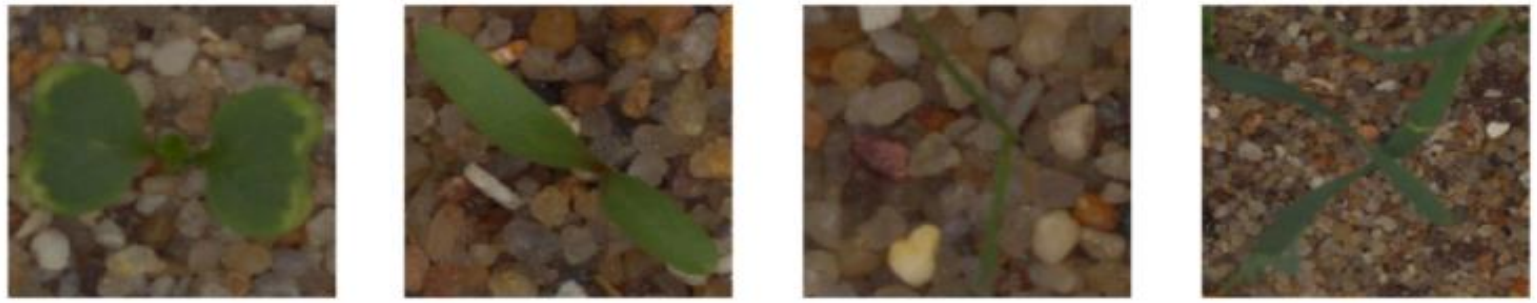
Plant Seedlings Classification

Using Transfer Learning



Plant Seedlings Classification - Description

- Classify the type of a plant seedling
- The data used for training consists of images of plant seedling taking from various angles, and is divided to 12 types (classes) as follows:

```
Black-grass 263 images
Charlock 390 images
Cleavers 287 images
Common Chickweed 611 images
Common wheat 221 images
Fat Hen 475 images
Loose Silky-bent 654 images
Maize 221 images
Scentless Mayweed 516 images
Shepherds Purse 231 images
Small-flowered Cranesbill 496 images
Sugar beet 385 images
```

- There is also a test file consisting of 794 images that we run our models on it and submit a csv file with label of each image to the kaggle competition site and get our score on this test images.

- This is a classification problem where upon a given image, we need to classify it to one of the 12 classes. So we will use a **Convolution Neural Network** for the task.
- It is almost practically inefficient to train a Convolution Neural Network from scratch.
- So, we take the weights of a pre trained CNN model on ImageNet with 1000 classes and keep the top layers frozen and replacing the last prediction layer with a dense layer consisting of 12 nodes instead of 1000 and train on only this last layer.
- This is because the top layers learn simple basic general features and we need not to train those layers and this pre-training can be directly applied to our task.
- This process is called **transfer learning**.

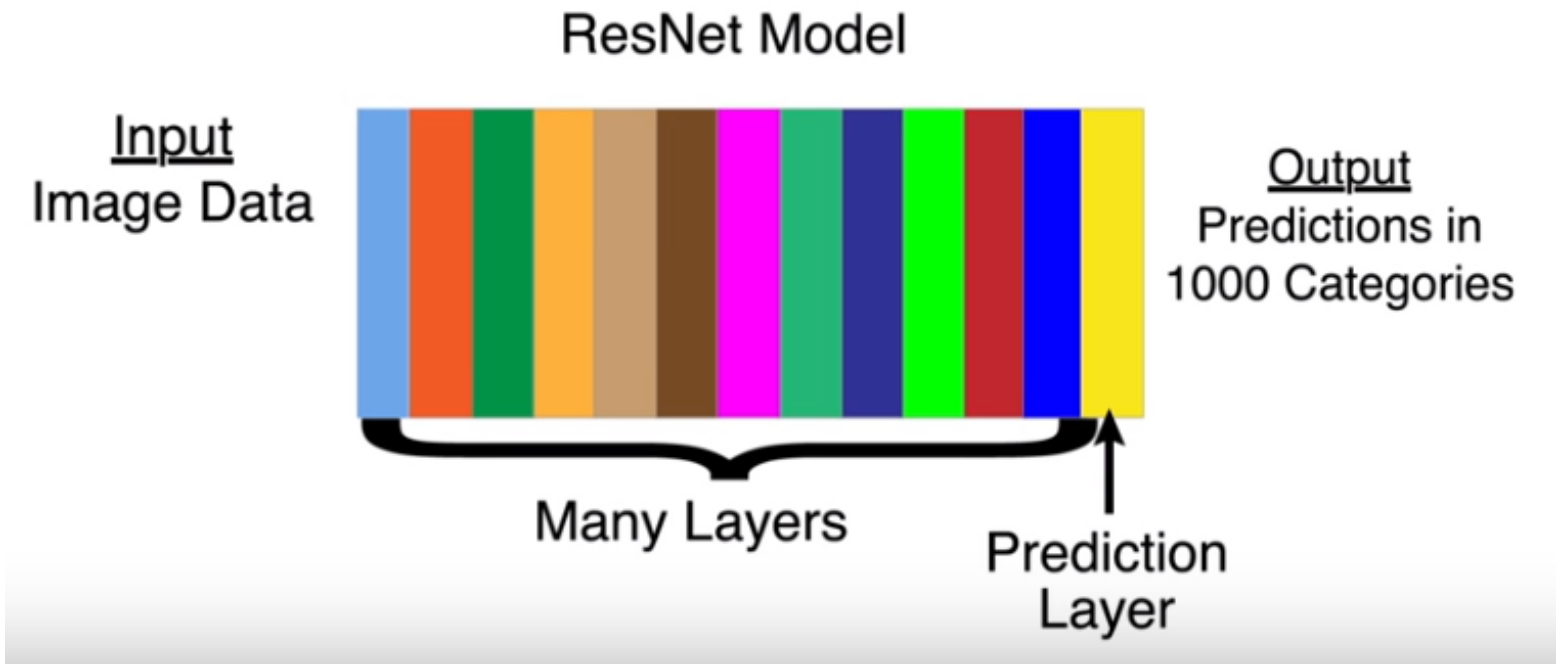
What is transfer learning ?

- Transfer learning takes what a model learned while solving one problem and applies it to a new application.
- Early layers of a deep learning model identify simple shapes, later layers identify more complex visual patterns and the very last layer makes predictions.
- Most layers from a pre trained model are useful in new applications because most computer vision problems involve similar low-level visual patterns

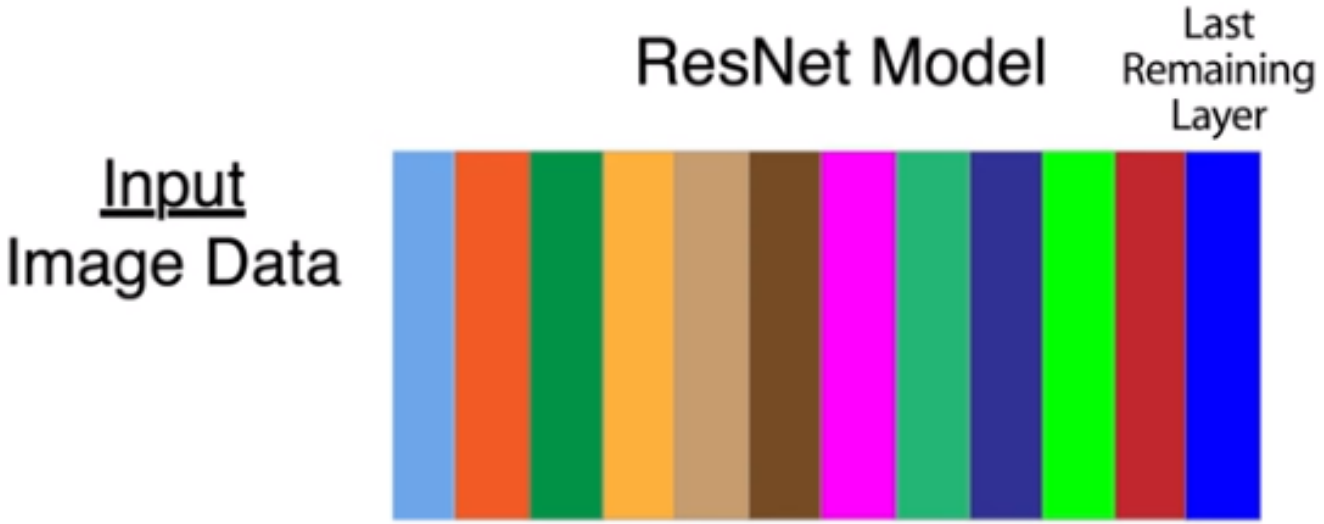
In our problem…

- We'll reuse all of the pre trained model and just replace the final layer that was used to make prediction.
- Some layers before the last layer in the pre trained that was trained on the ImageNet dataset, may identify features in **our** image dataset.
- We will drop in a replacement for the last layer in the pre trained model, this new last layer will predict to which class of 12 plant seedlings an image belongs to, based on the result of the previous layer.

A closer look (Resnet example)



We cut off the last layer, the last layer of what's left has information about our photo content stored as a series of numbers in a tensor.
It is a one-dimensional tensor i.e a vector.



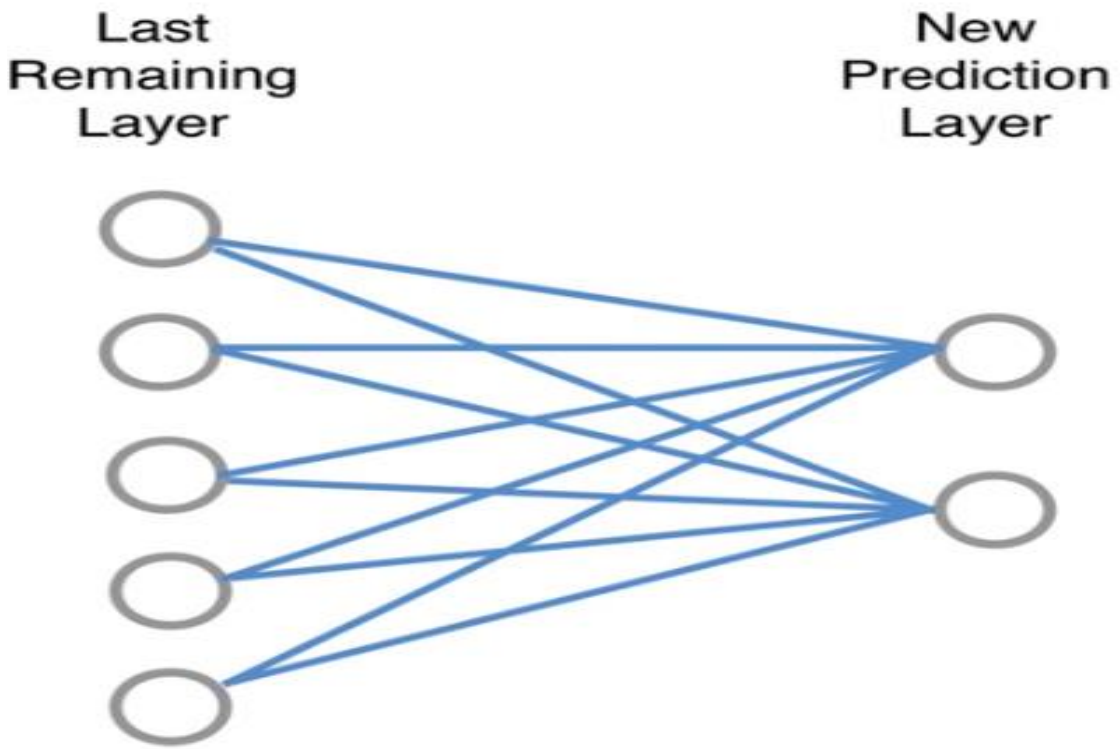
The image feature vector or the 'Bottleneck'

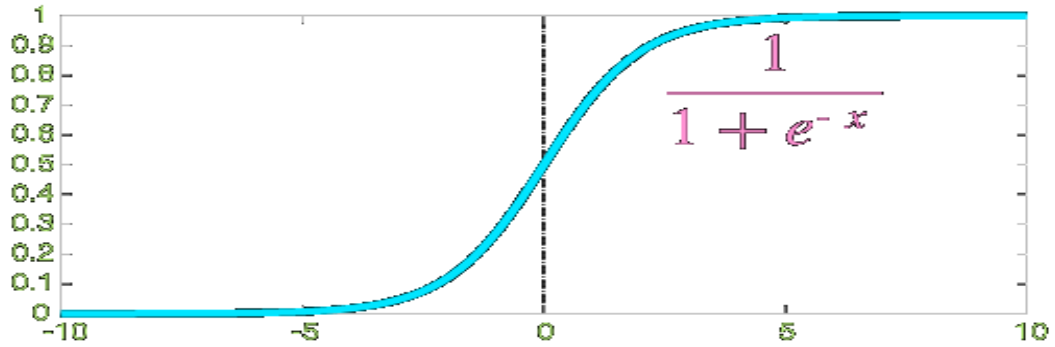
- 'Bottleneck' is an informal term often used for the layer just before the final output layer that actually does the classification.
- This penultimate layer has been trained to output a set of values that's good enough for the classifier to use to distinguish between all the classes it's been asked to recognize.
- That means it has to be a meaningful and compact summary of the images, since it has to contain enough information for the classifier to make a good choice on a very small set of values.
- The reason our final layer retraining can work on new classes is that it turns out the kind of information needed to distinguish between all the 1,000 classes in ImageNet is often also useful to distinguish between new kinds of objects.

