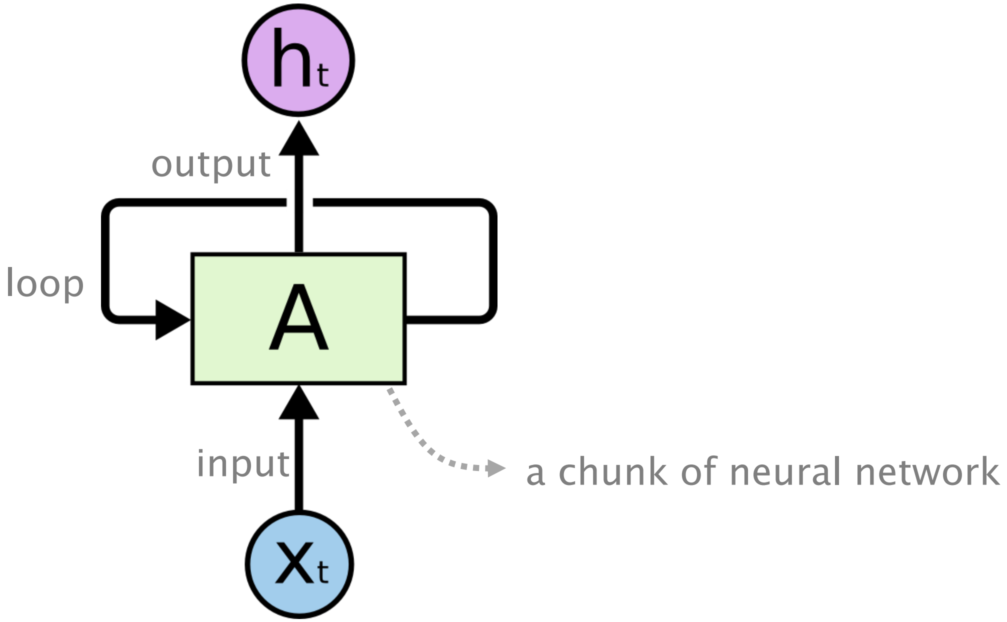
The Basic Concept of RNN and Google's Inception Model

2016.07.25

서지혜

RNN

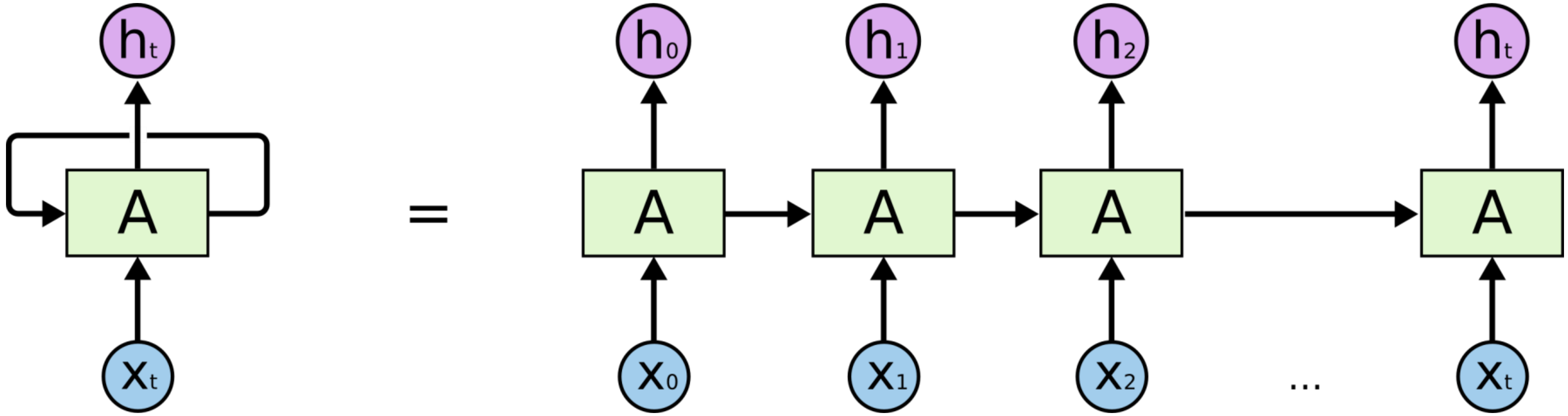
- Recurrent Neural Networks



Recurrent Neural Networks have loops.

Blog post by chrisolah http://colah.github.io/posts/2015-08-Understanding-LSTMs/

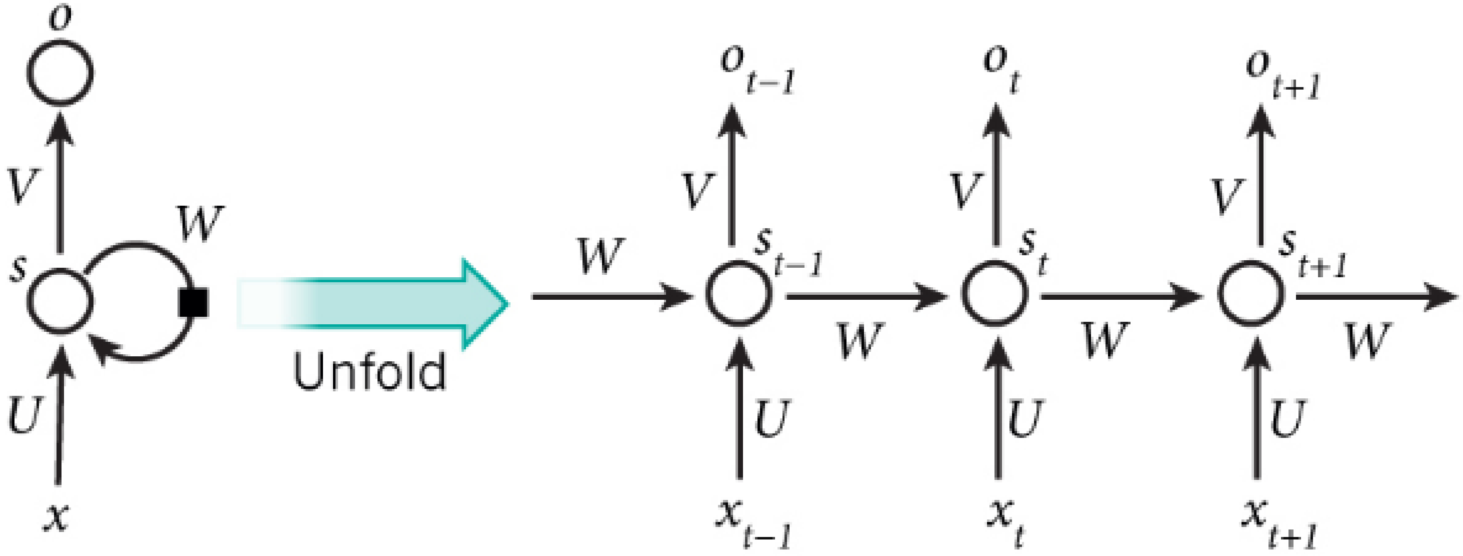
RNN



An unrolled recurrent neural network.

Blog post by chrisolah http://colah.github.io/posts/2015-08-Understanding-LSTMs/

RNN



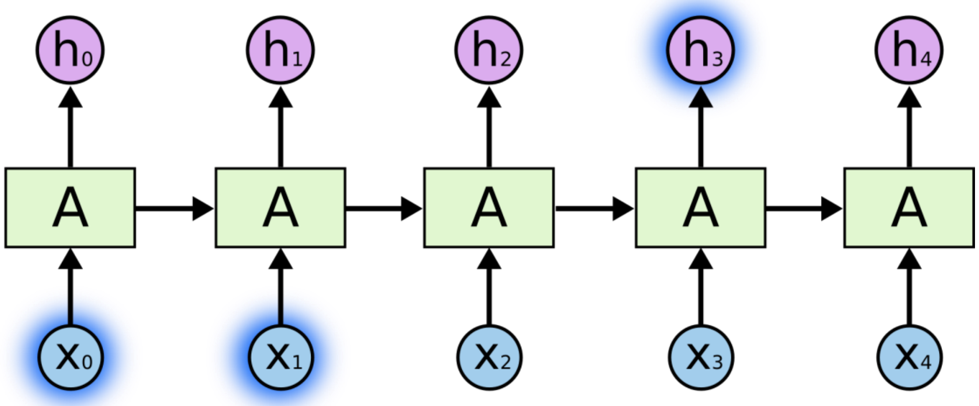
$$s_t = \tanh(Ux_t + Ws_{t-1})$$

$$o_t = softmax(Vs_t)$$

Blog post by Denny Britz http://www.wildml.com/2015/09/recurrent-neural-networks-tutorial-part-1-introduction-to-rnns/

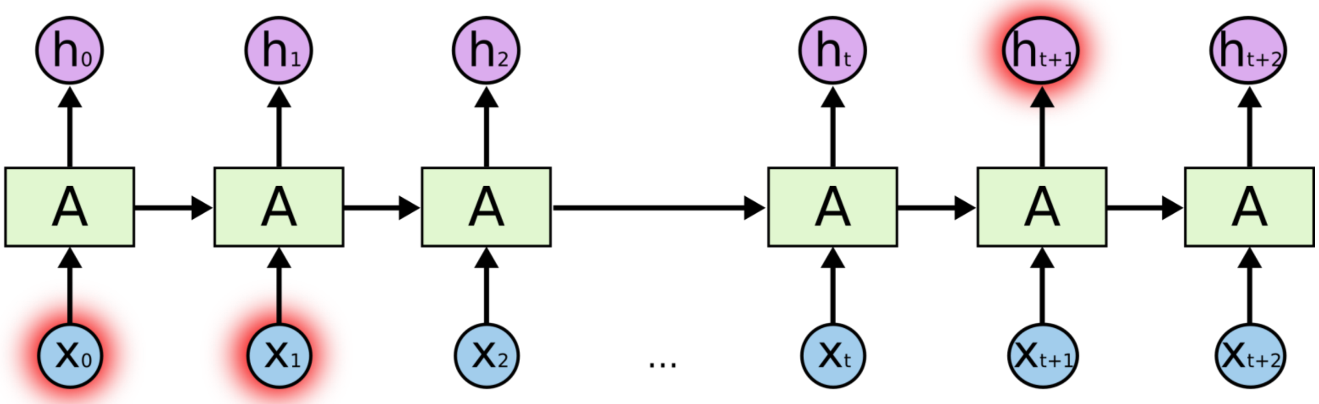
The Problem of Long-Term Dependencies

"the clouds are in the sky"



"I grew up in France…

I speak fluent French."



