Learning Deep Features for Discriminative Localization

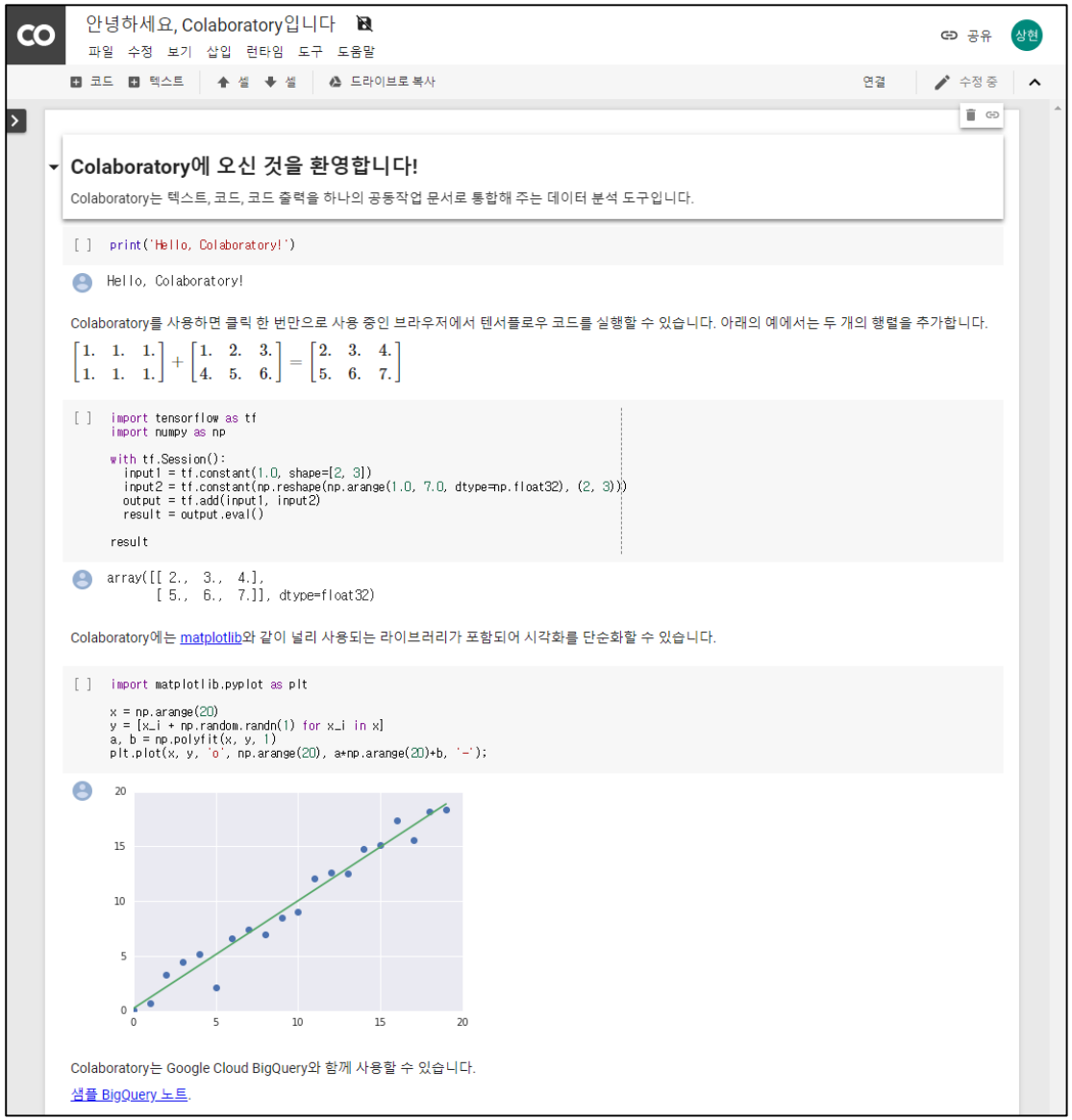
Sanghyun Seo

2018-04-13

Contents

Google Colaboratory

- Collaboration + jupyter notebook



Google Colaboratory

- Tesla K80(12.6gb GPU) ≥ GTX 1080TI(11gb GPU)