Newly added classification layer

- We want to classify the image into 12 categories.
- After the last layer we keep the pre-trained model, we add a new layer with 12 nodes, a node for every category.
- Any node in the last layer before prediction might inform how much an image belongs to a certain category, so the measure of how much an image belongs to a certain category depends on all the nodes in this layer, this why the newly added layer is a dense (fully-connected) layer where each node in the new layer is connected to all the nodes in the last remaining layer in the pre-trained model.
- The structure would be as follows in the next slide.

The structure is drawn for two nodes alone but can be generalized for 12 nodes



- For all the connection drawn in the previous slide we'll use training data to determine which nodes suggest which category an image belongs to.
- That is we'll use the training data to train the last layer of the model.
- The training data will be photos that are labeled by each category of the 12 categories.
- We allow all features from the last remaining layer to influence or be connected with prediction layer i.e the last newly added layer is a dense layer (fully-connected).
- At the newly added layer we'll get a score for each category and then apply a function called **softmax.**
- The softmax function will transform the scores to probabilities so they'll all be positive and will sum to one.
- 

Our implementation

Train validation split

- Our data consists of 4750 images divides into 12 classes (see slide 2)
- Using the code in **split.py** we divided it so that the training images would be 80% and the validation images 20%
- We shuffled the images in each category before splitting it so that the validation images would be sampled as generally as possible.
- At the end of this process we had two directories:
- A directory called train for the training data that holds 3800 images (80% of all the images), divided into 12 categories represented by 12 subdirectory.
- A directory called train for the training data that holds 950 images (20% of all the images), divided into 12 categories represented by 12 subdirectory.

The training code

Explaining the parameters and their chosen values.

SGD for Neural Networks

parameters:

number of iterations τ

step size sequence $\eta_1, \eta_2, \ldots, \eta_\tau$

regularization parameter $\lambda > 0$

input:

layered graph (V, E)

differentiable activation function $\sigma : \mathbb{R} \to \mathbb{R}$

initialize:

choose $\mathbf{w}^{(1)} \in \mathbb{R}^{|E|}$ at random

(from a distribution s.t. $\mathbf{w}^{(1)}$ is close enough to $\mathbf{0}$)

for $i = 1, 2, \ldots, \tau$

sample $(\mathbf{x}, \mathbf{y}) \sim \mathcal{D}$

calculate gradient $\mathbf{v}_i = \texttt{backpropagation}(\mathbf{x}, \mathbf{y}, \mathbf{w}, (V, E), \sigma)$

update $\mathbf{w}^{(i+1)} = \mathbf{w}^{(i)} - \eta_i(\mathbf{v}_i + \lambda \mathbf{w}^{(i)})$

output:

$\bar{\mathbf{w}}$ is the best performing $\mathbf{w}^{(i)}$ on a validation set

Backpropagation

input:

example $(\mathbf{x}, \mathbf{y})$, weight vector $\mathbf{w}$, layered graph (V, E),

activation function $\sigma : \mathbb{R} \to \mathbb{R}$

initialize:

denote layers of the graph $V_0, \ldots, V_T$ where $V_t = \{v_{t,1}, \ldots, v_{t,k_t}\}$

define $W_{t,i,j}$ as the weight of $(v_{t,j}, v_{t+1,i})$

(where we set $W_{t,i,j} = 0$ if $(v_{t,j}, v_{t+1,i}) \notin E$)

forward:

set $\mathbf{o}_0 = \mathbf{x}$

for $t = 1, \ldots, T$

for $i = 1, \ldots, k_t$

set $a_{t,i} = \sum_{j=1}^{k_{t-1}} W_{t-1,i,j}\, o_{t-1,j}$

set $o_{t,i} = \sigma(a_{t,i})$

backward:

set $\boldsymbol{\delta}_T = \mathbf{o}_T - \mathbf{y}$

for $t = T-1, T-2, \ldots, 1$

for $i = 1, \ldots, k_t$

$\delta_{t,i} = \sum_{j=1}^{k_{t+1}} W_{t,j,i}\, \delta_{t+1,j}\, \sigma'(a_{t+1,j})$

output:

foreach edge $(v_{t-1,j}, v_{t,i}) \in E$

set the partial derivative to $\delta_{t,i}\, \sigma'(a_{t,i})\, o_{t-1,j}$

Intuitive explanation of SGD

- We used stochastic gradient descent to set the weights that minimize our loss function.
- stochastic gradient descent “feels” which direction goes downhill most steeply on the loss function (with regards to weights values) and take a step in that direction, then it “feels” around again to find which direction is downhill and it takes another step, it repeats this processes until it can’t go down anymore.
- This is basically how gradient descent works, it looks at data and checks which weights it can change to get a little lower loss function and it changes the weights slightly in that direction then it repeats this to improve the weights slightly again.
- How does it find which way it can change the weights to improve the loss function ? Basically how does it find which way goes downhill ? For that it uses **backward propagation** which calculates the gradient.

More details about SGD

- It generally doesn't use all of the data to calculate each step doing so would require a lot of calculations so it would be slow.
- Instead it looks at some of the data at a time, the **batch size** is the number of images used to calculate each step.
- It takes one small batch and then the next until it used all of the data.
- One time through the data is called an **epoch.**
- it incrementally improve weights for multiple epochs so each image would be used to improve weights more than once.
- **Backpropagation** is the process by which we find out which way to change the weights at each step of gradient descent.

The learning rate and Adam optimizer

- The size of weight changes is determined by the **learning rate.**
- Low learning rates mean the model may take a long time training before it gets accurate, high learning rates mean the model may take huge steps around in a field always jumping over the best weights and never getting very accurate.
- We used the Adam optimizer which is a special variation of gradient descent that automatically figures out the best learning rate throughout the gradient descent process.

Choosing the values of batch size for training images and validation images

- Quoting the paper On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima:
- “It has been observed in practice that when using a larger batch there is a degradation in the quality of the model, as measured by its ability to generalize”
- “large-batch methods tend to converge to sharp minimizers of the training and testing functions—and as is well known, sharp minima lead to poorer generalization. n. In contrast, small-batch methods consistently converge to flat minimizers, and our experiments support a commonly held view that this is due to the inherent noise in the gradient estimation.”
- So we decided to stick with low batch size between 1-32

Choosing the values of batch size for training images and validation images

- Our training data consists of 3800 images and we need to apply the following equation : 3800 = batch_size * steps_per_epch
- Our validation data consists of 950 images and we need to apply the following equation : 950= batch_size * validation_steps.

Input interpretation:

divisors	3800

Divisors:

1 | 2 | 4 | 5 | 8 | 10 | 19 | 20 | 25 | 38 | 40 | 50 | 76 | 95 | 100 | 152 | 190 | 200 | 380 | 475 | 760 | 950 | 1900 | 3800 (24 divisors)

Input interpretation:

divisors	950

Divisors:

1 | 2 | 5 | 10 | 19 | 25 | 38 | 50 | 95 | 190 | 475 | 950 (12 divisors)

Number of epochs and image size

- The number of epochs was set to 10 as we saw the accuracy scores for the validation converges around that number.

- The image size was chosen according to the default image size of the pre-trained model.

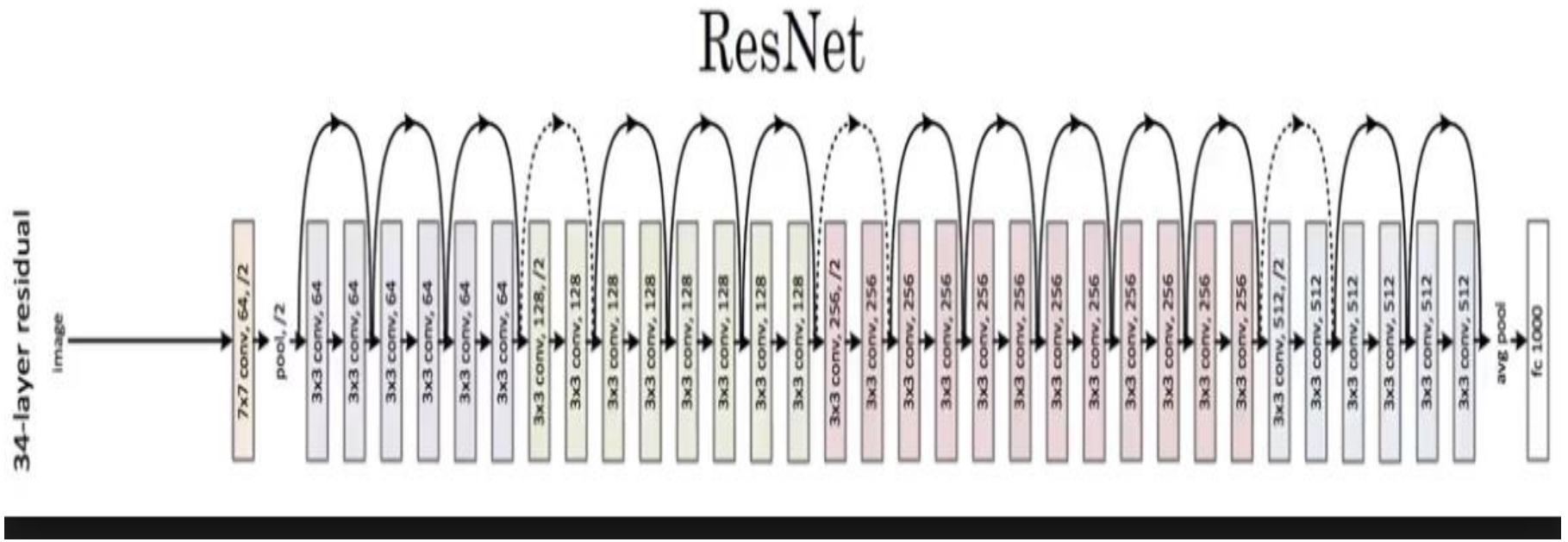
Results on different pre-trained models for the ImageNet competition.

Model	Size	Top-1 Accuracy	Top-5 Accuracy	Parameters	Depth
Xception	88 MB	0.790	0.945	22,910,480	126
VGG16	528 MB	0.713	0.901	138,357,544	23
VGG19	549 MB	0.713	0.900	143,667,240	26
ResNet50	99 MB	0.749	0.921	25,636,712	168
InceptionV3	92 MB	0.779	0.937	23,851,784	159
InceptionResNetV2	215 MB	0.803	0.953	55,873,736	572
MobileNet	16 MB	0.704	0.895	4,253,864	88
MobileNetV2	14 MB	0.713	0.901	3,538,984	88
DenseNet121	33 MB	0.750	0.923	8,062,504	121
DenseNet169	57 MB	0.762	0.932	14,307,880	169
DenseNet201	80 MB	0.773	0.936	20,242,984	201
NASNetMobile	23 MB	0.744	0.919	5,326,716	-
NASNetLarge	343 MB	0.825	0.960	88,949,818	-

Results

First we try the resnet50 model on different batch sizes so we can decide on the best batch size to use.

Resnet 50
The default input size for this model is 224x224.



Resnet 50 with train batch size = 2, validation batch size = 2, steps per epoch = 1900, validation steps = 475, epochs = 10

```
Layer (type)                 Output Shape              Param #
=================================================================
resnet50 (Model)             (None, 2048)              23587712
_________________________________________________________________
dense (Dense)                (None, 12)                24588
=================================================================
Total params: 23,612,300
Trainable params: 24,588
Non-trainable params: 23,587,712
_________________________________________________________________
Found 3800 images belonging to 12 classes.
Found 950 images belonging to 12 classes.
Epoch 1/10
1900/1900 [==============================] - 86s 45ms/step - loss: 1.9056 - acc: 0.3782 - val_loss: 1.8537 - val_acc: 0.6000
Epoch 2/10
1900/1900 [==============================] - 81s 43ms/step - loss: 1.3869 - acc: 0.5450 - val_loss: 2.1127 - val_acc: 0.6263
Epoch 3/10
1900/1900 [==============================] - 82s 43ms/step - loss: 1.2230 - acc: 0.5987 - val_loss: 1.9660 - val_acc: 0.6579
Epoch 4/10
1900/1900 [==============================] - 82s 43ms/step - loss: 1.1000 - acc: 0.6392 - val_loss: 2.3251 - val_acc: 0.6126
Epoch 5/10
1900/1900 [==============================] - 83s 44ms/step - loss: 1.0332 - acc: 0.6658 - val_loss: 1.9052 - val_acc: 0.6737
Epoch 6/10
1900/1900 [==============================] - 84s 44ms/step - loss: 0.9121 - acc: 0.7029 - val_loss: 1.5977 - val_acc: 0.6895
Epoch 7/10
1900/1900 [==============================] - 82s 43ms/step - loss: 0.9059 - acc: 0.6992 - val_loss: 1.8688 - val_acc: 0.6653
Epoch 8/10
1900/1900 [==============================] - 82s 43ms/step - loss: 0.9021 - acc: 0.6947 - val_loss: 1.7500 - val_acc: 0.6832
Epoch 9/10
1900/1900 [==============================] - 82s 43ms/step - loss: 0.8262 - acc: 0.7263 - val_loss: 1.8861 - val_acc: 0.6747
Epoch 10/10
1900/1900 [==============================] - 82s 43ms/step - loss: 0.7717 - acc: 0.7392 - val_loss: 2.1033 - val_acc: 0.6695
```

Name	Submitted	Wait time	Execution time	Score
resnet50_batch_size_2_2.csv	a few seconds ago	0 seconds	0 seconds	0.71914

Complete

Resnet 50 with train batch size = 10, validation batch size = 10, steps per epoch = 380, validation steps = 95, epochs = 10