Blog post by chrisolah http://colah.github.io/posts/2015-08-Understanding-LSTMs/

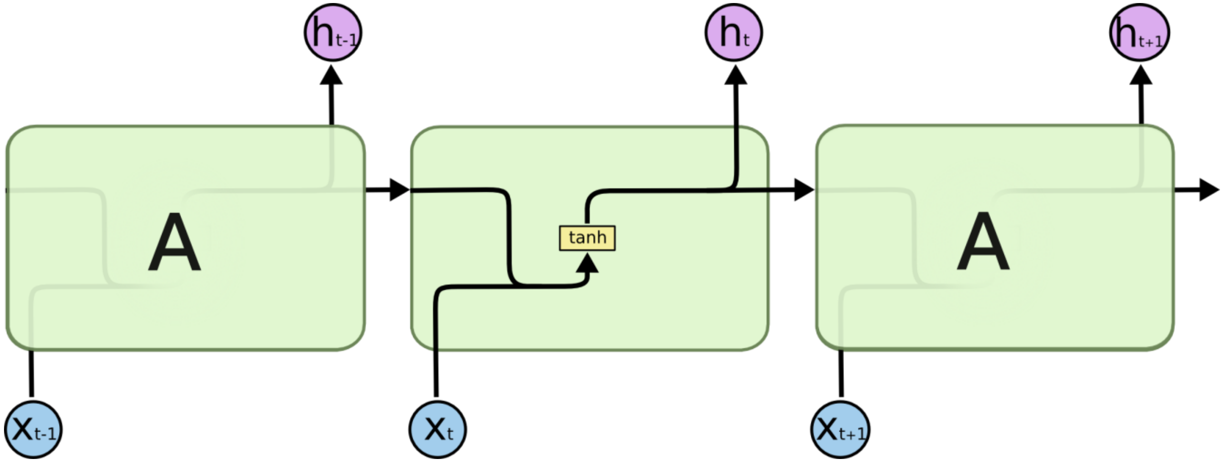
LSTM Networks

- Long Short Term Memory networks (LSTMs)
  - A special kind of RNN
  - Capable of learning long-term dependencies
  - Tremendously well on a large variety of problems
  - Now widely used

LSTM Networks

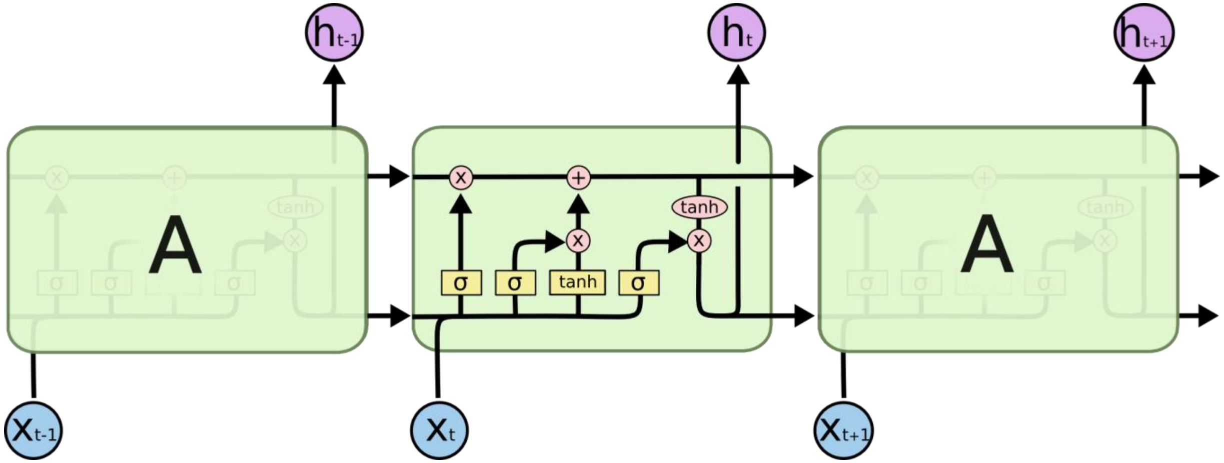
RNN

a single layer



LSTM

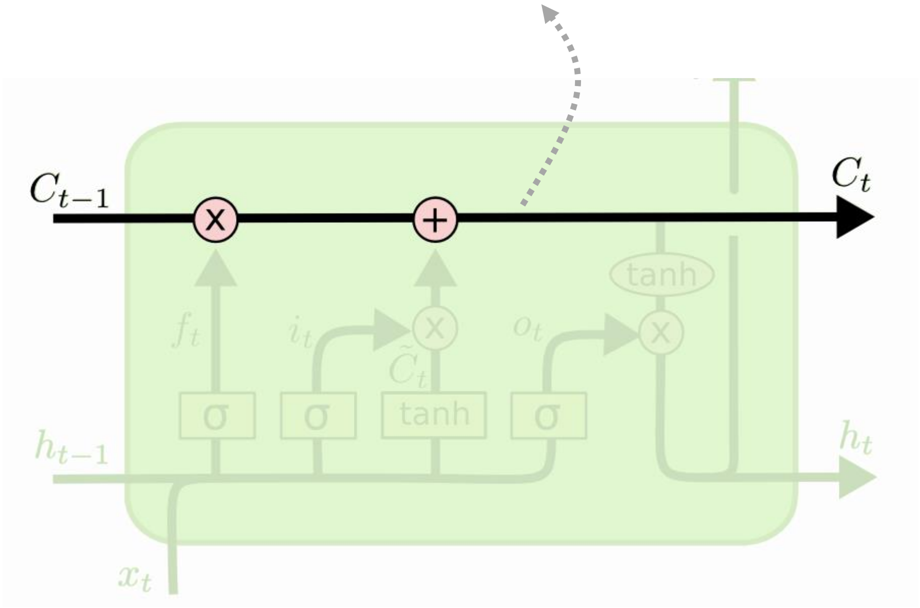
four interacting layers



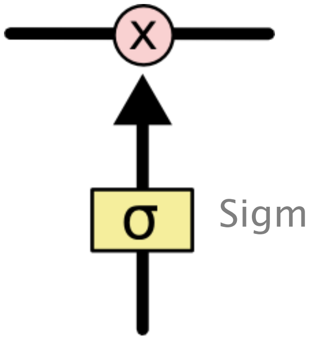
Blog post by chrisolah http://colah.github.io/posts/2015-08-Understanding-LSTMs/

The Core Idea Behind LSTMs

- The cell state



Gate



Pointwise Multiplication Operation

Sigmoid

remove or **add** information to the <u>cell state</u>, carefully regulated by structures called <u>gates</u>

Gates are a way to optionally let information through

Blog post by chrisolah http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Various types RNN

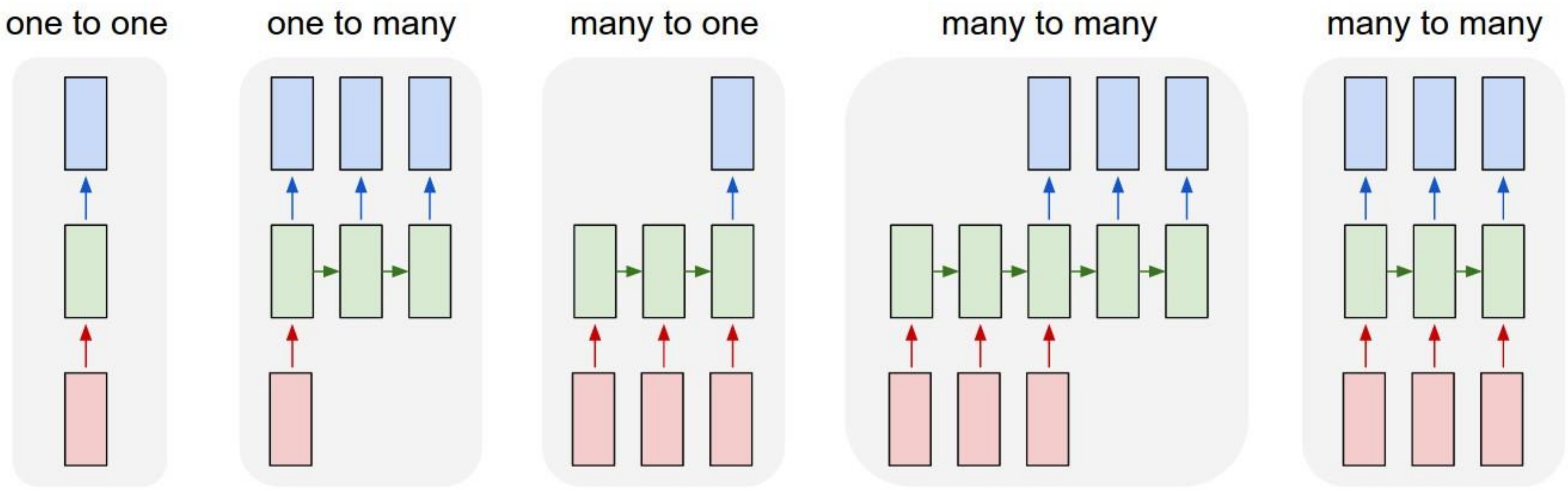
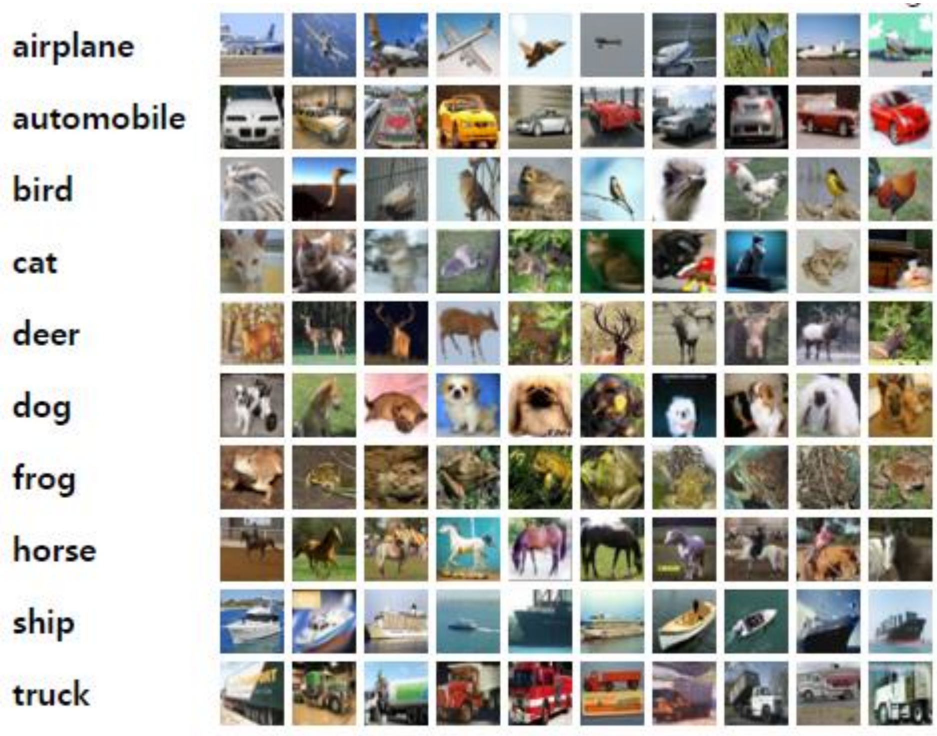