Untitled0.ipynb

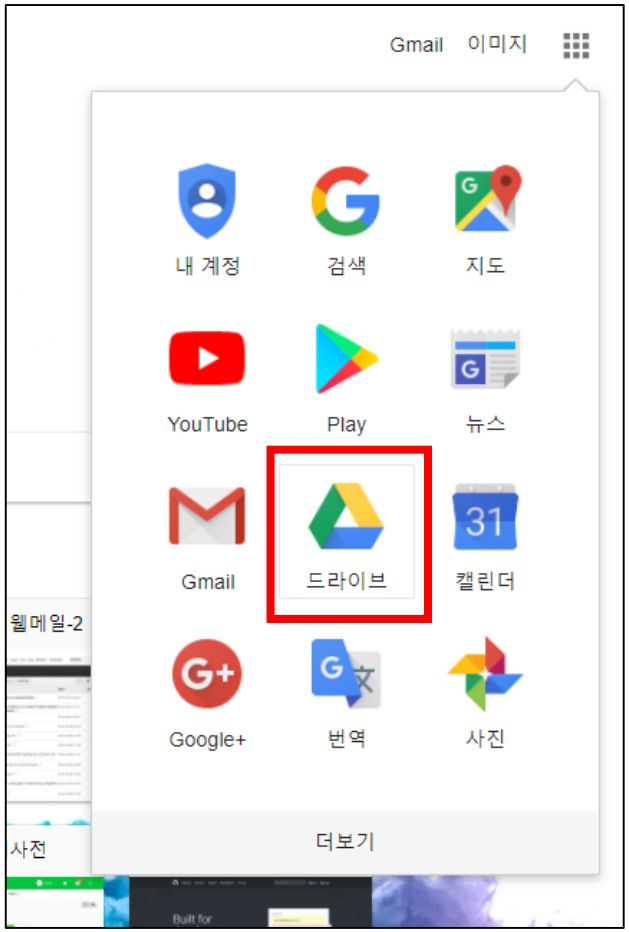
파일 수정 보기 삽입 런타임 도구 도움말

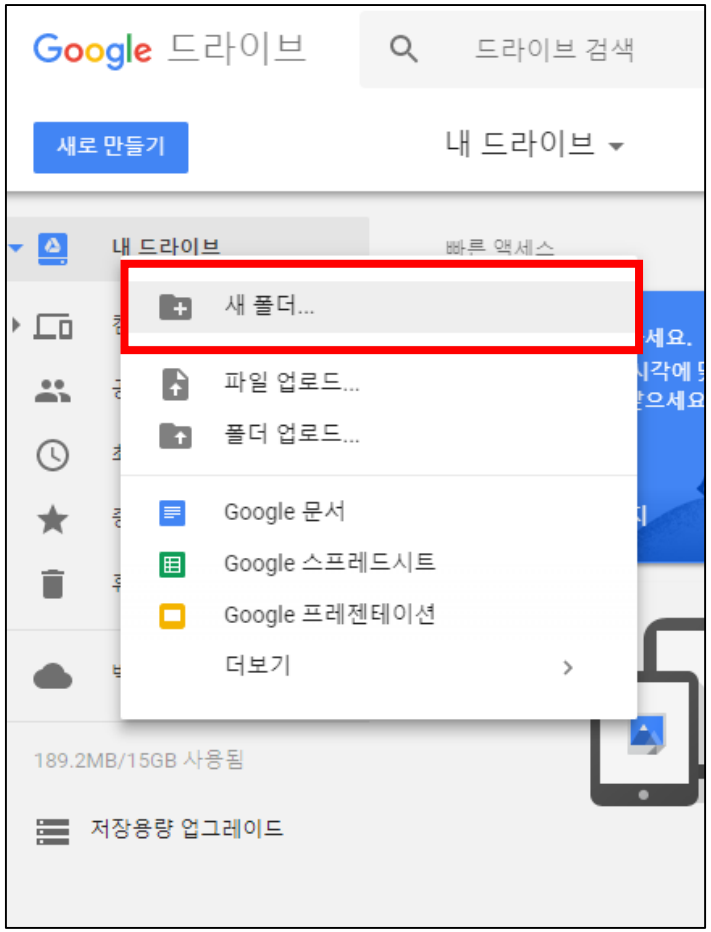
```
# memory footprint support libraries/code
!ln -sf /opt/bin/nvidia-smi /usr/bin/nvidia-smi
!pip install gputil
!pip install psutil
!pip install humanize
import psutil
import humanize
import os
import GPUtil as GPU
GPUs = GPU.getGPUs()
# XXX: only one GPU on Colab and isn't guaranteed
gpu = GPUs[0]
print(gpu.__dict__["name"])
def printm():
  process = psutil.Process(os.getpid())
  print("Gen RAM Free: " + humanize.naturalsize( psutil.virtual_memory().available ), " | Proc size: " + humanize.naturalsize( process.memo
  print("GPU RAM Free: {0:.0f}MB | Used: {1:.0f}MB | Util {2:3.0f}% | Total {3:.0f}MB".format(gpu.memoryFree, gpu.memoryUsed, gpu.memoryUti
printm()
```

```
Requirement already satisfied: gputil in /usr/local/lib/python3.6/dist-packages
Requirement already satisfied: numpy in /usr/local/lib/python3.6/dist-packages (from gputil)
Requirement already satisfied: psutil in /usr/local/lib/python3.6/dist-packages
Requirement already satisfied: humanize in /usr/local/lib/python3.6/dist-packages
Tesla K80
Gen RAM Free: 12.6 GB  | Proc size: 132.1 MB
GPU RAM Free: 11439MB | Used: 0MB | Util   0% | Total 11439MB
```