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
resnet50 (Model)             (None, 2048)              23587712
_________________________________________________________________
dense (Dense)                (None, 12)                24588
=================================================================
Total params: 23,612,300
Trainable params: 24,588
Non-trainable params: 23,587,712
_________________________________________________________________
Found 3800 images belonging to 12 classes.
Found 950 images belonging to 12 classes.
Epoch 1/10
380/380 [==============================] - 52s 138ms/step - loss: 1.3653 - acc: 0.5692 - val_loss: 0.8043 - val_acc: 0.7611
Epoch 2/10
380/380 [==============================] - 49s 130ms/step - loss: 0.7484 - acc: 0.7529 - val_loss: 0.7289 - val_acc: 0.7663
Epoch 3/10
380/380 [==============================] - 50s 130ms/step - loss: 0.5913 - acc: 0.8039 - val_loss: 0.5516 - val_acc: 0.8211
Epoch 4/10
380/380 [==============================] - 49s 130ms/step - loss: 0.5055 - acc: 0.8326 - val_loss: 0.5669 - val_acc: 0.8137
Epoch 5/10
380/380 [==============================] - 50s 131ms/step - loss: 0.4398 - acc: 0.8518 - val_loss: 0.5890 - val_acc: 0.8147
Epoch 6/10
380/380 [==============================] - 50s 130ms/step - loss: 0.4066 - acc: 0.8574 - val_loss: 0.5933 - val_acc: 0.8021
Epoch 7/10
380/380 [==============================] - 50s 130ms/step - loss: 0.3734 - acc: 0.8782 - val_loss: 0.5638 - val_acc: 0.8042
Epoch 8/10
380/380 [==============================] - 50s 131ms/step - loss: 0.3442 - acc: 0.8834 - val_loss: 0.5309 - val_acc: 0.8326
Epoch 9/10
380/380 [==============================] - 51s 135ms/step - loss: 0.3049 - acc: 0.8974 - val_loss: 0.5015 - val_acc: 0.8358
Epoch 10/10
380/380 [==============================] - 50s 133ms/step - loss: 0.3019 - acc: 0.8982 - val_loss: 0.5668 - val_acc: 0.8263
```

Name	Submitted	Wait time	Execution time	Score
resnet50_batch_size_10_10.csv	a few seconds ago	0 seconds	0 seconds	0.78589

Complete

Resnet 50 with train batch size = 25, validation batch size = 25, steps per epoch = 152, validation steps = 38, epochs = 10

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
resnet50 (Model)             (None, 2048)              23587712
_________________________________________________________________
dense (Dense)                (None, 12)                24588
=================================================================
Total params: 23,612,300
Trainable params: 24,588
Non-trainable params: 23,587,712
_________________________________________________________________
Found 3800 images belonging to 12 classes.
Found 950 images belonging to 12 classes.
Epoch 1/10
152/152 [==============================] - 47s 308ms/step - loss: 1.4409 - acc: 0.5432 - val_loss: 0.9258 - val_acc: 0.6863
Epoch 2/10
152/152 [==============================] - 44s 289ms/step - loss: 0.7276 - acc: 0.7829 - val_loss: 0.6769 - val_acc: 0.7874
Epoch 3/10
152/152 [==============================] - 43s 286ms/step - loss: 0.5459 - acc: 0.8418 - val_loss: 0.6174 - val_acc: 0.7905
Epoch 4/10
152/152 [==============================] - 44s 286ms/step - loss: 0.4597 - acc: 0.8624 - val_loss: 0.5572 - val_acc: 0.8168
Epoch 5/10
152/152 [==============================] - 44s 286ms/step - loss: 0.3846 - acc: 0.8916 - val_loss: 0.5599 - val_acc: 0.8179
Epoch 6/10
152/152 [==============================] - 44s 287ms/step - loss: 0.3378 - acc: 0.8979 - val_loss: 0.5319 - val_acc: 0.8168
Epoch 7/10
152/152 [==============================] - 44s 287ms/step - loss: 0.2968 - acc: 0.9174 - val_loss: 0.5147 - val_acc: 0.8274
Epoch 8/10
152/152 [==============================] - 44s 288ms/step - loss: 0.2691 - acc: 0.9268 - val_loss: 0.4752 - val_acc: 0.8389
Epoch 9/10
152/152 [==============================] - 44s 289ms/step - loss: 0.2389 - acc: 0.9339 - val_loss: 0.4907 - val_acc: 0.8326
Epoch 10/10
152/152 [==============================] - 44s 287ms/step - loss: 0.2268 - acc: 0.9355 - val_loss: 0.4781 - val_acc: 0.8284
```

Name	Submitted	Wait time	Execution time	Score
resnet50_batch_size_25_25.csv	a few seconds ago	0 seconds	0 seconds	0.85138

Complete

Resnet 50 with train batch size = 38, validation batch size = 38, steps per epoch = 100, validation steps = 25, epochs = 10

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
resnet50 (Model)             (None, 2048)              23587712
_________________________________________________________________
dense (Dense)                (None, 12)                24588
=================================================================
Total params: 23,612,300
Trainable params: 24,588
Non-trainable params: 23,587,712
_________________________________________________________________
Found 3800 images belonging to 12 classes.
Found 950 images belonging to 12 classes.
Epoch 1/10
100/100 [==============================] - 48s 477ms/step - loss: 1.5531 - acc: 0.5089 - val_loss: 1.1207 - val_acc: 0.6105
Epoch 2/10
100/100 [==============================] - 44s 438ms/step - loss: 0.7736 - acc: 0.7737 - val_loss: 0.7693 - val_acc: 0.7632
Epoch 3/10
100/100 [==============================] - 43s 434ms/step - loss: 0.5696 - acc: 0.8353 - val_loss: 0.6315 - val_acc: 0.8042
Epoch 4/10
100/100 [==============================] - 43s 434ms/step - loss: 0.4558 - acc: 0.8750 - val_loss: 0.5895 - val_acc: 0.8116
Epoch 5/10
100/100 [==============================] - 44s 438ms/step - loss: 0.3999 - acc: 0.8879 - val_loss: 0.5613 - val_acc: 0.8253
Epoch 6/10
100/100 [==============================] - 44s 435ms/step - loss: 0.3411 - acc: 0.9097 - val_loss: 0.5267 - val_acc: 0.8358
Epoch 7/10
100/100 [==============================] - 44s 436ms/step - loss: 0.2994 - acc: 0.9197 - val_loss: 0.5162 - val_acc: 0.8368
Epoch 8/10
100/100 [==============================] - 44s 435ms/step - loss: 0.2730 - acc: 0.9271 - val_loss: 0.5024 - val_acc: 0.8347
Epoch 9/10
100/100 [==============================] - 44s 444ms/step - loss: 0.2481 - acc: 0.9326 - val_loss: 0.4981 - val_acc: 0.8253
Epoch 10/10
100/100 [==============================] - 44s 436ms/step - loss: 0.2211 - acc: 0.9492 - val_loss: 0.4767 - val_acc: 0.8432
```

Name	Submitted	Wait time	Execution time	Score
resnet50_batch_size_38_38.csv	just now	0 seconds	0 seconds	0.82808

Complete

Resnet 50 with train batch size = 50, validation batch size = 50, steps per epoch = 76, validation steps = 19, epochs = 10

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
resnet50 (Model)             (None, 2048)              23587712
_________________________________________________________________
dense (Dense)                (None, 12)                24588
=================================================================
Total params: 23,612,300
Trainable params: 24,588
Non-trainable params: 23,587,712
_________________________________________________________________
Found 3800 images belonging to 12 classes.
Found 950 images belonging to 12 classes.
Epoch 1/10
76/76 [==============================] - 47s 622ms/step - loss: 1.5985 - acc: 0.5016 - val_loss: 1.2558 - val_acc: 0.5726
Epoch 2/10
76/76 [==============================] - 43s 562ms/step - loss: 0.8073 - acc: 0.7682 - val_loss: 0.8108 - val_acc: 0.7389
Epoch 3/10
76/76 [==============================] - 43s 566ms/step - loss: 0.6030 - acc: 0.8318 - val_loss: 0.7046 - val_acc: 0.7842
Epoch 4/10
76/76 [==============================] - 43s 565ms/step - loss: 0.4923 - acc: 0.8584 - val_loss: 0.6054 - val_acc: 0.8242
Epoch 5/10
76/76 [==============================] - 44s 574ms/step - loss: 0.4124 - acc: 0.8913 - val_loss: 0.5827 - val_acc: 0.8126
Epoch 6/10
76/76 [==============================] - 43s 567ms/step - loss: 0.3544 - acc: 0.9079 - val_loss: 0.5352 - val_acc: 0.8263
Epoch 7/10
76/76 [==============================] - 43s 564ms/step - loss: 0.3163 - acc: 0.9142 - val_loss: 0.5339 - val_acc: 0.8168
Epoch 8/10
76/76 [==============================] - 43s 566ms/step - loss: 0.2787 - acc: 0.9387 - val_loss: 0.5086 - val_acc: 0.8389
Epoch 9/10
76/76 [==============================] - 43s 566ms/step - loss: 0.2586 - acc: 0.9355 - val_loss: 0.5262 - val_acc: 0.8326
Epoch 10/10
76/76 [==============================] - 43s 565ms/step - loss: 0.2361 - acc: 0.9426 - val_loss: 0.4966 - val_acc: 0.8368
```

Name	Submitted	Wait time	Execution time	Score
resnet50_batch_size_50_50.csv	a few seconds ago	0 seconds	0 seconds	0.83816

Complete

Resnet50 with default values: train batch size = 32, validation batch size = 32, steps per epoch = 200, validation steps = 1, epochs = 10
no preservation of the equation ,3800 = batch_size * steps_per_epch or 950= batch_size * validation_steps

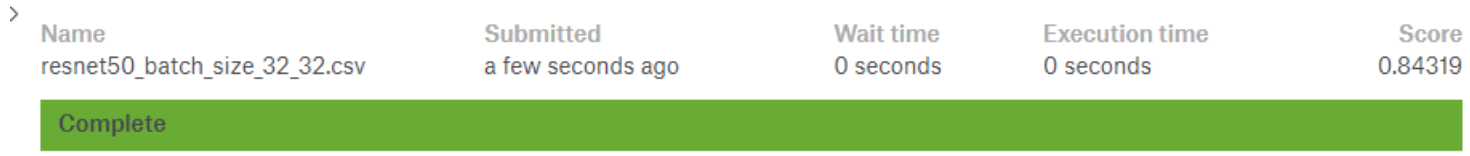
```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
resnet50 (Model)             (None, 2048)              23587712
_________________________________________________________________
dense (Dense)                (None, 12)                24588
=================================================================
Total params: 23,612,300
Trainable params: 24,588
Non-trainable params: 23,587,712
_________________________________________________________________
Found 3800 images belonging to 12 classes.
Found 950 images belonging to 12 classes.
Epoch 1/10
200/200 [==============================] - 62s 311ms/step - loss: 1.1795 - acc: 0.6271 - val_loss: 0.6481 - val_acc: 0.8125
Epoch 2/10
200/200 [==============================] - 58s 290ms/step - loss: 0.5633 - acc: 0.8307 - val_loss: 0.5099 - val_acc: 0.8750
Epoch 3/10
200/200 [==============================] - 58s 290ms/step - loss: 0.4012 - acc: 0.8842 - val_loss: 0.4551 - val_acc: 0.9062
Epoch 4/10
200/200 [==============================] - 58s 289ms/step - loss: 0.3178 - acc: 0.9109 - val_loss: 0.5137 - val_acc: 0.8438
Epoch 5/10
200/200 [==============================] - 58s 290ms/step - loss: 0.2620 - acc: 0.9327 - val_loss: 0.4367 - val_acc: 0.9062
Epoch 6/10
200/200 [==============================] - 58s 290ms/step - loss: 0.2253 - acc: 0.9398 - val_loss: 0.3758 - val_acc: 0.9062
Epoch 7/10
200/200 [==============================] - 58s 290ms/step - loss: 0.1912 - acc: 0.9527 - val_loss: 0.3227 - val_acc: 0.8750
Epoch 8/10
200/200 [==============================] - 58s 289ms/step - loss: 0.1755 - acc: 0.9548 - val_loss: 0.3385 - val_acc: 0.8750
Epoch 9/10
200/200 [==============================] - 58s 290ms/step - loss: 0.1478 - acc: 0.9652 - val_loss: 0.4712 - val_acc: 0.8750
Epoch 10/10
200/200 [==============================] - 58s 290ms/step - loss: 0.1295 - acc: 0.9721 - val_loss: 0.4202 - val_acc: 0.9062
```

resnet50_batch_size_32_32.csv	a few seconds ago	0 seconds	0 seconds	0.83816

Complete

Maybe we are overfitting, so we tried less number of epochs, train batch size = 32, validation batch size = 32,
steps per epoch = 200, validation steps = 1, **epochs = 6**
no preservation of the equation ,3800 = batch_size * steps_per_epch or
950= batch_size * validation_steps

