs

Conclusion

Problem encountered

- Separations of tasks
- Underestimate the time of training a model on images
- Time differences with some members of the team

Future work

- Deeper Network Topology
- Explore how other algorithm tuning methods could affect the accuracy of the model such as **early stopping** or using different batch types and epochs.
- Try different resampling methods such as: **K-folds cross validation.**


Thank
You!