Google Colaboratory

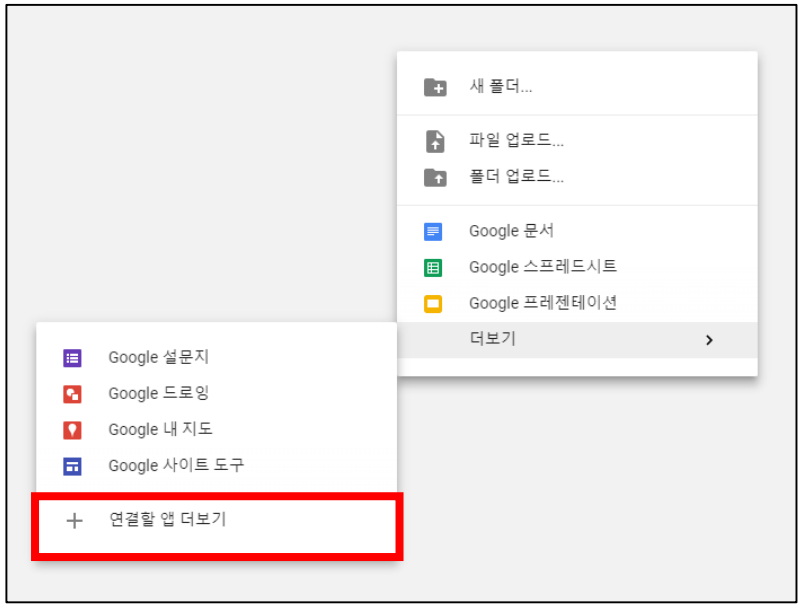
- Login google → go to the google drive → my drive → new folder(Optional)