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
resnet50 (Model)             (None, 2048)              23587712
_________________________________________________________________
dense (Dense)                (None, 12)                24588
=================================================================
Total params: 23,612,300
Trainable params: 24,588
Non-trainable params: 23,587,712
_________________________________________________________________
Found 3800 images belonging to 12 classes.
Found 950 images belonging to 12 classes.
Epoch 1/6
200/200 [==============================] - 62s 311ms/step - loss: 1.1819 - acc: 0.6341 - val_loss: 0.4328 - val_acc: 0.8750
Epoch 2/6
200/200 [==============================] - 58s 290ms/step - loss: 0.5605 - acc: 0.8340 - val_loss: 0.4034 - val_acc: 0.8438
Epoch 3/6
200/200 [==============================] - 58s 290ms/step - loss: 0.4015 - acc: 0.8881 - val_loss: 0.2664 - val_acc: 0.9688
Epoch 4/6
200/200 [==============================] - 58s 290ms/step - loss: 0.3165 - acc: 0.9158 - val_loss: 0.2224 - val_acc: 0.9375
Epoch 5/6
200/200 [==============================] - 58s 291ms/step - loss: 0.2612 - acc: 0.9304 - val_loss: 0.2110 - val_acc: 0.9688
Epoch 6/6
200/200 [==============================] - 58s 292ms/step - loss: 0.2212 - acc: 0.9465 - val_loss: 0.2437 - val_acc: 0.9062
```



Maybe we are overfitting, so we tried less number of epochs, train batch size = 32, validation batch size = 32,
steps per epoch = 200, validation steps = 1, **epochs = 3**
no preservation of the equation ,3800 = batch_size * steps_per_epch or
950= batch_size * validation_steps

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
resnet50 (Model)             (None, 2048)              23587712
_________________________________________________________________
dense (Dense)                (None, 12)                24588
=================================================================
Total params: 23,612,300
Trainable params: 24,588
Non-trainable params: 23,587,712
_________________________________________________________________
Found 3800 images belonging to 12 classes.
Found 950 images belonging to 12 classes.
Epoch 1/3
200/200 [==============================] - 62s 310ms/step - loss: 1.1755 - acc: 0.6396 - val_loss: 0.6261 - val_acc: 0.8125
Epoch 2/3
200/200 [==============================] - 58s 289ms/step - loss: 0.5560 - acc: 0.8391 - val_loss: 0.5356 - val_acc: 0.8750
Epoch 3/3
200/200 [==============================] - 58s 290ms/step - loss: 0.4043 - acc: 0.8859 - val_loss: 0.4948 - val_acc: 0.8438
```

Name	Submitted	Wait time	Execution time	Score
resnet50_batch_size_32_32.csv	a few seconds ago	0 seconds	0 seconds	0.79785

Complete

Best result is :Resnet 50 with train batch size = 25, validation batch size = 25, steps per epoch = 152, validation steps = 38, epochs = 10

- We will try with same parameters, with less epochs in case we are overfitting.
- With epochs = 8 we got:

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
resnet50 (Model)             (None, 2048)              23587712
_________________________________________________________________
dense (Dense)                (None, 12)                24588
=================================================================
Total params: 23,612,300
Trainable params: 24,588
Non-trainable params: 23,587,712
_________________________________________________________________
Found 3800 images belonging to 12 classes.
Found 950 images belonging to 12 classes.
Epoch 1/8
152/152 [==============================] - 47s 309ms/step - loss: 1.4299 - acc: 0.5387 - val_loss: 0.9366 - val_acc: 0.6926
Epoch 2/8
152/152 [==============================] - 43s 281ms/step - loss: 0.7294 - acc: 0.7811 - val_loss: 0.7246 - val_acc: 0.7811
Epoch 3/8
152/152 [==============================] - 43s 282ms/step - loss: 0.5434 - acc: 0.8411 - val_loss: 0.6158 - val_acc: 0.8032
Epoch 4/8
152/152 [==============================] - 43s 286ms/step - loss: 0.4428 - acc: 0.8718 - val_loss: 0.5630 - val_acc: 0.8221
Epoch 5/8
152/152 [==============================] - 43s 285ms/step - loss: 0.3871 - acc: 0.8918 - val_loss: 0.5712 - val_acc: 0.8084
Epoch 6/8
152/152 [==============================] - 43s 284ms/step - loss: 0.3414 - acc: 0.9018 - val_loss: 0.4936 - val_acc: 0.8295
Epoch 7/8
152/152 [==============================] - 43s 285ms/step - loss: 0.2880 - acc: 0.9234 - val_loss: 0.5021 - val_acc: 0.8432
Epoch 8/8
152/152 [==============================] - 43s 284ms/step - loss: 0.2729 - acc: 0.9187 - val_loss: 0.4859 - val_acc: 0.8389
```

Name	Submitted	Wait time	Execution time	Score
resnet50_batch_size_25_25.csv	a few seconds ago	0 seconds	0 seconds	0.80415

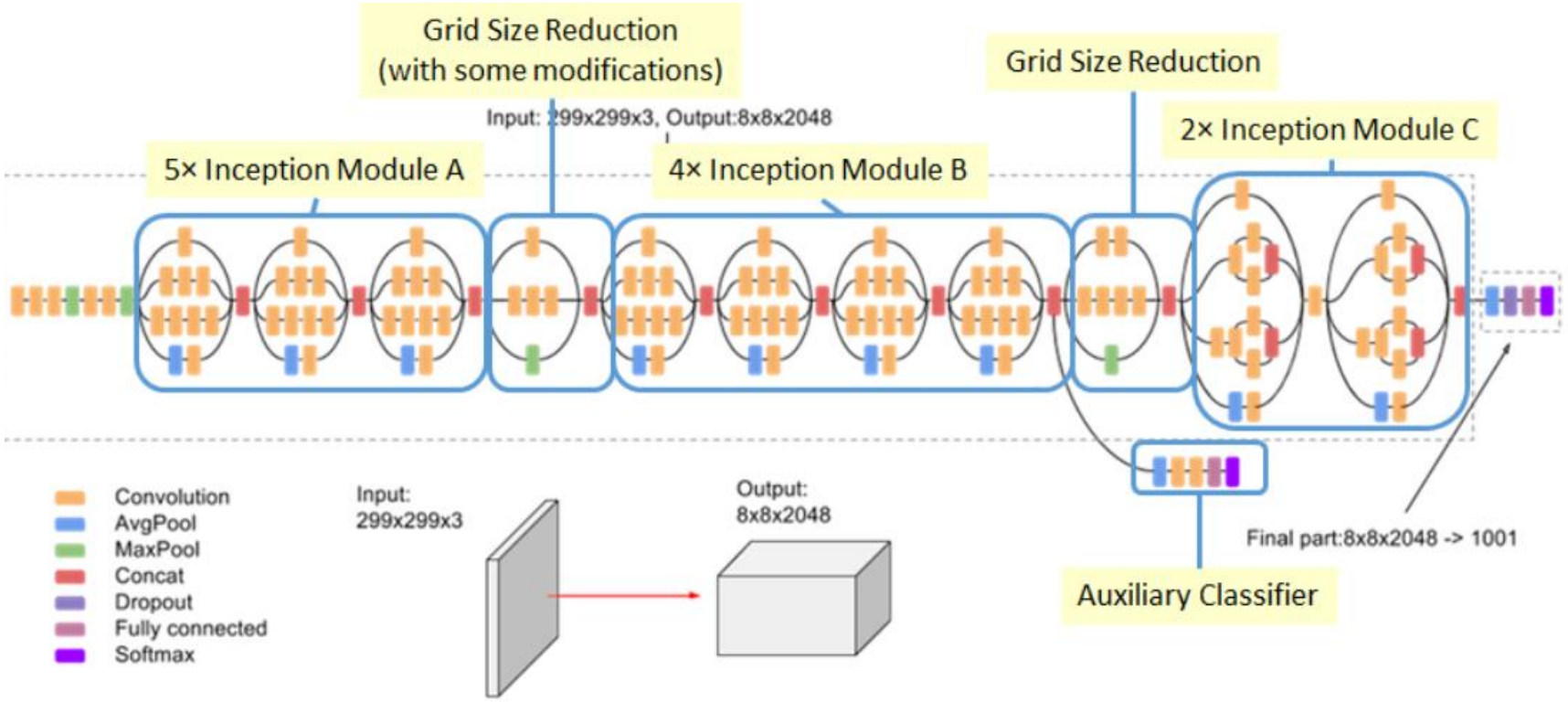
Complete

Still the best result is Resnet 50 with train batch size = 25, validation batch size = 25,
steps per epoch = 152, validation steps = 38, epochs = 10

- From now on we use these parameters and try on different pre-trained models on ImageNet.

InceptionV3

The default input size for this model is 299x299.



InceptionV3with train batch size = 25, validation batch size = 25, steps per epoch = 152, validation steps = 38, epochs = 10

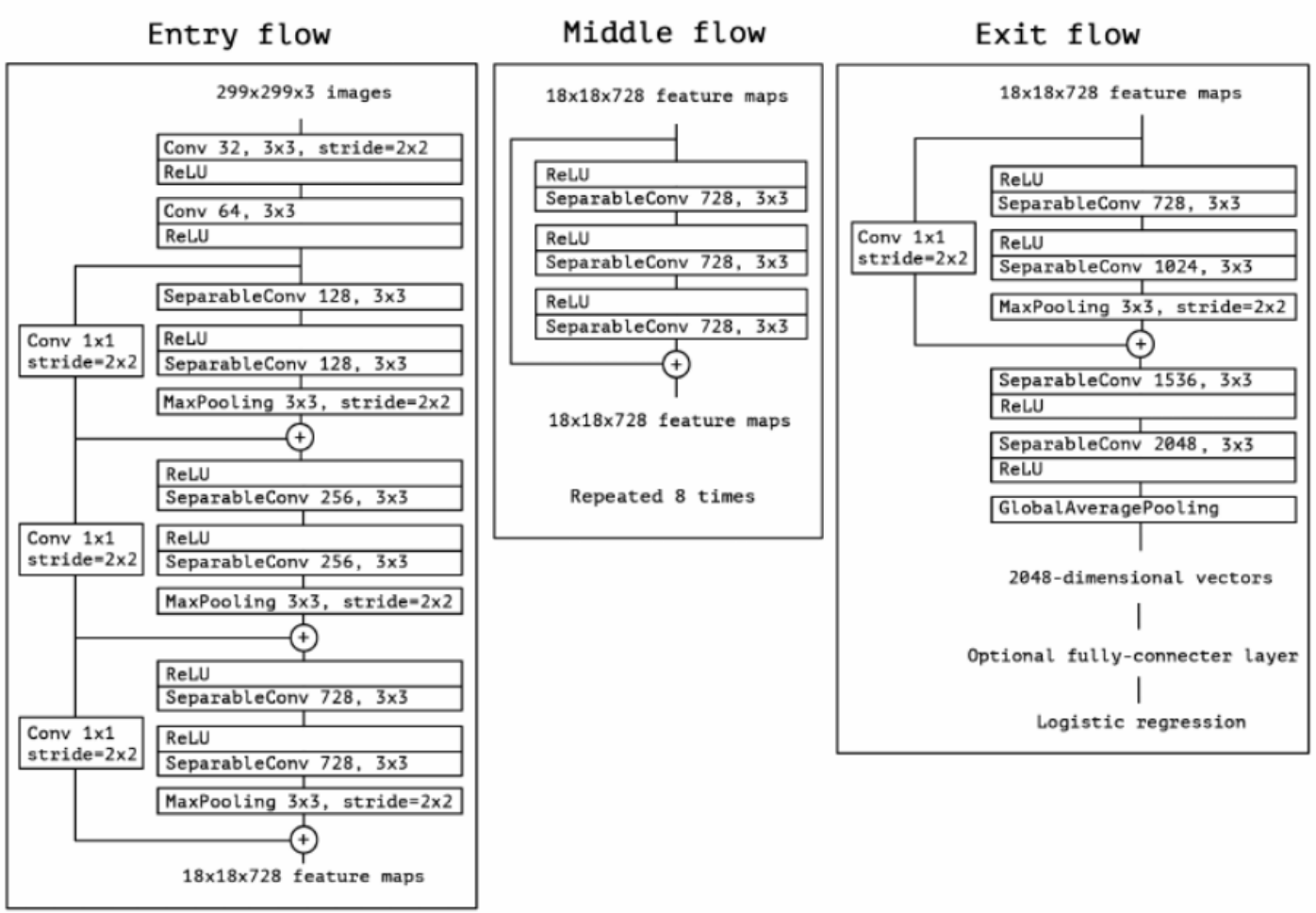
```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
inception_v3 (Model)         (None, 2048)              21802784
_________________________________________________________________
dense (Dense)                (None, 12)                24588
=================================================================
Total params: 21,827,372
Trainable params: 24,588
Non-trainable params: 21,802,784
_________________________________________________________________
Found 3800 images belonging to 12 classes.
Found 950 images belonging to 12 classes.
Epoch 1/10
152/152 [==============================] - 61s 398ms/step - loss: 1.7875 - acc: 0.4376 - val_loss: 1.3303 - val_acc: 0.6116
Epoch 2/10
152/152 [==============================] - 55s 359ms/step - loss: 1.1531 - acc: 0.6653 - val_loss: 1.0327 - val_acc: 0.7011
Epoch 3/10
152/152 [==============================] - 55s 361ms/step - loss: 0.9396 - acc: 0.7229 - val_loss: 0.9607 - val_acc: 0.7063
Epoch 4/10
152/152 [==============================] - 55s 361ms/step - loss: 0.8202 - acc: 0.7574 - val_loss: 0.8645 - val_acc: 0.7474
Epoch 5/10
152/152 [==============================] - 55s 360ms/step - loss: 0.7351 - acc: 0.7837 - val_loss: 0.7977 - val_acc: 0.7484
Epoch 6/10
152/152 [==============================] - 55s 363ms/step - loss: 0.6789 - acc: 0.7987 - val_loss: 0.7366 - val_acc: 0.7768
Epoch 7/10
152/152 [==============================] - 55s 363ms/step - loss: 0.6473 - acc: 0.8000 - val_loss: 0.7456 - val_acc: 0.7558
Epoch 8/10
152/152 [==============================] - 55s 362ms/step - loss: 0.5920 - acc: 0.8174 - val_loss: 0.6874 - val_acc: 0.7874
Epoch 9/10
152/152 [==============================] - 55s 362ms/step - loss: 0.5622 - acc: 0.8287 - val_loss: 0.6814 - val_acc: 0.7916
Epoch 10/10
152/152 [==============================] - 55s 362ms/step - loss: 0.5512 - acc: 0.8318 - val_loss: 0.6853 - val_acc: 0.7779
```

Name	Submitted	Wait time	Execution time	Score
inception3_batch_size_25_25.csv	just now	0 seconds	0 seconds	0.77644

Complete

Xception

The default input size for this model is 299x299



Xception with train batch size = 25, validation batch size = 25, steps per epoch = 152, validation steps = 38, epochs = 10

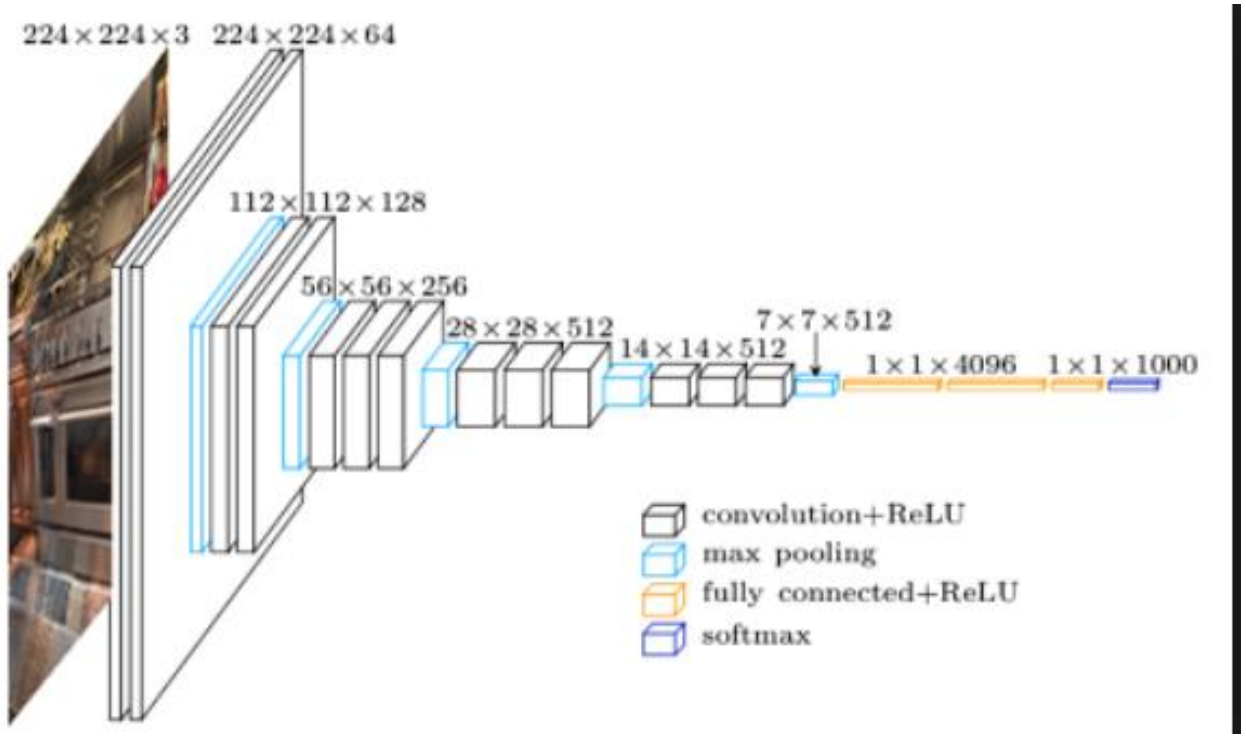
```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
xception (Model)             (None, 2048)              20861480
_________________________________________________________________
dense (Dense)                (None, 12)                24588
=================================================================
Total params: 20,886,068
Trainable params: 24,588
Non-trainable params: 20,861,480
_________________________________________________________________
Found 3800 images belonging to 12 classes.
Found 950 images belonging to 12 classes.
Epoch 1/10
2018-10-21 20:34:06.137908: W T:\src\github\tensorflow\tensorflow\core\common_runtime\bfc_allocator.cc:219] Allocator (GPU_0_b
rying to allocate 3.44GiB. The caller indicates that this is not a failure, but may mean that there could be performance gains
ailable.
152/152 [==============================] - 100s 660ms/step - loss: 1.5907 - acc: 0.5374 - val_loss: 1.1869 - val_acc: 0.6389
Epoch 2/10
152/152 [==============================] - 96s 634ms/step - loss: 0.9176 - acc: 0.7613 - val_loss: 0.8264 - val_acc: 0.7642
Epoch 3/10
152/152 [==============================] - 97s 635ms/step - loss: 0.7216 - acc: 0.7987 - val_loss: 0.7134 - val_acc: 0.8074
Epoch 4/10
152/152 [==============================] - 96s 635ms/step - loss: 0.6148 - acc: 0.8311 - val_loss: 0.6363 - val_acc: 0.8179
Epoch 5/10
152/152 [==============================] - 97s 638ms/step - loss: 0.5374 - acc: 0.8561 - val_loss: 0.5984 - val_acc: 0.8232
Epoch 6/10
152/152 [==============================] - 97s 635ms/step - loss: 0.5067 - acc: 0.8642 - val_loss: 0.5807 - val_acc: 0.8126
Epoch 7/10
152/152 [==============================] - 96s 634ms/step - loss: 0.4460 - acc: 0.8732 - val_loss: 0.5479 - val_acc: 0.8368
Epoch 8/10
152/152 [==============================] - 97s 635ms/step - loss: 0.4232 - acc: 0.8774 - val_loss: 0.5402 - val_acc: 0.8411
Epoch 9/10
152/152 [==============================] - 97s 636ms/step - loss: 0.3991 - acc: 0.8895 - val_loss: 0.5129 - val_acc: 0.8463
Epoch 10/10
152/152 [==============================] - 97s 637ms/step - loss: 0.3785 - acc: 0.8905 - val_loss: 0.5031 - val_acc: 0.8505
```

Name	Submitted	Wait time	Execution time	Score
xception_batch_size_25_25.csv	a few seconds ago	0 seconds	0 seconds	0.84949

Complete

VGG16

The default input size for this model is 224x224



VGG16 with train batch size = 25, validation batch size = 25, steps per epoch = 152, validation steps = 38, epochs = 10

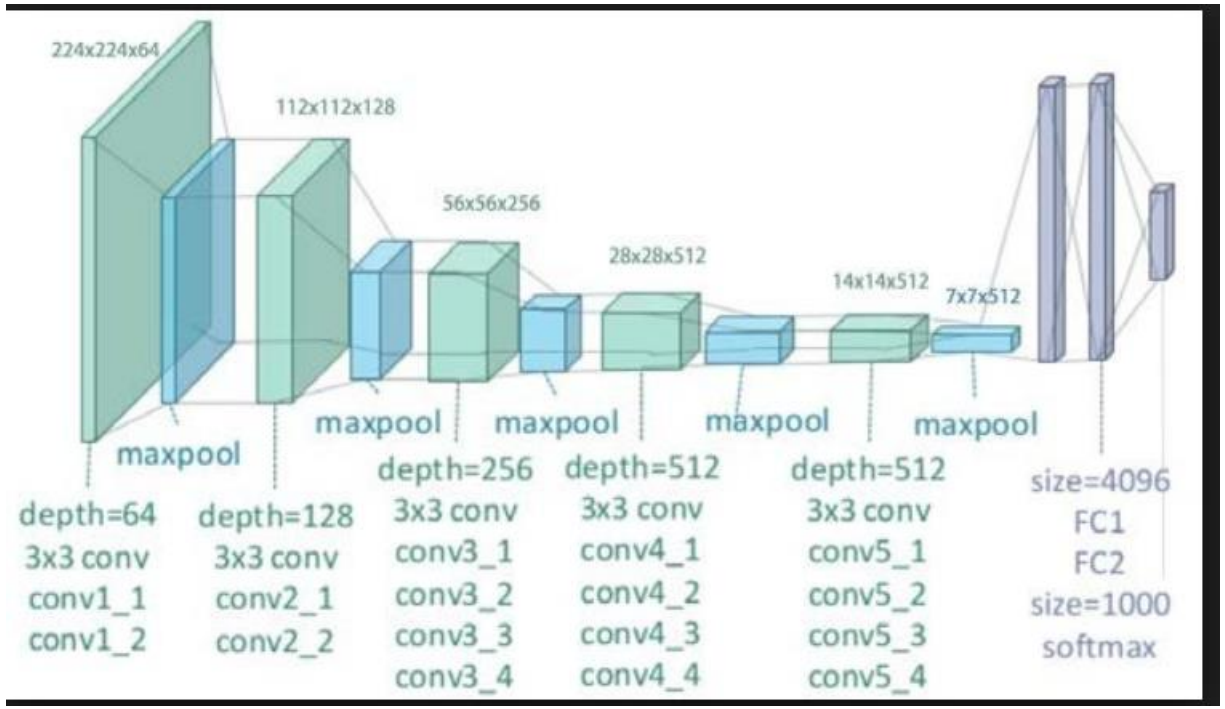
```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
vgg16 (Model)                (None, 512)               14714688
_________________________________________________________________
dense (Dense)                (None, 12)                6156
=================================================================
Total params: 14,720,844
Trainable params: 6,156
Non-trainable params: 14,714,688
_________________________________________________________________
Found 3800 images belonging to 12 classes.
Found 950 images belonging to 12 classes.
Epoch 1/10
2018-10-21 20:54:20.857680: W T:\src\github\tensorflow\tensorflow\core\common_runtime\bfc_allocator.cc:219] Allocator (GPU_0
rying to allocate 2.69GiB. The caller indicates that this is not a failure, but may mean that there could be performance gai
ailable.
2018-10-21 20:54:20.865160: W T:\src\github\tensorflow\tensorflow\core\common_runtime\bfc_allocator.cc:219] Allocator (GPU_0
rying to allocate 2.80GiB. The caller indicates that this is not a failure, but may mean that there could be performance gai
ailable.
2018-10-21 20:54:21.565041: W T:\src\github\tensorflow\tensorflow\core\common_runtime\bfc_allocator.cc:219] Allocator (GPU_0
rying to allocate 2.43GiB. The caller indicates that this is not a failure, but may mean that there could be performance gai
ailable.
152/152 [==============================] - 62s 410ms/step - loss: 2.5199 - acc: 0.3316 - val_loss: 1.5911 - val_acc: 0.4905
Epoch 2/10
152/152 [==============================] - 59s 390ms/step - loss: 1.1917 - acc: 0.6037 - val_loss: 1.1050 - val_acc: 0.6253
Epoch 3/10
152/152 [==============================] - 60s 392ms/step - loss: 0.8591 - acc: 0.7176 - val_loss: 0.9238 - val_acc: 0.7084
Epoch 4/10
152/152 [==============================] - 59s 391ms/step - loss: 0.6880 - acc: 0.7747 - val_loss: 0.8112 - val_acc: 0.7284
Epoch 5/10
152/152 [==============================] - 59s 390ms/step - loss: 0.5859 - acc: 0.8084 - val_loss: 0.7512 - val_acc: 0.7537