Image Classification Models

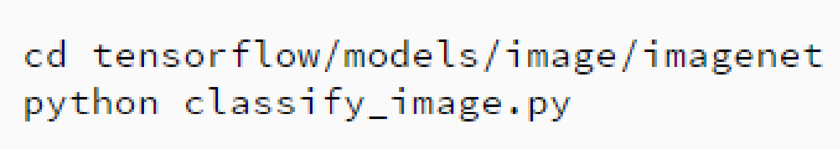
- TensorFlow Image Processing Tutorials
  - CIFAR-10



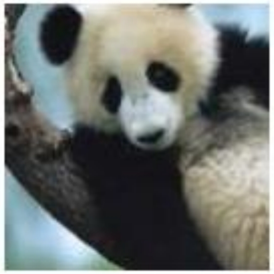
http://www.cs.toronto.edu/~kriz/cifar.h

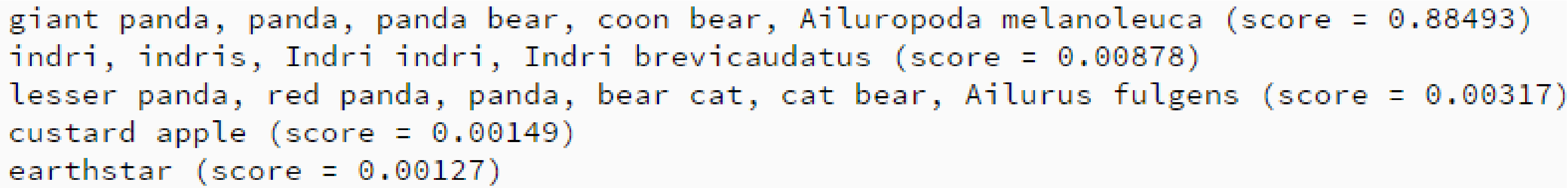
Image Classification Models

- TensorFlow Image Processing Tutorials
  - ImageNet



```
cd tensorflow/models/image/imagenet
python classify_image.py
```



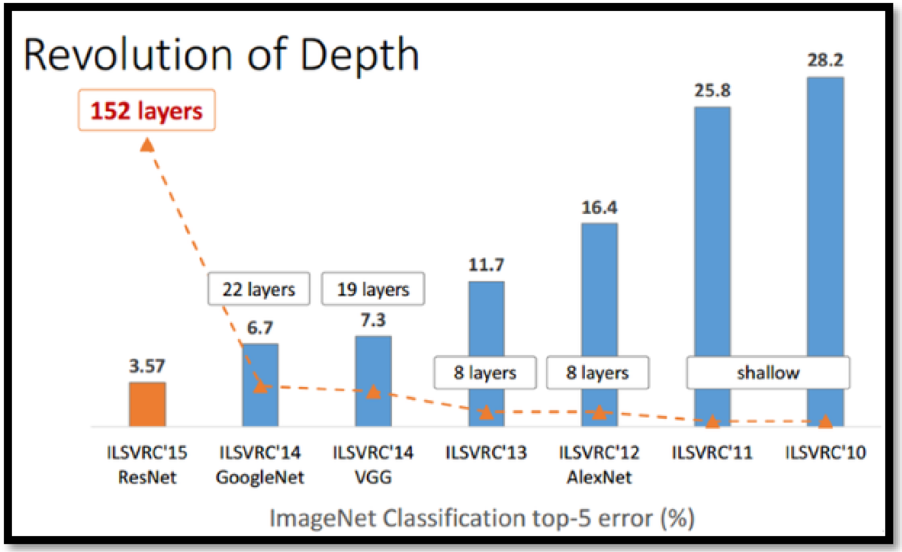


```
giant panda, panda, panda bear, coon bear, Ailuropoda melanoleuca (score = 0.88493)
indri, indris, Indri indri, Indri brevicaudatus (score = 0.00878)
lesser panda, red panda, panda, bear cat, cat bear, Ailurus fulgens (score = 0.00317)
custard apple (score = 0.00149)
earthstar (score = 0.00127)
```

https://www.tensorflow.org/versions/r0.9/tutorials/image_recognition/inde
tml

Network

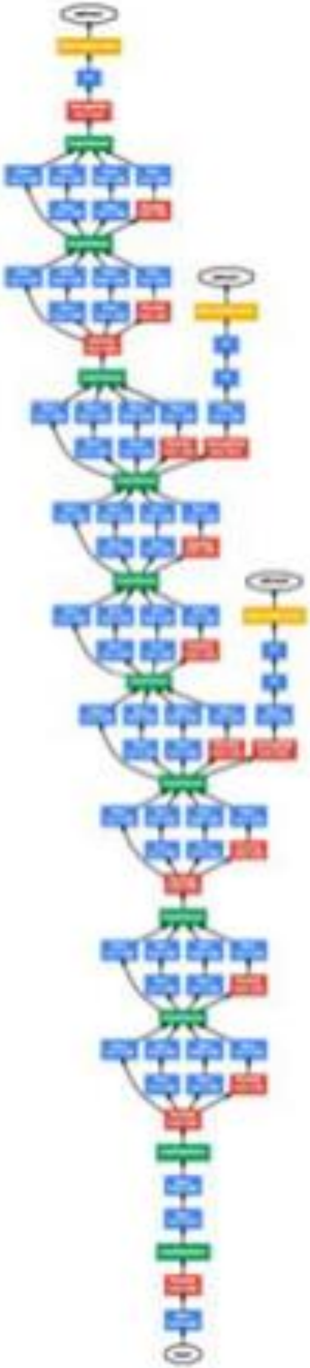
- GoogleLeNet
  - 1st at ILSVRC 2014
  - Inception module



- Problems
  - Large number of parameters
  - Dramatically increased use of computational resources
  - Vanishing gradient → overfitting, slow speed