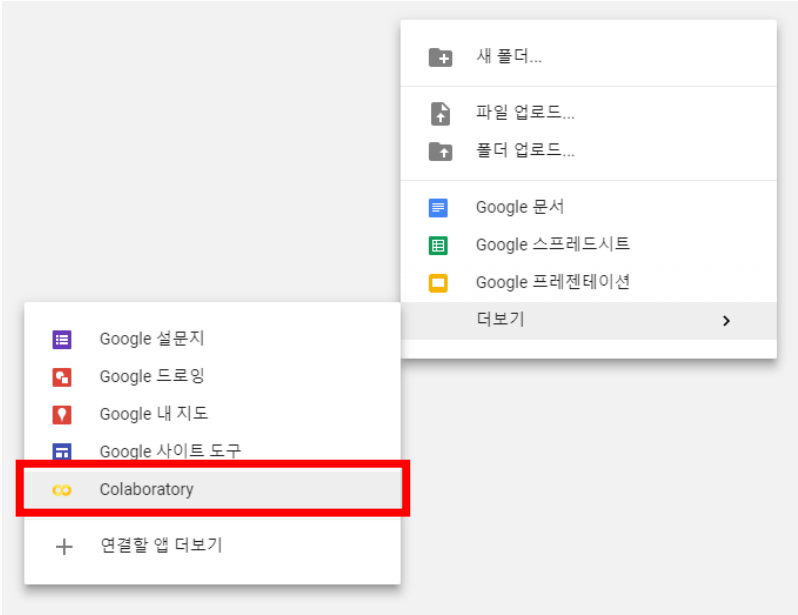


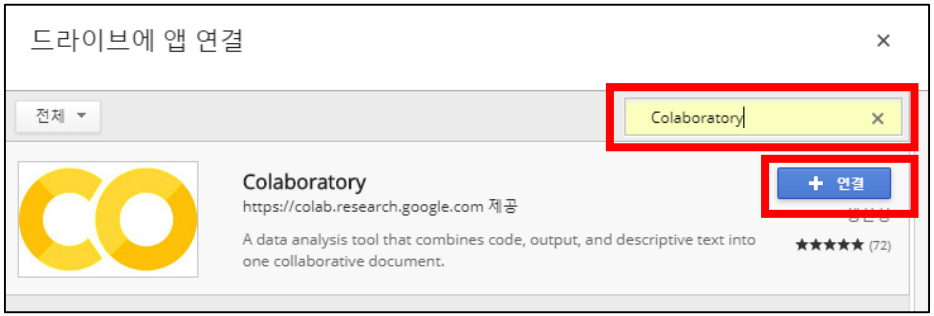


Google Colaboratory

- Mouse right click → See more → See more app to connect → search the Colaboratory → Connect
- Mouse right click → See more → Colaboratory