Epoch 6/10
152/152 [==============================] - 59s 390ms/step - loss: 0.5047 - acc: 0.8384 - val_loss: 0.7417 - val_acc: 0.7684
Epoch 7/10
152/152 [==============================] - 59s 389ms/step - loss: 0.4541 - acc: 0.8608 - val_loss: 0.6932 - val_acc: 0.7832
Epoch 8/10
152/152 [==============================] - 59s 389ms/step - loss: 0.4031 - acc: 0.8795 - val_loss: 0.6665 - val_acc: 0.7989
Epoch 9/10
152/152 [==============================] - 59s 389ms/step - loss: 0.3715 - acc: 0.8876 - val_loss: 0.6668 - val_acc: 0.7926
Epoch 10/10
152/152 [==============================] - 59s 390ms/step - loss: 0.3353 - acc: 0.9003 - val_loss: 0.6723 - val_acc: 0.7758
```

Name	Submitted	Wait time	Execution time	Score
vgg16_batch_size_25_25.csv	a few seconds ago	0 seconds	0 seconds	0.79093

Complete

VGG19

The default input size for this model is 224x224



VGG19 with train batch size = 25, validation batch size = 25, steps per epoch = 152, validation steps = 38, epochs = 10

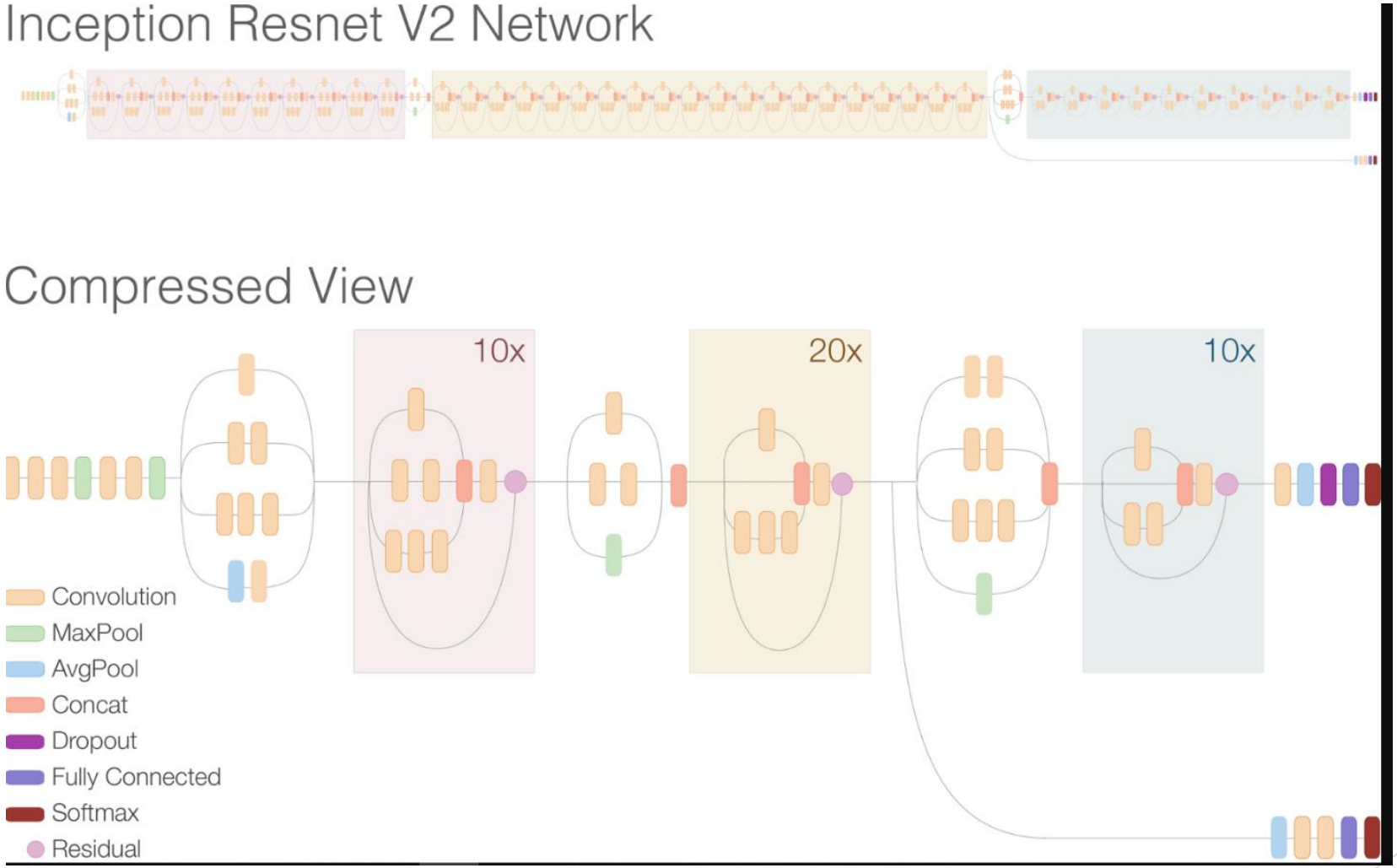
```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
vgg19 (Model)                (None, 512)               20024384
_________________________________________________________________
dense (Dense)                (None, 12)                6156
=================================================================
Total params: 20,030,540
Trainable params: 6,156
Non-trainable params: 20,024,384
_________________________________________________________________
Found 3800 images belonging to 12 classes.
Found 950 images belonging to 12 classes.
Epoch 1/10
2018-10-21 21:13:00.961641: W T:\src\github\tensorflow\tensorflow\core\common_runtime\bfc_allocator.cc:219] Allocator (GPU_0_b
n out of memory trying to allocate 2.69GiB. The caller indicates that this is not a failure, but may mean that there could be
mance gains if more memory were available.
2018-10-21 21:13:00.968227: W T:\src\github\tensorflow\tensorflow\core\common_runtime\bfc_allocator.cc:219] Allocator (GPU_0_b
n out of memory trying to allocate 2.80GiB. The caller indicates that this is not a failure, but may mean that there could be
mance gains if more memory were available.
2018-10-21 21:13:01.674254: W T:\src\github\tensorflow\tensorflow\core\common_runtime\bfc_allocator.cc:219] Allocator (GPU_0_b
n out of memory trying to allocate 2.43GiB. The caller indicates that this is not a failure, but may mean that there could be
mance gains if more memory were available.
152/152 [==============================] - 75s 496ms/step - loss: 2.1622 - acc: 0.3705 - val_loss: 1.5144 - val_acc: 0.5189
Epoch 2/10
152/152 [==============================] - 70s 462ms/step - loss: 1.0447 - acc: 0.6505 - val_loss: 1.0756 - val_acc: 0.6463
Epoch 3/10
152/152 [==============================] - 70s 464ms/step - loss: 0.7633 - acc: 0.7492 - val_loss: 0.8559 - val_acc: 0.7316
Epoch 4/10
152/152 [==============================] - 70s 463ms/step - loss: 0.6089 - acc: 0.8016 - val_loss: 0.7834 - val_acc: 0.7411
Epoch 5/10
152/152 [==============================] - 71s 464ms/step - loss: 0.5251 - acc: 0.8305 - val_loss: 0.6985 - val_acc: 0.7758
Epoch 6/10
152/152 [==============================] - 71s 464ms/step - loss: 0.4524 - acc: 0.8589 - val_loss: 0.6426 - val_acc: 0.7874
Epoch 7/10
152/152 [==============================] - 70s 463ms/step - loss: 0.4048 - acc: 0.8726 - val_loss: 0.6176 - val_acc: 0.7989
Epoch 8/10
152/152 [==============================] - 70s 463ms/step - loss: 0.3674 - acc: 0.8818 - val_loss: 0.6305 - val_acc: 0.7905
Epoch 9/10
152/152 [==============================] - 70s 463ms/step - loss: 0.3393 - acc: 0.8961 - val_loss: 0.5945 - val_acc: 0.8042
Epoch 10/10
152/152 [==============================] - 70s 463ms/step - loss: 0.3075 - acc: 0.9100 - val_loss: 0.6035 - val_acc: 0.8021
```

Name	Submitted	Wait time	Execution time	Score
vgg19_batch_size_25_25.csv	a few seconds ago	0 seconds	0 seconds	0.80478

Complete

InceptionResNetV2

The default input size for this model is 299x299



InceptionResNetV2 with train batch size = 25, validation batch size = 25, steps per epoch = 152, validation steps = 38, epochs = 10

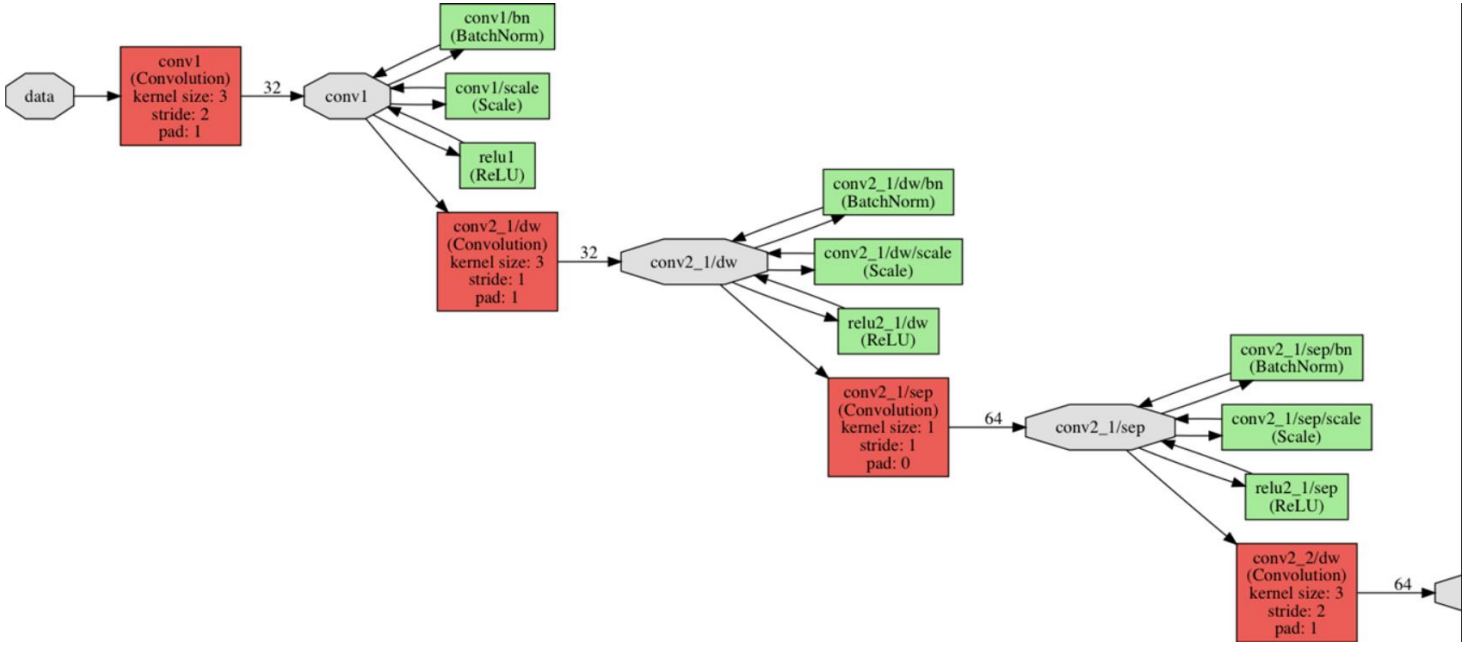
```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
inception_resnet_v2 (Model)  (None, 1536)              54336736
_________________________________________________________________
dense (Dense)                (None, 12)                18444
=================================================================
Total params: 54,355,180
Trainable params: 18,444
Non-trainable params: 54,336,736
_________________________________________________________________
Found 3800 images belonging to 12 classes.
Found 950 images belonging to 12 classes.
Epoch 1/10
2018-10-22 20:35:50.416014: W T:\src\github\tensorflow\tensorflow\core\common_runtime\bfc_allocator.cc:219] Allocator (GPU_0_b
trying to allocate 2.83GiB. The caller indicates that this is not a failure, but may mean that there could be performance gain
available.
2018-10-22 20:35:50.868917: W T:\src\github\tensorflow\tensorflow\core\common_runtime\bfc_allocator.cc:219] Allocator (GPU_0_b
trying to allocate 3.47GiB. The caller indicates that this is not a failure, but may mean that there could be performance gain
available.
152/152 [==============================] - 137s 899ms/step - loss: 1.7962 - acc: 0.4503 - val_loss: 1.3228 - val_acc: 0.5874
Epoch 2/10
152/152 [==============================] - 123s 810ms/step - loss: 1.1972 - acc: 0.6647 - val_loss: 1.0667 - val_acc: 0.6874
Epoch 3/10
152/152 [==============================] - 123s 811ms/step - loss: 0.9853 - acc: 0.7255 - val_loss: 0.9829 - val_acc: 0.7137
Epoch 4/10
152/152 [==============================] - 123s 810ms/step - loss: 0.8426 - acc: 0.7576 - val_loss: 0.9015 - val_acc: 0.7347
Epoch 5/10
152/152 [==============================] - 123s 811ms/step - loss: 0.7767 - acc: 0.7729 - val_loss: 0.8348 - val_acc: 0.7484
Epoch 6/10
152/152 [==============================] - 124s 813ms/step - loss: 0.7043 - acc: 0.7937 - val_loss: 0.7950 - val_acc: 0.7526
Epoch 7/10
152/152 [==============================] - 124s 815ms/step - loss: 0.6552 - acc: 0.8029 - val_loss: 0.7967 - val_acc: 0.7495
Epoch 8/10
152/152 [==============================] - 124s 813ms/step - loss: 0.6303 - acc: 0.8147 - val_loss: 0.7736 - val_acc: 0.7621
Epoch 9/10
152/152 [==============================] - 124s 813ms/step - loss: 0.6004 - acc: 0.8224 - val_loss: 0.7242 - val_acc: 0.7684
Epoch 10/10
152/152 [==============================] - 124s 813ms/step - loss: 0.5775 - acc: 0.8282 - val_loss: 0.7184 - val_acc: 0.7758
2018-10-22 20:56:31.345068: W T:\src\github\tensorflow\tensorflow\core\common_runtime\bfc_allocator.cc:219] Allocator (GPU_0_b
```

Name	Submitted	Wait time	Execution time	Score
inception_resnet_v2_batch_size_25_25....	a few seconds ago	1 seconds	0 seconds	0.80982

Complete

MobileNet

The default input size for this model is 224x224



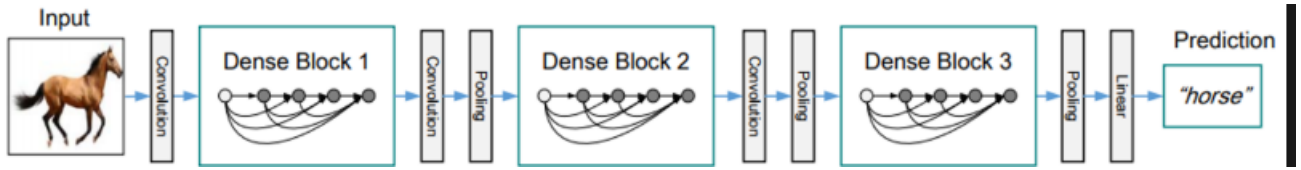
MobileNet with train batch size = 25, validation batch size = 25, steps per epoch = 152, validation steps = 38, epochs = 10

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
mobilenet_1.00_224 (Model)   (None, 1024)              3228864
_________________________________________________________________
dense (Dense)                (None, 12)                12300
=================================================================
Total params: 3,241,164
Trainable params: 12,300
Non-trainable params: 3,228,864
_________________________________________________________________
Found 3800 images belonging to 12 classes.
Found 950 images belonging to 12 classes.
Epoch 1/10
152/152 [==============================] - 41s 272ms/step - loss: 1.5349 - acc: 0.5071 - val_loss: 0.9568 - val_acc: 0.7032
Epoch 2/10
152/152 [==============================] - 39s 260ms/step - loss: 0.8111 - acc: 0.7495 - val_loss: 0.7325 - val_acc: 0.7789
Epoch 3/10
152/152 [==============================] - 39s 254ms/step - loss: 0.6340 - acc: 0.8003 - val_loss: 0.6381 - val_acc: 0.8021
Epoch 4/10
152/152 [==============================] - 40s 262ms/step - loss: 0.5322 - acc: 0.8374 - val_loss: 0.5934 - val_acc: 0.8053
Epoch 5/10
152/152 [==============================] - 40s 261ms/step - loss: 0.4722 - acc: 0.8447 - val_loss: 0.5717 - val_acc: 0.8189
Epoch 6/10
152/152 [==============================] - 39s 254ms/step - loss: 0.4312 - acc: 0.8687 - val_loss: 0.5984 - val_acc: 0.8021
Epoch 7/10
152/152 [==============================] - 39s 255ms/step - loss: 0.3833 - acc: 0.8818 - val_loss: 0.5409 - val_acc: 0.8147
Epoch 8/10
152/152 [==============================] - 38s 250ms/step - loss: 0.3535 - acc: 0.8937 - val_loss: 0.5317 - val_acc: 0.8263
Epoch 9/10
152/152 [==============================] - 38s 253ms/step - loss: 0.3292 - acc: 0.9003 - val_loss: 0.5147 - val_acc: 0.8326
Epoch 10/10
152/152 [==============================] - 39s 255ms/step - loss: 0.3238 - acc: 0.9016 - val_loss: 0.5136 - val_acc: 0.8305
```