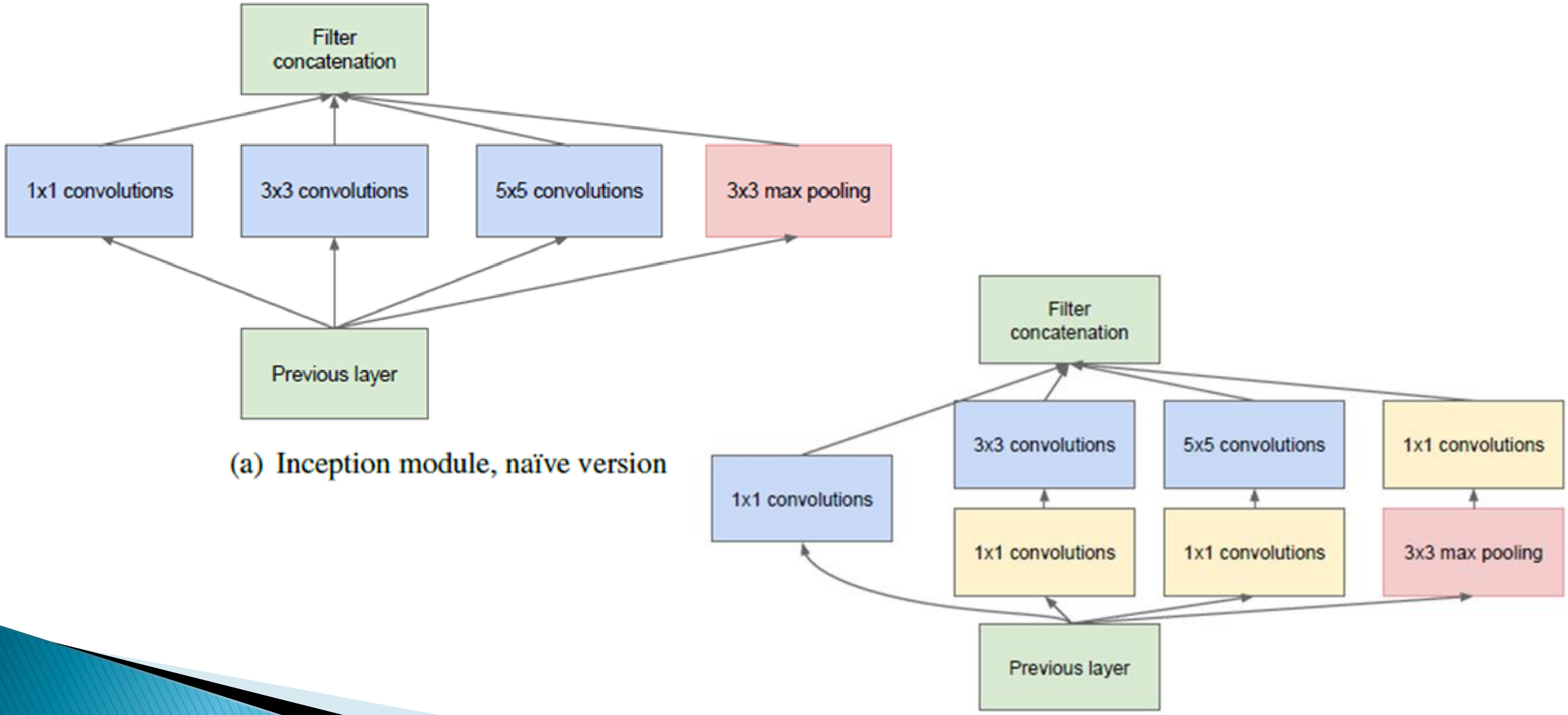
Google : Inception

C. Szegedy et al, "Going Deeper with Convolutions" (CVPR 2015)



Google : Inception

- Inception Module



(a) Inception module, naïve version

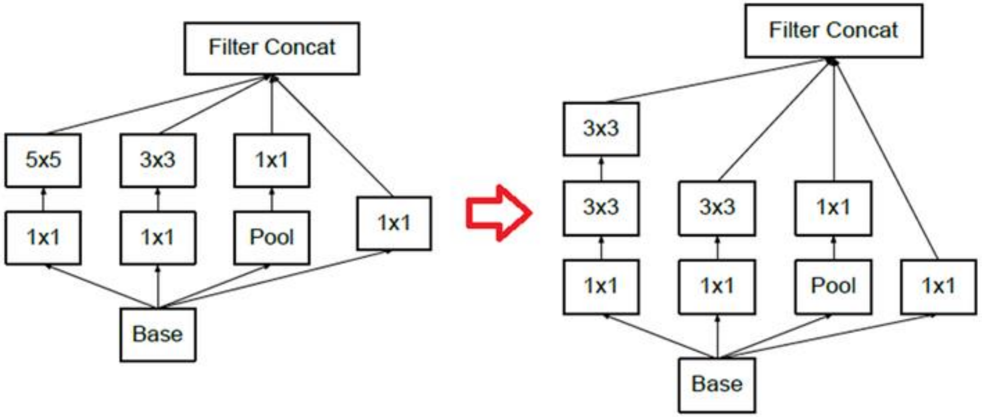
(b) Inception module with dimensionality reduction

Google : Inception

- Solution
  - 1x1 Convolutional Layer
  - Auxiliary Classifier

    Regularizer 와 같은 역할로 gradient 값을 증가시킨다.
  - Factorization into smaller convolutions

    큰 conv 커널 → 여러 개의 작은 conv 커널로

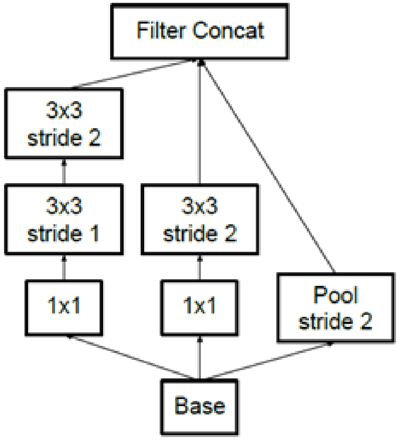


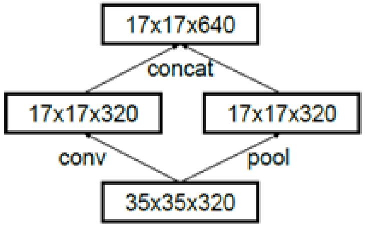
C. Szegedy et al. Rethinking the Inception Architecture for Computer Vision

Google : Inception

- Solution
  - Grid size reduction

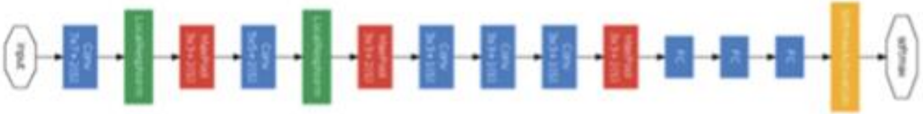
    Pooing & conv 나란히 적용



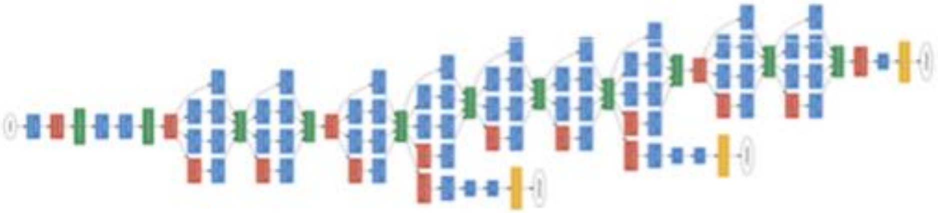


	params	FLOPs
Krizhevsky, Suskever and Hinton (2012)	60M	2B
Szegedy et al (2014)	5M	1.5B

AlexNet
(8 layers)

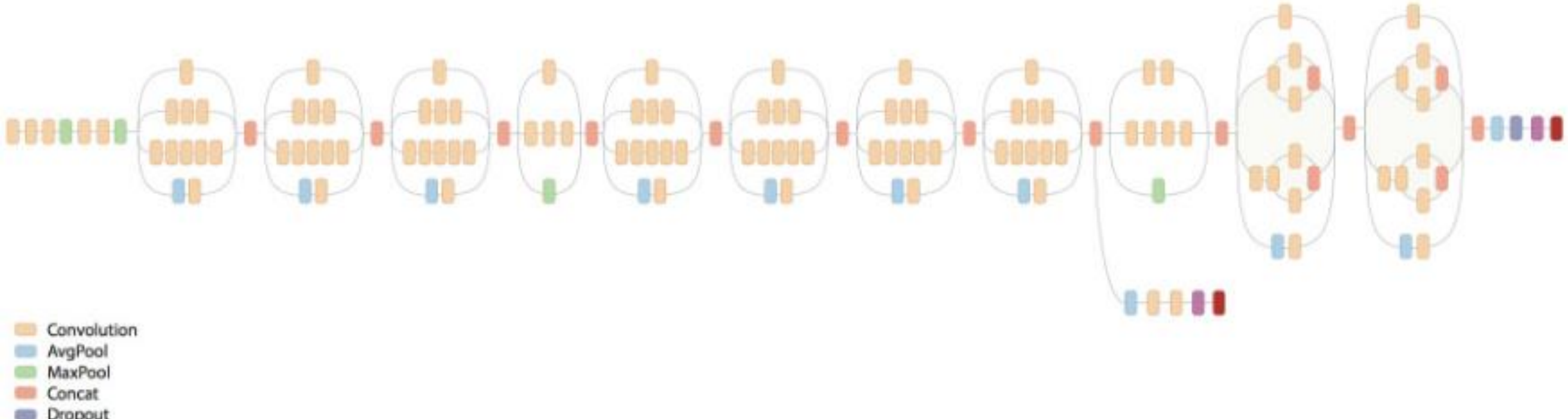


GoogleLeNet
(22 layers)



Google : Inception

- Evaluation result
  - Top-1 error : 21.2%
  - Top-5 error : 5.6%
- The Model Architecture



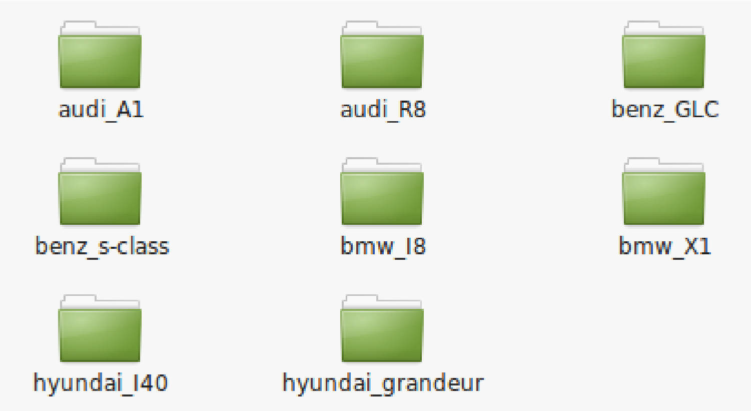
논문 : Rethinking the Inception Architecture for Computer Vision(C. Szegrdy et

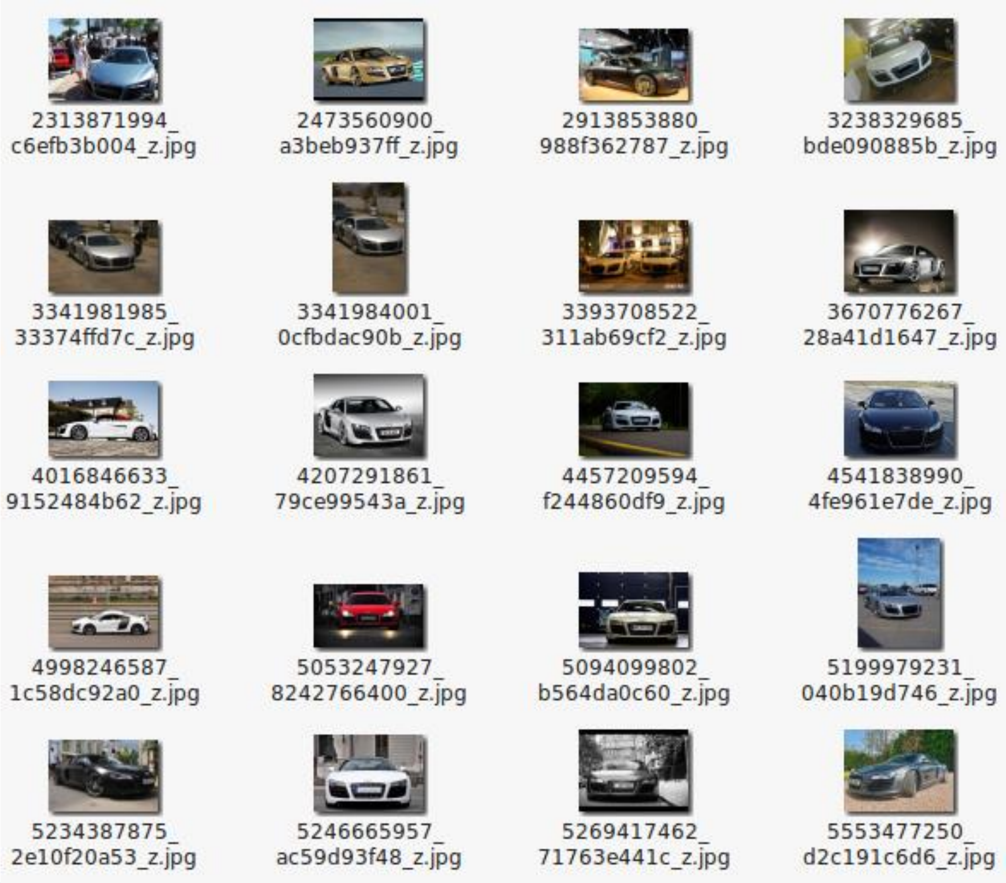
Train your own image classifier with Inception in TensorFlow

- Ubuntu 14.04 (64 bit)
- Python 2.7.6
- TensorFlow 0.8.0 (CPU only)
- Bazel 0.2.3

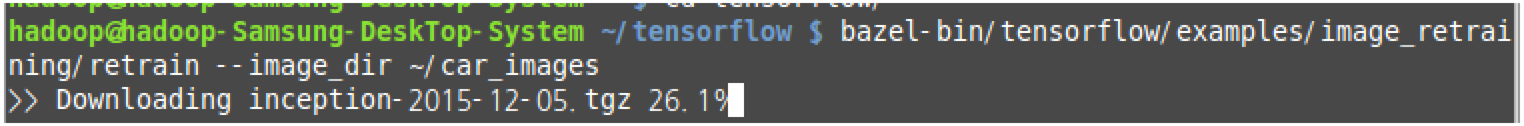
Train your own image classifier with Inception in TensorFlow

- Training set – 91.5 MB (854)

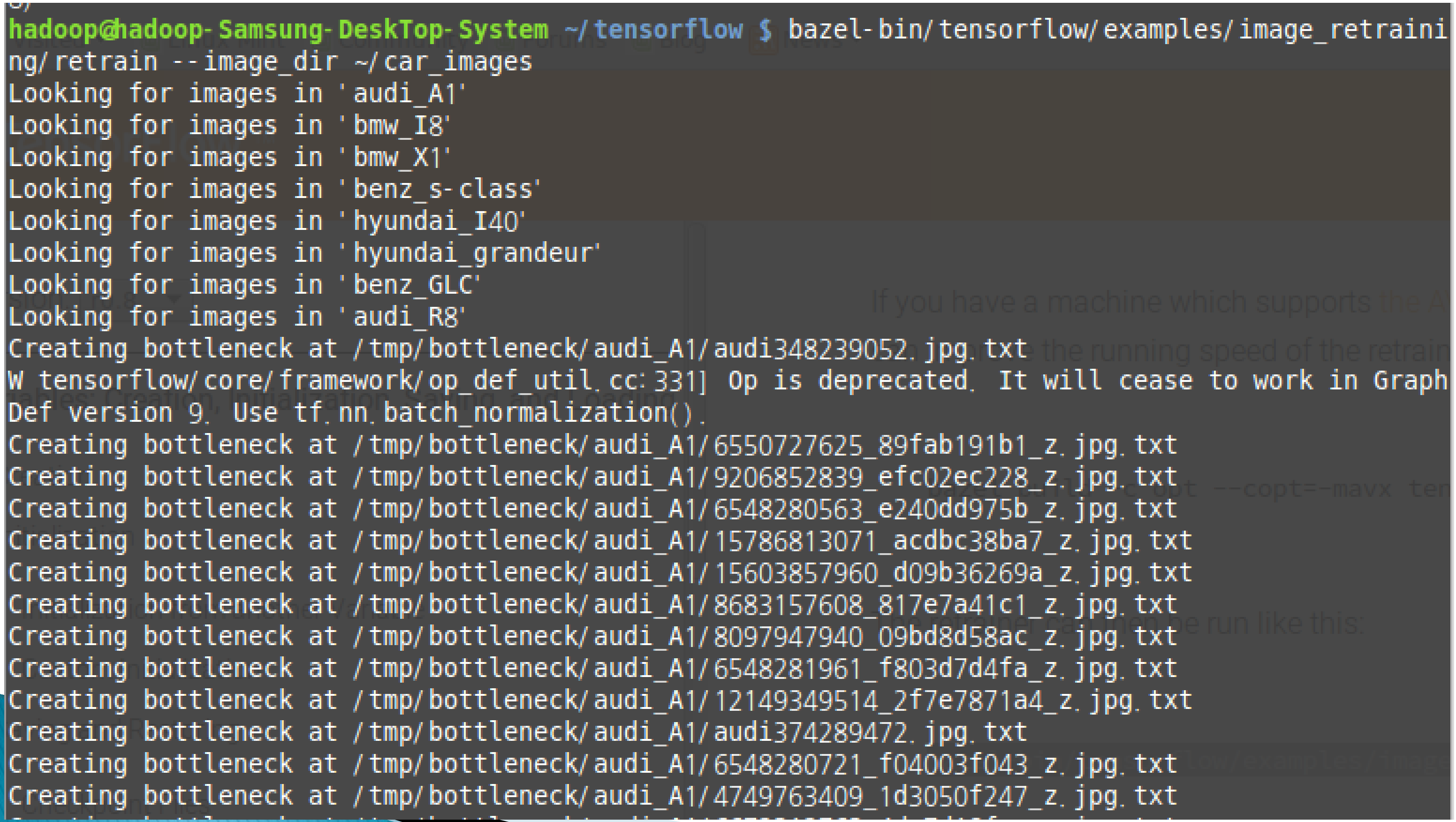



2313871994_c6efb3b004_z.jpg
2473560900_a3beb937ff_z.jpg
2913853880_988f362787_z.jpg
3238329685_bde090885b_z.jpg
3341981985_33374ffd7c_z.jpg
3341984001_0cfbdac90b_z.jpg
3393708522_311ab69cf2_z.jpg
3670776267_28a41d1647_z.jpg
4016846633_9152484b62_z.jpg
4207291861_79ce99543a_z.jpg
4457209594_f244860df9_z.jpg
4541838990_4fe961e7de_z.jpg
4998246587_1c58dc92a0_z.jpg
5053247927_8242766400_z.jpg
5094099802_b564da0c60_z.jpg
5199979231_040b19d746_z.jpg
5234387875_2e10f20a53_z.jpg
5246665957_ac59d93f48_z.jpg
5269417462_71763e441c_z.jpg
5553477250_d2c191c6d6_z.jpg

Train your own image classifier with Inception in TensorFlow

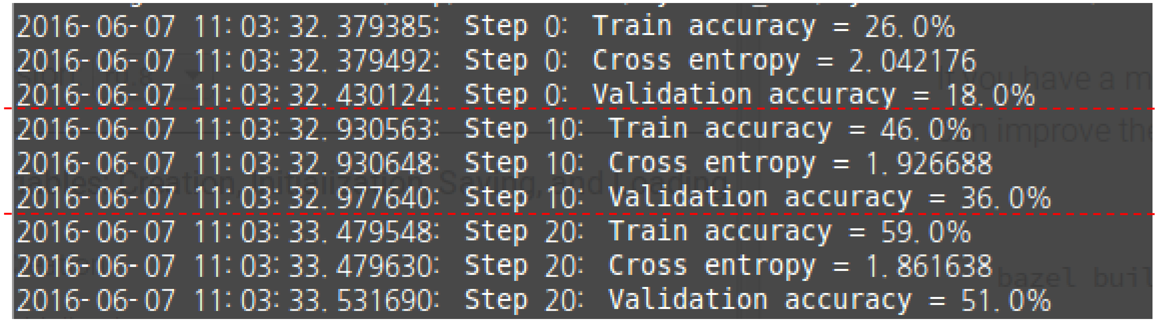


```
hadoop@hadoop-Samsung-DeskTop-System ~/tensorflow $ bazel-bin/tensorflow/examples/image_retrai
ning/retrain --image_dir ~/car_images
>> Downloading inception-2015-12-05.tgz 26.1%
```