Name	Submitted	Wait time	Execution time	Score
mobilenet_batch_size_25_25.csv	a few seconds ago	0 seconds	0 seconds	0.82241

Complete

DenseNet121
The default input size for this model is 224x224



DenseNet121 with train batch size = 25, validation batch size = 25, steps per epoch = 152, validation steps = 38, epochs = 10

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
densenet121 (Model)          (None, 1024)              7037504
_________________________________________________________________
dense (Dense)                (None, 12)                12300
=================================================================
Total params: 7,049,804
Trainable params: 12,300
Non-trainable params: 7,037,504
_________________________________________________________________
Found 3800 images belonging to 12 classes.
Found 950 images belonging to 12 classes.
Epoch 1/10
152/152 [==============================] - 49s 319ms/step - loss: 1.5745 - acc: 0.4982 - val_loss: 1.1061 - val_acc: 0.6463
Epoch 2/10
152/152 [==============================] - 41s 269ms/step - loss: 0.8792 - acc: 0.7229 - val_loss: 0.8040 - val_acc: 0.7400
Epoch 3/10
152/152 [==============================] - 39s 259ms/step - loss: 0.6948 - acc: 0.7800 - val_loss: 0.6915 - val_acc: 0.7737
Epoch 4/10
152/152 [==============================] - 40s 263ms/step - loss: 0.5882 - acc: 0.8171 - val_loss: 0.6469 - val_acc: 0.7811
Epoch 5/10
152/152 [==============================] - 40s 262ms/step - loss: 0.5331 - acc: 0.8416 - val_loss: 0.6149 - val_acc: 0.8000
Epoch 6/10
152/152 [==============================] - 39s 260ms/step - loss: 0.4811 - acc: 0.8529 - val_loss: 0.5743 - val_acc: 0.8032
Epoch 7/10
152/152 [==============================] - 39s 256ms/step - loss: 0.4491 - acc: 0.8611 - val_loss: 0.5609 - val_acc: 0.8137
Epoch 8/10
152/152 [==============================] - 39s 258ms/step - loss: 0.4101 - acc: 0.8724 - val_loss: 0.5429 - val_acc: 0.8189
Epoch 9/10
152/152 [==============================] - 39s 258ms/step - loss: 0.3925 - acc: 0.8826 - val_loss: 0.5353 - val_acc: 0.8221
Epoch 10/10
152/152 [==============================] - 39s 259ms/step - loss: 0.3773 - acc: 0.8845 - val_loss: 0.5332 - val_acc: 0.8168
```

Name	Submitted	Wait time	Execution time	Score
densenet121_batch_size_25_25.csv	a few seconds ago	0 seconds	0 seconds	0.80352

Complete

DenseNet169 with train batch size = 25, validation batch size = 25, steps per epoch = 152, validation steps = 38, epochs = 10

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
densenet169 (Model)          (None, 1664)              12642880
_________________________________________________________________
dense (Dense)                (None, 12)                19980
=================================================================
Total params: 12,662,860
Trainable params: 19,980
Non-trainable params: 12,642,880
_________________________________________________________________
Found 3800 images belonging to 12 classes.
Found 950 images belonging to 12 classes.
Epoch 1/10
152/152 [==============================] - 56s 372ms/step - loss: 1.4822 - acc: 0.5300 - val_loss: 0.9926 - val_acc: 0.6737
Epoch 2/10
152/152 [==============================] - 47s 310ms/step - loss: 0.7656 - acc: 0.7705 - val_loss: 0.7321 - val_acc: 0.7589
Epoch 3/10
152/152 [==============================] - 47s 308ms/step - loss: 0.6013 - acc: 0.8234 - val_loss: 0.6251 - val_acc: 0.8084
Epoch 4/10
152/152 [==============================] - 47s 308ms/step - loss: 0.5005 - acc: 0.8518 - val_loss: 0.6028 - val_acc: 0.7947
Epoch 5/10
152/152 [==============================] - 47s 307ms/step - loss: 0.4387 - acc: 0.8668 - val_loss: 0.5593 - val_acc: 0.8158
Epoch 6/10
152/152 [==============================] - 46s 305ms/step - loss: 0.3795 - acc: 0.8932 - val_loss: 0.5284 - val_acc: 0.8253
Epoch 7/10
152/152 [==============================] - 47s 307ms/step - loss: 0.3536 - acc: 0.8984 - val_loss: 0.5196 - val_acc: 0.8253
Epoch 8/10
152/152 [==============================] - 46s 302ms/step - loss: 0.3230 - acc: 0.9066 - val_loss: 0.5050 - val_acc: 0.8263
Epoch 9/10
152/152 [==============================] - 46s 306ms/step - loss: 0.3022 - acc: 0.9129 - val_loss: 0.5108 - val_acc: 0.8263
Epoch 10/10
152/152 [==============================] - 46s 304ms/step - loss: 0.2751 - acc: 0.9211 - val_loss: 0.5081 - val_acc: 0.8295
```

Name	Submitted	Wait time	Execution time	Score
densenet169_batch_size_25_25.csv	a few seconds ago	0 seconds	0 seconds	0.82682

Complete

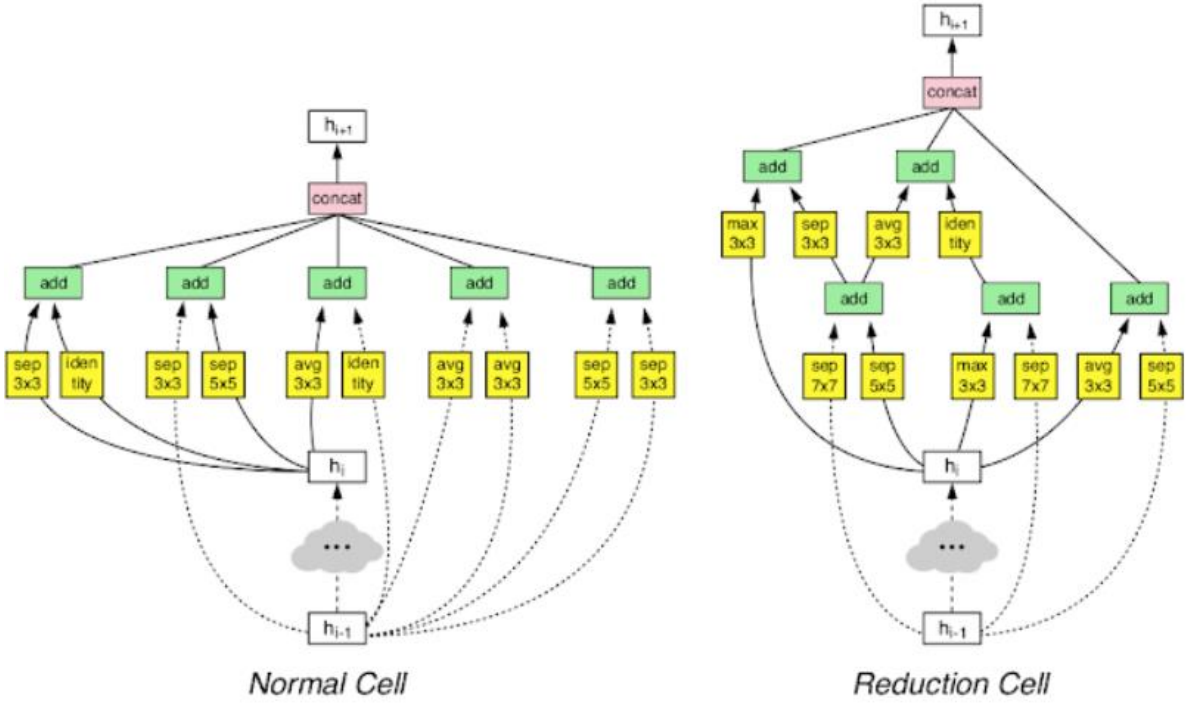
DenseNet201 with train batch size = 25, validation batch size = 25, steps per epoch = 152, validation steps = 38, epochs = 10

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
densenet201 (Model)          (None, 1920)              18321984
_________________________________________________________________
dense (Dense)                (None, 12)                23052
=================================================================
Total params: 18,345,036
Trainable params: 23,052
Non-trainable params: 18,321,984
_________________________________________________________________
Found 3800 images belonging to 12 classes.
Found 950 images belonging to 12 classes.
Epoch 1/10
2018-10-23 20:36:55.752963: W T:\src\github\tensorflow\tensorflow\core\common_runtime\bfc_allocator.cc:219] Allocator (GPU_0
e 3.55GiB. The caller indicates that this is not a failure, but may mean that there could be performance gains if more memor
152/152 [==============================] - 71s 468ms/step - loss: 1.4309 - acc: 0.5579 - val_loss: 0.8976 - val_acc: 0.7305
Epoch 2/10
152/152 [==============================] - 58s 384ms/step - loss: 0.7309 - acc: 0.7882 - val_loss: 0.6726 - val_acc: 0.7926
Epoch 3/10
152/152 [==============================] - 59s 386ms/step - loss: 0.5450 - acc: 0.8426 - val_loss: 0.6101 - val_acc: 0.8126
Epoch 4/10
152/152 [==============================] - 59s 385ms/step - loss: 0.4556 - acc: 0.8682 - val_loss: 0.5816 - val_acc: 0.8116
Epoch 5/10
152/152 [==============================] - 58s 385ms/step - loss: 0.4020 - acc: 0.8863 - val_loss: 0.5328 - val_acc: 0.8232
Epoch 6/10
152/152 [==============================] - 59s 387ms/step - loss: 0.3492 - acc: 0.9008 - val_loss: 0.5229 - val_acc: 0.8305
Epoch 7/10
152/152 [==============================] - 59s 387ms/step - loss: 0.3233 - acc: 0.9068 - val_loss: 0.5053 - val_acc: 0.8284
Epoch 8/10
152/152 [==============================] - 59s 385ms/step - loss: 0.2989 - acc: 0.9142 - val_loss: 0.4943 - val_acc: 0.8368
Epoch 9/10
152/152 [==============================] - 59s 389ms/step - loss: 0.2736 - acc: 0.9197 - val_loss: 0.4927 - val_acc: 0.8379
Epoch 10/10
152/152 [==============================] - 59s 385ms/step - loss: 0.2581 - acc: 0.9276 - val_loss: 0.4967 - val_acc: 0.8274
```

Name	Submitted	Wait time	Execution time	Score
densenet201_batch_size_25_25.csv	a few seconds ago	0 seconds	0 seconds	0.82745

Complete

NASNetMobil
The default input size for this model is 224x224



Our NASNet architecture is composed of two types of layers: Normal Layer (left), and Reduction Layer (right). These two layers are designed by AutoML.

NASNetMobil with train batch size = 25, validation batch size = 25, steps per epoch = 152, validation steps = 38, epochs = 10

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
NASNet (Model)               (None, 1056)              4269716
_________________________________________________________________
dense (Dense)                (None, 12)                12684
=================================================================
Total params: 4,282,400
Trainable params: 12,684
Non-trainable params: 4,269,716
_________________________________________________________________
Found 3800 images belonging to 12 classes.
Found 950 images belonging to 12 classes.
Epoch 1/10
152/152 [==============================] - 48s 317ms/step - loss: 1.9130 - acc: 0.3724 - val_loss: 2.2887 - val_acc: 0.1968
Epoch 2/10
152/152 [==============================] - 40s 262ms/step - loss: 1.3572 - acc: 0.5708 - val_loss: 2.1585 - val_acc: 0.2926
Epoch 3/10
152/152 [==============================] - 40s 262ms/step - loss: 1.1340 - acc: 0.6547 - val_loss: 2.3361 - val_acc: 0.2105
Epoch 4/10
152/152 [==============================] - 40s 262ms/step - loss: 1.0240 - acc: 0.6755 - val_loss: 2.3703 - val_acc: 0.2400
Epoch 5/10
152/152 [==============================] - 40s 262ms/step - loss: 0.9470 - acc: 0.7055 - val_loss: 2.2633 - val_acc: 0.2705
Epoch 6/10
152/152 [==============================] - 40s 265ms/step - loss: 0.8902 - acc: 0.7137 - val_loss: 2.3163 - val_acc: 0.2695
Epoch 7/10
152/152 [==============================] - 40s 262ms/step - loss: 0.8656 - acc: 0.7232 - val_loss: 2.1462 - val_acc: 0.3147
Epoch 8/10
152/152 [==============================] - 40s 260ms/step - loss: 0.8202 - acc: 0.7324 - val_loss: 2.0595 - val_acc: 0.3505
Epoch 9/10
152/152 [==============================] - 40s 263ms/step - loss: 0.7954 - acc: 0.7474 - val_loss: 2.0286 - val_acc: 0.3568
Epoch 10/10
152/152 [==============================] - 40s 261ms/step - loss: 0.7681 - acc: 0.7487 - val_loss: 1.9645 - val_acc: 0.3632
```