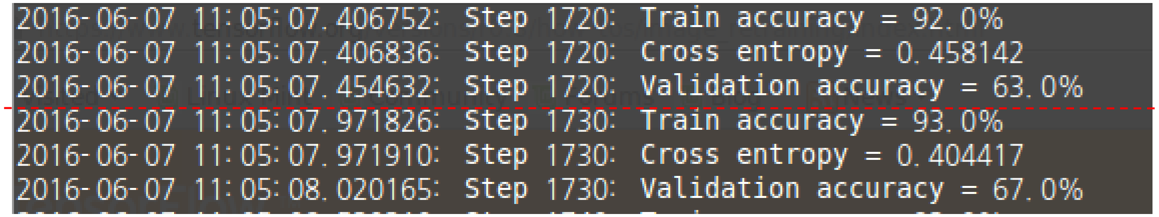


```
hadoop@hadoop-Samsung-DeskTop-System ~/tensorflow $ bazel-bin/tensorflow/examples/image_retraini
ng/retrain --image_dir ~/car_images
Looking for images in 'audi_A1'
Looking for images in 'bmw_I8'
Looking for images in 'bmw_X1'
Looking for images in 'benz_s-class'
Looking for images in 'hyundai_I40'
Looking for images in 'hyundai_grandeur'
Looking for images in 'benz_GLC'
Looking for images in 'audi_R8'
Creating bottleneck at /tmp/bottleneck/audi_A1/audi348239052.jpg.txt
W tensorflow/core/framework/op_def_util.cc:331] Op is deprecated. It will cease to work in Graph
Def version 9. Use tf.nn.batch_normalization().
Creating bottleneck at /tmp/bottleneck/audi_A1/6550727625_89fab191b1_z.jpg.txt
Creating bottleneck at /tmp/bottleneck/audi_A1/9206852839_efc02ec228_z.jpg.txt
Creating bottleneck at /tmp/bottleneck/audi_A1/6548280563_e240dd975b_z.jpg.txt
Creating bottleneck at /tmp/bottleneck/audi_A1/15786813071_acdbc38ba7_z.jpg.txt
Creating bottleneck at /tmp/bottleneck/audi_A1/15603857960_d09b36269a_z.jpg.txt
Creating bottleneck at /tmp/bottleneck/audi_A1/8683157608_817e7a41c1_z.jpg.txt
Creating bottleneck at /tmp/bottleneck/audi_A1/8097947940_09bd8d58ac_z.jpg.txt
Creating bottleneck at /tmp/bottleneck/audi_A1/6548281961_f803d7d4fa_z.jpg.txt
Creating bottleneck at /tmp/bottleneck/audi_A1/12149349514_2f7e7871a4_z.jpg.txt
Creating bottleneck at /tmp/bottleneck/audi_A1/audi374289472.jpg.txt
Creating bottleneck at /tmp/bottleneck/audi_A1/6548280721_f04003f043_z.jpg.txt
Creating bottleneck at /tmp/bottleneck/audi_A1/4749763409_1d3050f247_z.jpg.txt
```

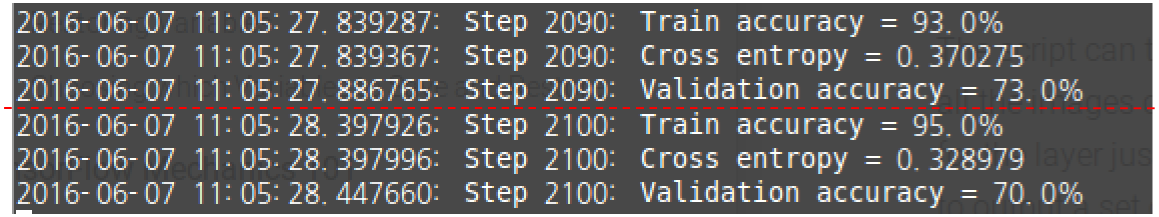
Train your own image classifier with Inception in TensorFlow



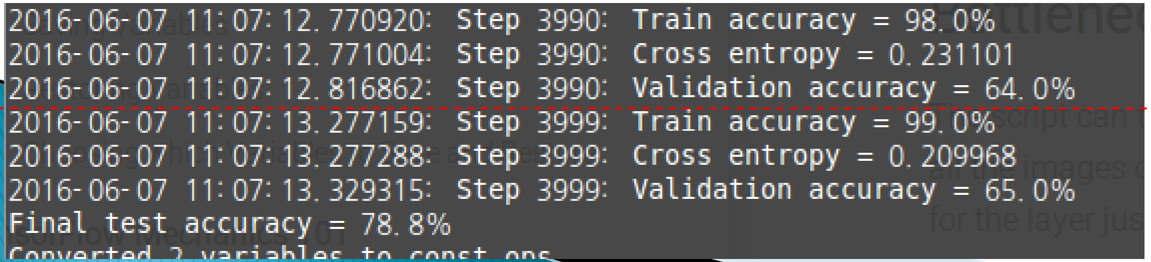
```
2016-06-07 11:03:32.379385: Step 0: Train accuracy = 26.0%
2016-06-07 11:03:32.379492: Step 0: Cross entropy = 2.042176
2016-06-07 11:03:32.430124: Step 0: Validation accuracy = 18.0%
2016-06-07 11:03:32.930563: Step 10: Train accuracy = 46.0%
2016-06-07 11:03:32.930648: Step 10: Cross entropy = 1.926688
2016-06-07 11:03:32.977640: Step 10: Validation accuracy = 36.0%
2016-06-07 11:03:33.479548: Step 20: Train accuracy = 59.0%
2016-06-07 11:03:33.479630: Step 20: Cross entropy = 1.861638
2016-06-07 11:03:33.531690: Step 20: Validation accuracy = 51.0%
```



```
2016-06-07 11:05:07.406752: Step 1720: Train accuracy = 92.0%
2016-06-07 11:05:07.406836: Step 1720: Cross entropy = 0.458142
2016-06-07 11:05:07.454632: Step 1720: Validation accuracy = 63.0%
2016-06-07 11:05:07.971826: Step 1730: Train accuracy = 93.0%
2016-06-07 11:05:07.971910: Step 1730: Cross entropy = 0.404417
2016-06-07 11:05:08.020165: Step 1730: Validation accuracy = 67.0%
```

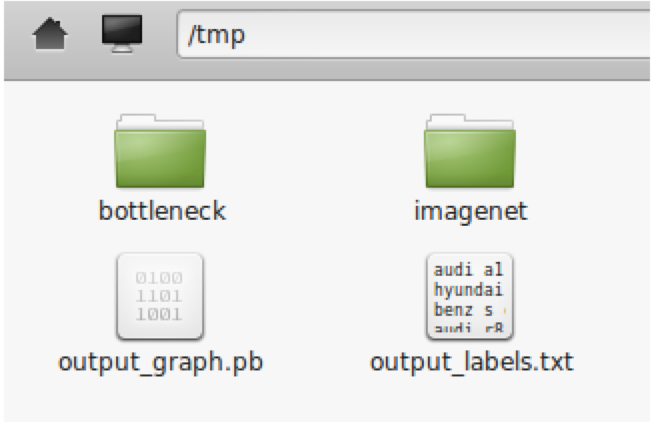


```
2016-06-07 11:05:27.839287: Step 2090: Train accuracy = 93.0%
2016-06-07 11:05:27.839367: Step 2090: Cross entropy = 0.370275
2016-06-07 11:05:27.886765: Step 2090: Validation accuracy = 73.0%
2016-06-07 11:05:28.397926: Step 2100: Train accuracy = 95.0%
2016-06-07 11:05:28.397996: Step 2100: Cross entropy = 0.328979
2016-06-07 11:05:28.447660: Step 2100: Validation accuracy = 70.0%
```

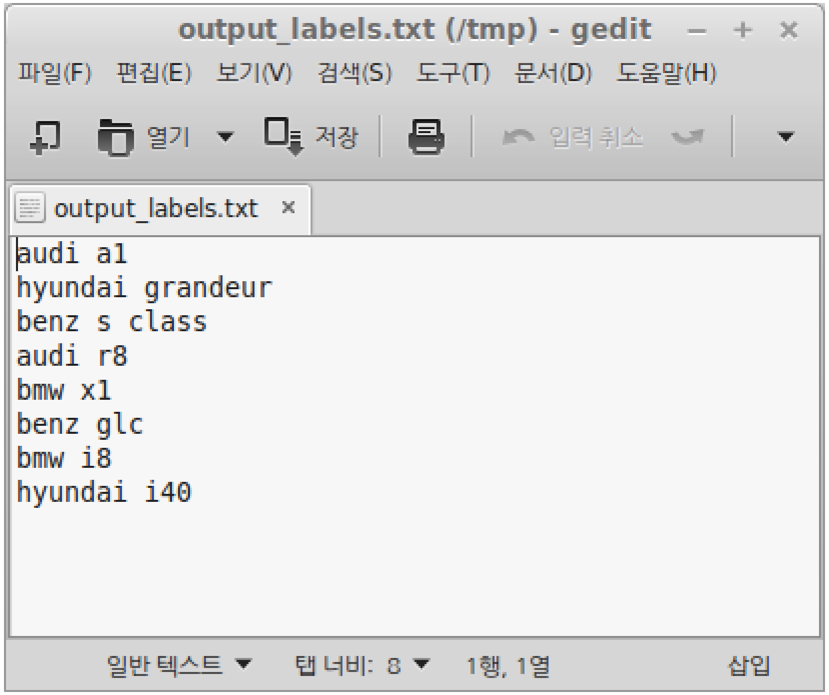


```
2016-06-07 11:07:12.770920: Step 3990: Train accuracy = 98.0%
2016-06-07 11:07:12.771004: Step 3990: Cross entropy = 0.231101
2016-06-07 11:07:12.816862: Step 3990: Validation accuracy = 64.0%
2016-06-07 11:07:13.277159: Step 3999: Train accuracy = 99.0%
2016-06-07 11:07:13.277288: Step 3999: Cross entropy = 0.209968
2016-06-07 11:07:13.329315: Step 3999: Validation accuracy = 65.0%
Final test accuracy = 78.8%
Converted 2 variables to const ops.
```