Name	Submitted	Wait time	Execution time	Score
NASNetMobile_batch_size_25_25.csv	a few seconds ago	0 seconds	0 seconds	0.41309

Complete

NASNetLarge with train batch size = 25, validation batch size = 25, steps per epoch = 152, validation steps = 38, epochs = 10

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
NASNet (Model)               (None, 4032)              84916818
_________________________________________________________________
dense (Dense)                (None, 12)                48396
=================================================================
Total params: 84,965,214
Trainable params: 48,396
Non-trainable params: 84,916,818
_________________________________________________________________
Found 3800 images belonging to 12 classes.
Found 950 images belonging to 12 classes.
Epoch 1/10
2018-10-23 21:17:52.205328: W T:\src\github\tensorflow\tensorflow\core\common_runtime\bfc_allocator.cc:219] Allocator (GPU_
e 3.17GiB. The caller indicates that this is not a failure, but may mean that there could be performance gains if more memo
2018-10-23 21:17:52.588349: W T:\src\github\tensorflow\tensorflow\core\common_runtime\bfc_allocator.cc:219] Allocator (GPU_
e 2.27GiB. The caller indicates that this is not a failure, but may mean that there could be performance gains if more memo
2018-10-23 21:17:54.158665: W T:\src\github\tensorflow\tensorflow\core\common_runtime\bfc_allocator.cc:219] Allocator (GPU_
e 3.93GiB. The caller indicates that this is not a failure, but may mean that there could be performance gains if more memo
152/152 [==============================] - 404s 3s/step - loss: 1.3727 - acc: 0.5805 - val_loss: 1.4879 - val_acc: 0.5011
Epoch 2/10
152/152 [==============================] - 390s 3s/step - loss: 0.8187 - acc: 0.7432 - val_loss: 1.4981 - val_acc: 0.4737
Epoch 3/10
152/152 [==============================] - 391s 3s/step - loss: 0.6745 - acc: 0.7866 - val_loss: 1.3998 - val_acc: 0.5105
Epoch 4/10
152/152 [==============================] - 392s 3s/step - loss: 0.6005 - acc: 0.8095 - val_loss: 1.3043 - val_acc: 0.5358
Epoch 5/10
152/152 [==============================] - 391s 3s/step - loss: 0.5493 - acc: 0.8147 - val_loss: 1.2556 - val_acc: 0.5705
Epoch 6/10
152/152 [==============================] - 390s 3s/step - loss: 0.5027 - acc: 0.8397 - val_loss: 1.2276 - val_acc: 0.5895
Epoch 7/10
152/152 [==============================] - 390s 3s/step - loss: 0.4574 - acc: 0.8503 - val_loss: 1.3404 - val_acc: 0.5589
Epoch 8/10
152/152 [==============================] - 390s 3s/step - loss: 0.4250 - acc: 0.8653 - val_loss: 1.1910 - val_acc: 0.5653
Epoch 9/10
152/152 [==============================] - 390s 3s/step - loss: 0.4138 - acc: 0.8668 - val_loss: 1.1490 - val_acc: 0.6021
Epoch 10/10
152/152 [==============================] - 390s 3s/step - loss: 0.3671 - acc: 0.8861 - val_loss: 1.1407 - val_acc: 0.5979
```

Name	Submitted	Wait time	Execution time	Score
NASNetLarge_batch_size_25_25.csv	a few seconds ago	0 seconds	0 seconds	0.67506

Complete

Still the best result on test data is Resnet 50 with train batch size = 25, validation batch size = 25,
steps per epoch = 152, validation steps = 38, epochs = 10

Name	Submitted	Wait time	Execution time	Score
resnet50_batch_size_25_25.csv	a few seconds ago	0 seconds	0 seconds	0.85138

Complete

Trying some data augmentation

Resnet 50 with train batch size = 25, validation batch size = 25,
steps per epoch = 152, validation steps = 38, epochs = 10
data augmentation :(horizontal_flip=True, width_shift_range = 0.2, height_shift_range = 0.2)

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
resnet50 (Model)             (None, 2048)              23587712
_________________________________________________________________
dense (Dense)                (None, 12)                24588
=================================================================
Total params: 23,612,300
Trainable params: 24,588
Non-trainable params: 23,587,712
_________________________________________________________________
Found 3800 images belonging to 12 classes.
Found 950 images belonging to 12 classes.
Epoch 1/10
152/152 [==============================] - 72s 473ms/step - loss: 1.4802 - acc: 0.5316 - val_loss: 0.9412 - val_acc: 0.6758
Epoch 2/10
152/152 [==============================] - 65s 429ms/step - loss: 0.8548 - acc: 0.7350 - val_loss: 0.7281 - val_acc: 0.7568
Epoch 3/10
152/152 [==============================] - 64s 424ms/step - loss: 0.6733 - acc: 0.7958 - val_loss: 0.6118 - val_acc: 0.8116
Epoch 4/10
152/152 [==============================] - 64s 418ms/step - loss: 0.5921 - acc: 0.8129 - val_loss: 0.5828 - val_acc: 0.8011
Epoch 5/10
152/152 [==============================] - 63s 418ms/step - loss: 0.5540 - acc: 0.8221 - val_loss: 0.5528 - val_acc: 0.8147
Epoch 6/10
152/152 [==============================] - 63s 416ms/step - loss: 0.4967 - acc: 0.8374 - val_loss: 0.5066 - val_acc: 0.8368
Epoch 7/10
152/152 [==============================] - 63s 414ms/step - loss: 0.4570 - acc: 0.8463 - val_loss: 0.5427 - val_acc: 0.8221
Epoch 8/10
152/152 [==============================] - 63s 413ms/step - loss: 0.4403 - acc: 0.8545 - val_loss: 0.5035 - val_acc: 0.8232
Epoch 9/10
152/152 [==============================] - 62s 409ms/step - loss: 0.4275 - acc: 0.8655 - val_loss: 0.5077 - val_acc: 0.8358
Epoch 10/10
152/152 [==============================] - 64s 420ms/step - loss: 0.3888 - acc: 0.8687 - val_loss: 0.4932 - val_acc: 0.8379
```

Name	Submitted	Wait time	Execution time	Score
resnet50_batch_size_25_25_data_aug....	a few seconds ago	1 seconds	0 seconds	0.85768

Complete

Resnet 50 with train batch size = 25, validation batch size = 25,
steps per epoch = 152, validation steps = 38, epochs = 10
data augmentation :(rotation_range=20, width_shift_range=.2, height_shift_range=.2, shear_range=0.2, zoom_range=0.2, channel_shift_range=1, horizontal_flip=True, vertical_flip=False)

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
resnet50 (Model)             (None, 2048)              23587712
_________________________________________________________________
dense (Dense)                (None, 12)                24588
=================================================================
Total params: 23,612,300
Trainable params: 24,588
Non-trainable params: 23,587,712
_________________________________________________________________
Found 3800 images belonging to 12 classes.
Found 950 images belonging to 12 classes.
Epoch 1/10
152/152 [==============================] - 74s 488ms/step - loss: 1.4841 - acc: 0.5239 - val_loss: 1.0172 - val_acc: 0.6242
Epoch 2/10
152/152 [==============================] - 68s 449ms/step - loss: 0.8906 - acc: 0.7234 - val_loss: 0.7904 - val_acc: 0.7358
Epoch 3/10
152/152 [==============================] - 66s 436ms/step - loss: 0.7117 - acc: 0.7682 - val_loss: 0.6415 - val_acc: 0.7863
Epoch 4/10
152/152 [==============================] - 65s 430ms/step - loss: 0.6203 - acc: 0.8068 - val_loss: 0.6924 - val_acc: 0.7558
Epoch 5/10
152/152 [==============================] - 65s 430ms/step - loss: 0.5752 - acc: 0.8145 - val_loss: 0.6161 - val_acc: 0.7863
Epoch 6/10
152/152 [==============================] - 65s 429ms/step - loss: 0.5174 - acc: 0.8353 - val_loss: 0.6106 - val_acc: 0.7884
Epoch 7/10
152/152 [==============================] - 65s 427ms/step - loss: 0.5102 - acc: 0.8308 - val_loss: 0.6532 - val_acc: 0.7758
Epoch 8/10
152/152 [==============================] - 65s 427ms/step - loss: 0.4597 - acc: 0.8447 - val_loss: 0.5327 - val_acc: 0.8232
Epoch 9/10
152/152 [==============================] - 67s 439ms/step - loss: 0.4451 - acc: 0.8487 - val_loss: 0.5398 - val_acc: 0.8221
Epoch 10/10
152/152 [==============================] - 67s 444ms/step - loss: 0.4204 - acc: 0.8613 - val_loss: 0.5899 - val_acc: 0.8042
```

Resnet 50 with train batch size = 25, validation batch size = 25,
steps per epoch = 152, validation steps = 38, epochs = 20
data augmentation :(rotation_range=20, width_shift_range=.2, height_shift_range=.2, shear_range=0.2, zoom_range=0.2, channel_shift_range=1, horizontal_flip=True, vertical_flip=False)

```
152/152 [==============================] - 64s 421ms/step - loss: 0.5346 - acc: 0.8174 - val_loss: 0.5575 - val_acc: 0.8147
Epoch 9/20
152/152 [==============================] - 64s 422ms/step - loss: 0.5005 - acc: 0.8347 - val_loss: 0.5922 - val_acc: 0.8147
Epoch 10/20
152/152 [==============================] - 64s 419ms/step - loss: 0.4444 - acc: 0.8521 - val_loss: 0.5175 - val_acc: 0.8284
Epoch 11/20
152/152 [==============================] - 65s 426ms/step - loss: 0.4441 - acc: 0.8505 - val_loss: 0.4881 - val_acc: 0.8484
Epoch 12/20
152/152 [==============================] - 64s 424ms/step - loss: 0.4406 - acc: 0.8450 - val_loss: 0.5780 - val_acc: 0.8084
Epoch 13/20
152/152 [==============================] - 64s 420ms/step - loss: 0.4404 - acc: 0.8526 - val_loss: 0.6117 - val_acc: 0.7926
Epoch 14/20
152/152 [==============================] - 65s 429ms/step - loss: 0.4149 - acc: 0.8542 - val_loss: 0.5708 - val_acc: 0.8053
Epoch 15/20
152/152 [==============================] - 64s 422ms/step - loss: 0.4036 - acc: 0.8629 - val_loss: 0.6127 - val_acc: 0.8074
Epoch 16/20
152/152 [==============================] - 64s 424ms/step - loss: 0.4027 - acc: 0.8603 - val_loss: 0.5159 - val_acc: 0.8263
Epoch 17/20
152/152 [==============================] - 64s 420ms/step - loss: 0.3909 - acc: 0.8634 - val_loss: 0.5532 - val_acc: 0.8074
Epoch 18/20
152/152 [==============================] - 64s 421ms/step - loss: 0.3796 - acc: 0.8708 - val_loss: 0.5194 - val_acc: 0.8200
Epoch 19/20
152/152 [==============================] - 64s 421ms/step - loss: 0.3759 - acc: 0.8711 - val_loss: 0.5785 - val_acc: 0.8211
Epoch 20/20
152/152 [==============================] - 64s 422ms/step - loss: 0.3664 - acc: 0.8737 - val_loss: 0.4599 - val_acc: 0.8400
```

Best result on test data:

- Resnet 50 with train batch size = 25, validation batch size = 25, steps per epoch = 152, validation steps = 38, epochs = 10
data augmentation :(horizontal_flip=True, width_shift_range = 0.2, height_shift_range = 0.2)

Summary of results in a table

Table for result of the pre-trained models on ImageNet with train batch size = 25, validation batch size = 25, steps per epoch = 152, validation steps = 38, epochs = 10.

Model	Validation accuracy	Test accuracy
Resnet50	82%	85.138%
InceptionV3	77%	77%
Xception	85%	84%
VGG16	77%	79%
VGG19	80%	80%
InceptionResNetV2	77%	80%
MobileNet	83%	82%
DenseNet121	81%	80%
DenseNet169	82%	82%
DenseNet201	82%	82%
NASNetMobil	36%	41%
NASNetLarge	59%	67%
Resnet50 with data augmentation	83%	85.768%

Sources:

- On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima
- Kaggle tutorial on Transfer Learning.