Train your own image classifier with Inception in TensorFlow

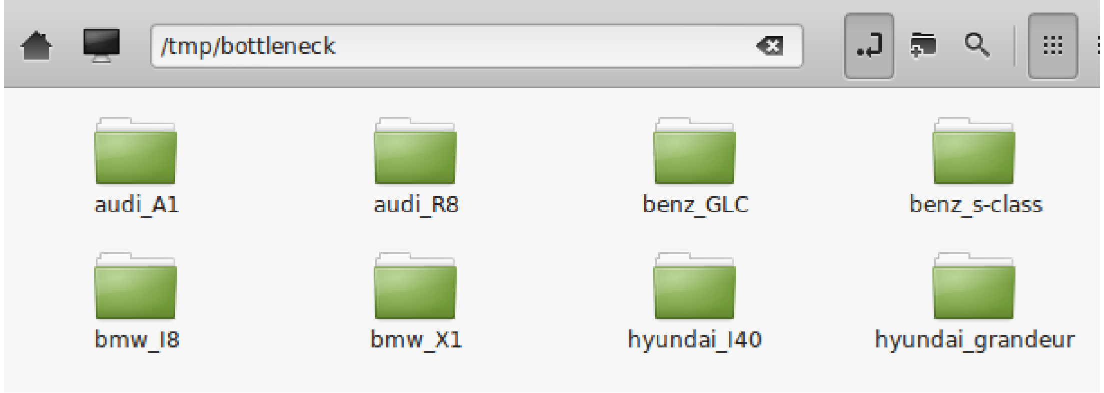


output_labels

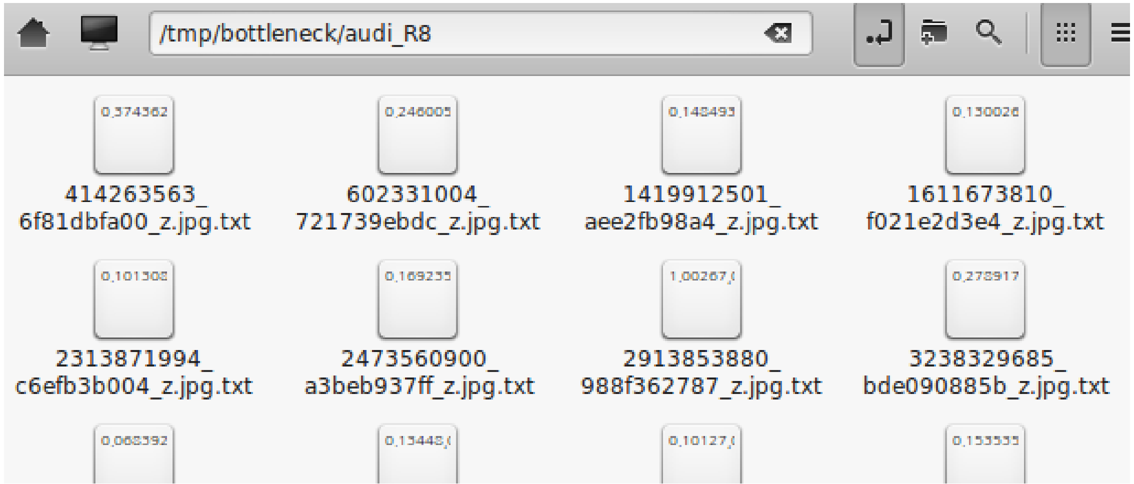

output_labels.txt (/tmp) - gedit
파일(F) 편집(E) 보기(V) 검색(S) 도구(T) 문서(D) 도움말(H)
열기
저장
입력 취소
output_labels.txt
일반 텍스트
탭 너비: 8
1행, 1열
삽입

```
audi a1
hyundai grandeur
benz s class
audi r8
bmw x1
benz glc
bmw i8
hyundai i40
```

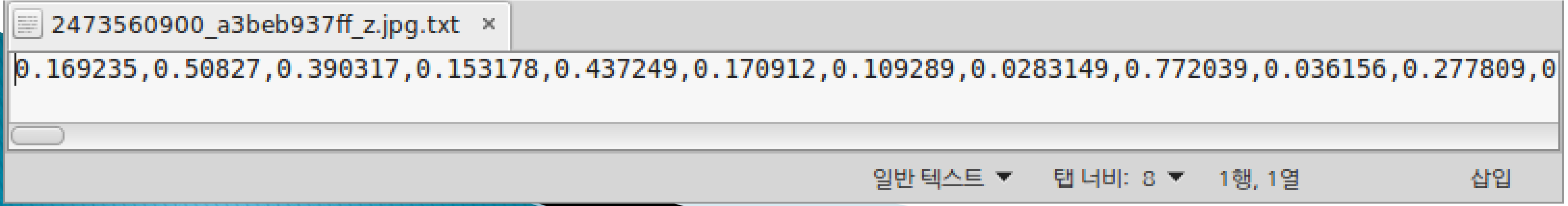
Train your own image classifier with Inception in TensorFlow



audi_A1
audi_R8
benz_GLC
benz_s-class
bmw_I8
bmw_X1
hyundai_I40
hyundai_grandeur



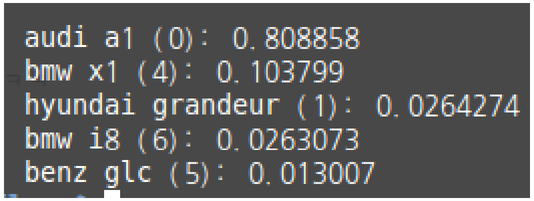
414263563_6f81dbfa00_z.jpg.txt
602331004_721739ebdc_z.jpg.txt
1419912501_aee2fb98a4_z.jpg.txt
1611673810_f021e2d3e4_z.jpg.txt
2313871994_c6efb3b004_z.jpg.txt
2473560900_a3beb937ff_z.jpg.txt
2913853880_988f362787_z.jpg.txt
3238329685_bde090885b_z.jpg.txt



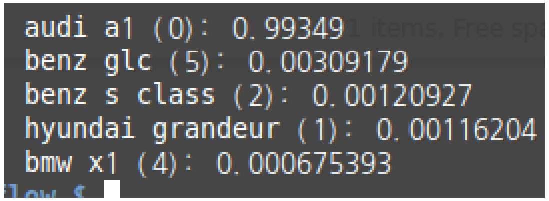
```
0.169235,0.50827,0.390317,0.153178,0.437249,0.170912,0.109289,0.0283149,0.772039,0.036156,0.277809,0
```

Train your own image classifier with Inception in TensorFlow

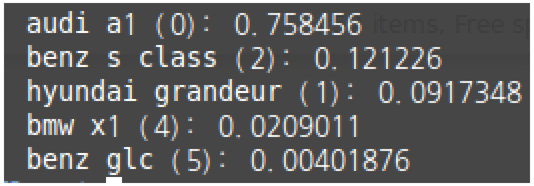
Audi a1



```
audi a1 (0): 0.808858
bmw x1 (4): 0.103799
hyundai grandeur (1): 0.0264274
bmw i8 (6): 0.0263073
benz glc (5): 0.013007
```

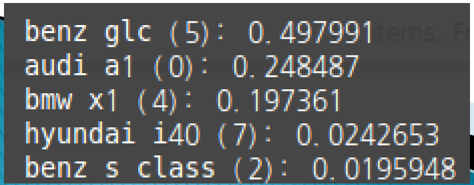


```
audi a1 (0): 0.99349
benz glc (5): 0.00309179
benz s class (2): 0.00120927
hyundai grandeur (1): 0.00116204
bmw x1 (4): 0.000675393
```



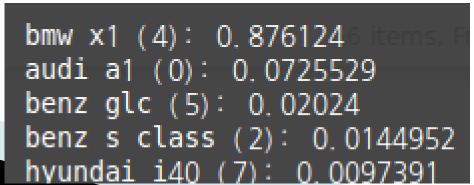
```
audi a1 (0): 0.758456
benz s class (2): 0.121226
hyundai grandeur (1): 0.0917348
bmw x1 (4): 0.0209011
benz glc (5): 0.00401876
```

Bmw x1

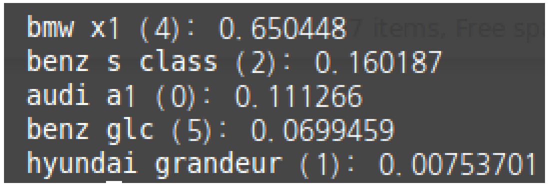


```
benz glc (5): 0.497991
audi a1 (0): 0.248487
bmw x1 (4): 0.197361
hyundai i40 (7): 0.0242653
benz s class (2): 0.0195948
```

✓



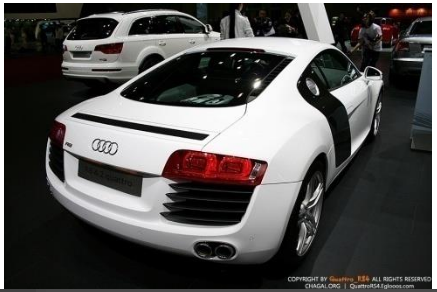
```
bmw x1 (4): 0.876124
audi a1 (0): 0.0725529
benz glc (5): 0.02024
benz s class (2): 0.0144952
hyundai i40 (7): 0.0097391
```

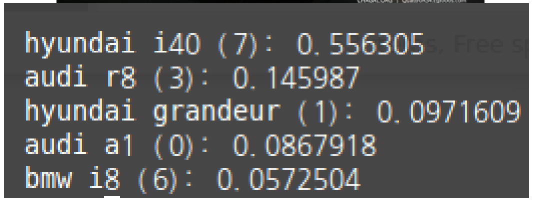


```
bmw x1 (4): 0.650448
benz s class (2): 0.160187
audi a1 (0): 0.111266
benz glc (5): 0.0699459
hyundai grandeur (1): 0.00753701
```

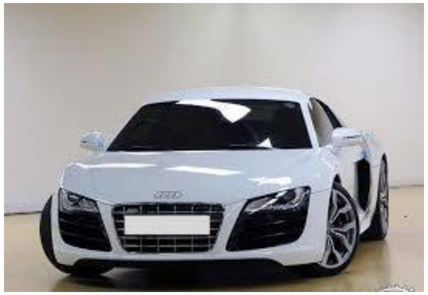
Train your own image classifier with Inception in TensorFlow

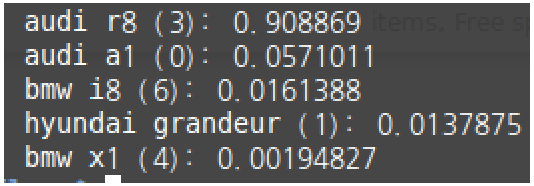
Audi r8



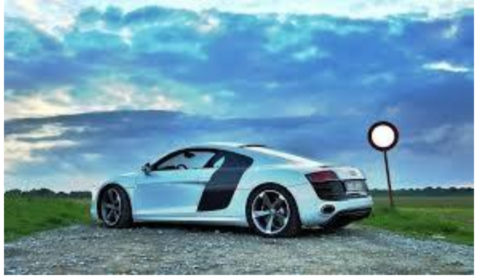


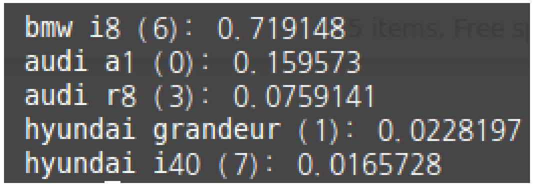
✓
```
hyundai i40 (7): 0.556305
audi r8 (3): 0.145987
hyundai grandeur (1): 0.0971609
audi a1 (0): 0.0867918
bmw i8 (6): 0.0572504
```





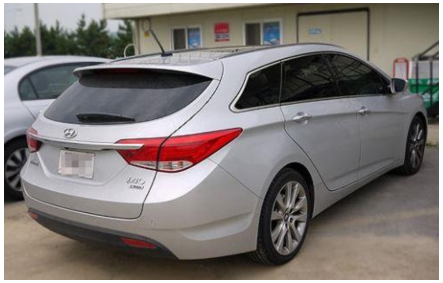
```
audi r8 (3): 0.908869
audi a1 (0): 0.0571011
bmw i8 (6): 0.0161388
hyundai grandeur (1): 0.0137875
bmw x1 (4): 0.00194827
```

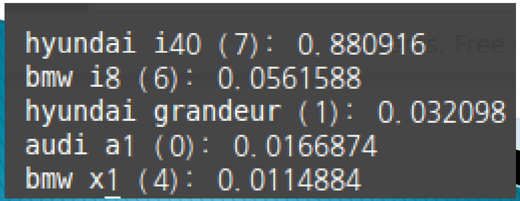




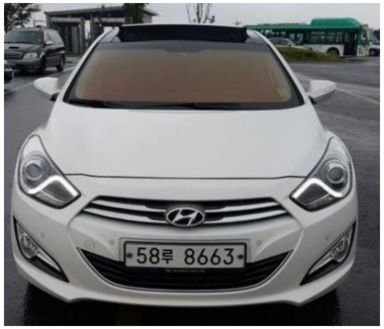
✓
```
bmw i8 (6): 0.719148
audi a1 (0): 0.159573
audi r8 (3): 0.0759141
hyundai grandeur (1): 0.0228197
hyundai i40 (7): 0.0165728
```

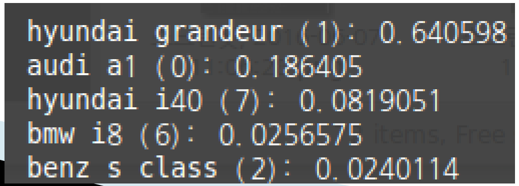
Hyundai i40



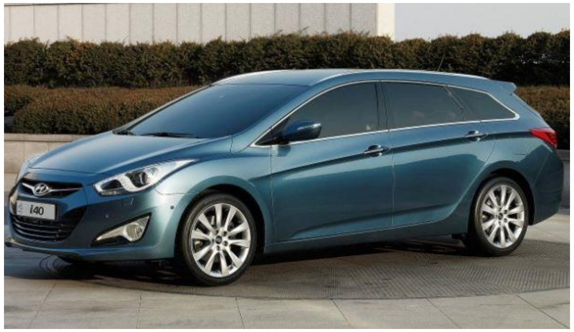


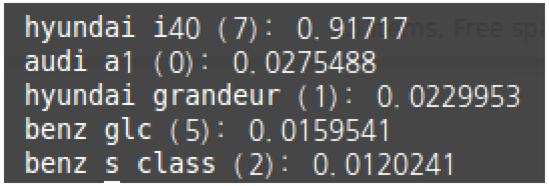
```
hyundai i40 (7): 0.880916
bmw i8 (6): 0.0561588
hyundai grandeur (1): 0.032098
audi a1 (0): 0.0166874
bmw x1 (4): 0.0114884
```





✓
```
hyundai grandeur (1): 0.640598
audi a1 (0): 0.186405
hyundai i40 (7): 0.0819051
bmw i8 (6): 0.0256575
benz s class (2): 0.0240114
```





```
hyundai i40 (7): 0.91717
audi a1 (0): 0.0275488
hyundai grandeur (1): 0.0229953
benz glc (5): 0.0159541
benz s class (2): 0.0120241
```

Train your own image classifier with Inception in TensorFlow

Training set

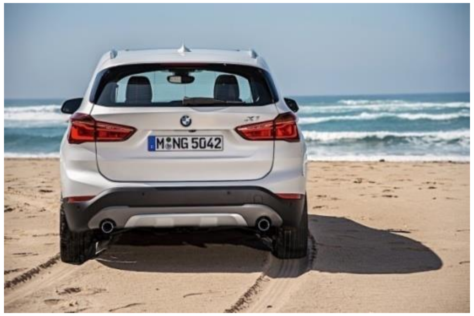
91.5 MB (854) → 97.8 MB (902)

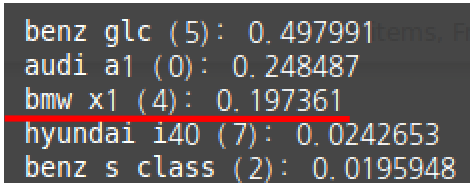
Final test accuracy

78.8 % → 77.4 %

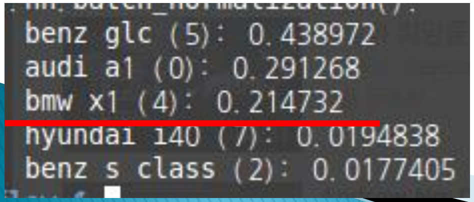
Train your own image classifier with Inception in TensorFlow

Bmw x1



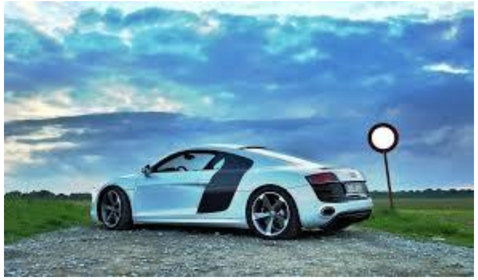


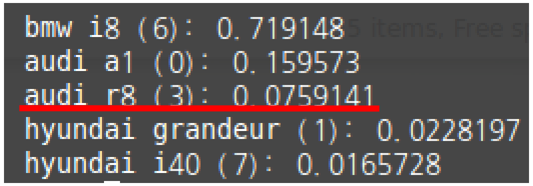
```
benz glc (5): 0.497991
audi a1 (0): 0.248487
bmw x1 (4): 0.197361
hyundai i40 (7): 0.0242653
benz s class (2): 0.0195948
```



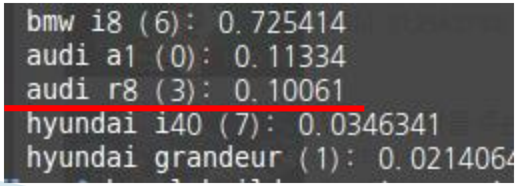
```
benz glc (5): 0.438972
audi a1 (0): 0.291268
bmw x1 (4): 0.214732
hyundai i40 (7): 0.0194838
benz s class (2): 0.0177405
```

Audi r8



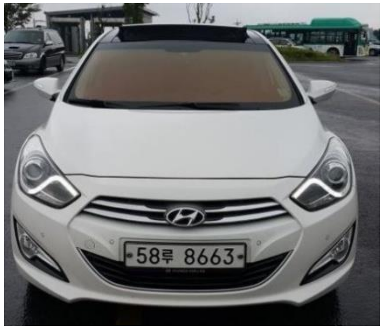


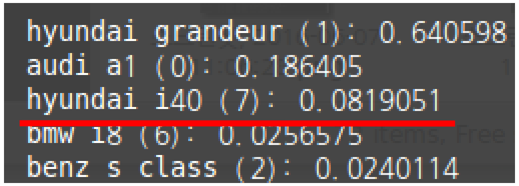
```
bmw i8 (6): 0.719148
audi a1 (0): 0.159573
audi r8 (3): 0.0759141
hyundai grandeur (1): 0.0228197
hyundai i40 (7): 0.0165728
```



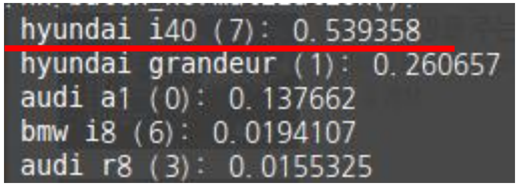
```
bmw i8 (6): 0.725414
audi a1 (0): 0.11334
audi r8 (3): 0.10061
hyundai i40 (7): 0.0346341
hyundai grandeur (1): 0.0214064
```

Hyundai i40





```
hyundai grandeur (1): 0.640598
audi a1 (0): 0.186405
hyundai i40 (7): 0.0819051
bmw i8 (6): 0.0256575
benz s class (2): 0.0240114
```



```
hyundai i40 (7): 0.539358
hyundai grandeur (1): 0.260657
audi a1 (0): 0.137662
bmw i8 (6): 0.0194107
audi r8 (3): 0.0155325
```

References

- C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015.
- C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. arXiv preprint arXiv:1512.00567, 2015.
- TensorFlow official home page (https://www.tensorflow.org)
- TensorFlow Github (https://github.com/tensorflow/models/tree/master/inception)
- Laonpeople Naver Blog (http://laonple.blog.me)