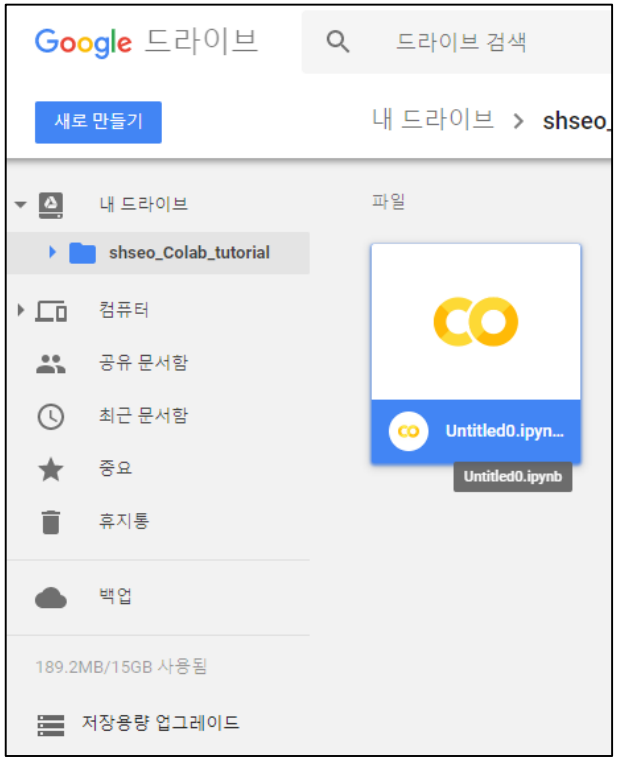


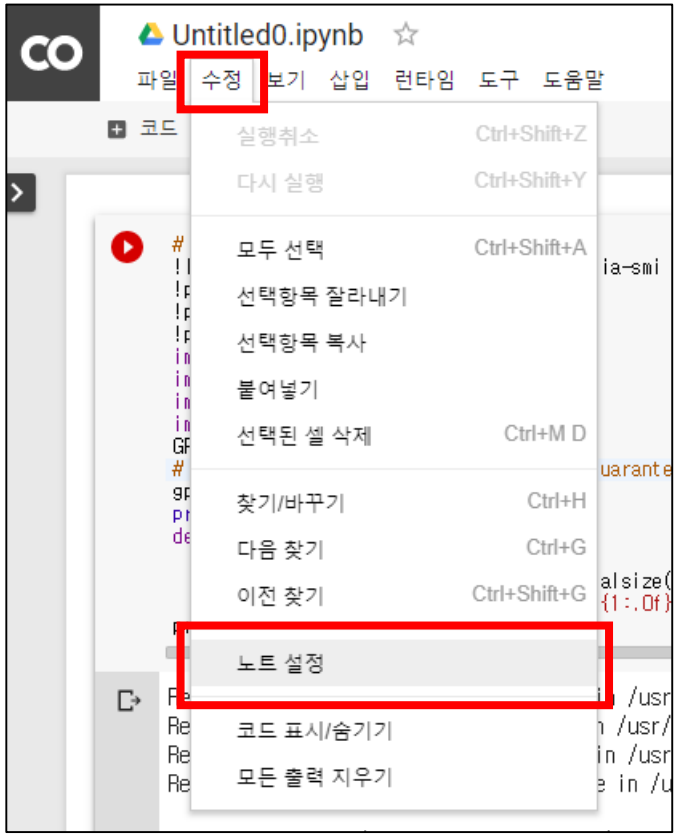


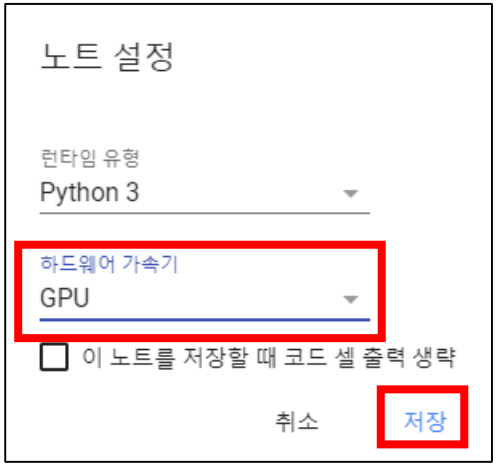


Google Colaboratory

- You can make .ipynb file
- Edit → Setting note → GPU → Save







Google Colaboratory

- Simple test
- Import tensorflow

Google Colaboratory

- Import torch, but there are no module named ‘torch’
- Install TORCH → click insert → run new cell

목차 코드 스니펫

import torch

Install [pytorch](http://pytorch.org/)

Install [pytorch](http://pytorch.org/) 삽입

```
# http://pytorch.org/
from os import path
from wheel.pep425tags import get_abbr_impl,
platform = '{}{}-{}'.format(get_abbr_impl(

accelerator = 'cu80' if path.exists('/opt/b

!pip install -q http://download.pytorch.org
import torch
```

```
[7] import torch

---------------------------------------------------------------------------
ModuleNotFoundError                       Traceback (most recent call last)
<ipython-input-7-eb42ca6e4af3> in <module>()
----> 1 import torch

ModuleNotFoundError: No module named 'torch'

---------------------------------------------------------------------------
NOTE: If your import is failing due to a missing package, you can
manually install dependencies using either !pip or !apt.

To install torch, click the button below.
---------------------------------------------------------------------------
```

INSTALL TORCH SEARCH STACK OVERFLOW

+ 양식

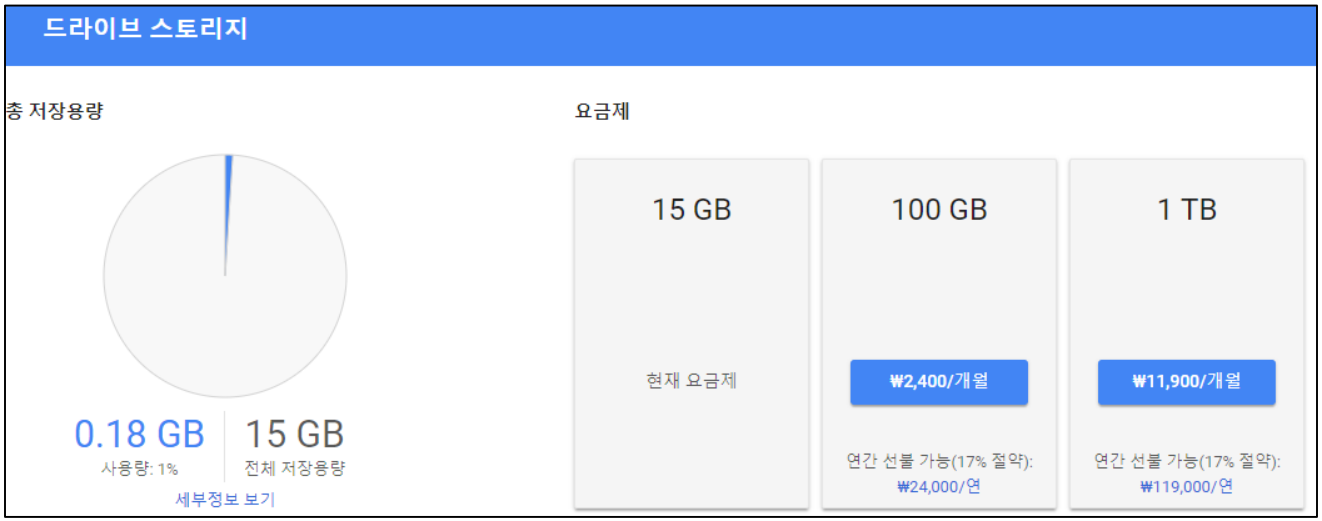
```
# http://pytorch.org/
from os import path
from wheel.pep425tags import get_abbr_impl, get_impl_ver, get_abi_tag
platform = '{}{}-{}'.format(get_abbr_impl(), get_impl_ver(), get_abi_tag())

accelerator = 'cu80' if path.exists('/opt/bin/nvidia-smi') else 'cpu'

!pip install -q http://download.pytorch.org/whl/{accelerator}/torch-0.3.0.post4-{platform}-
import torch
```

```
import torch
print("torch version: %s" % torch.__version__)
```

```
torch version: 0.3.0.post4
```

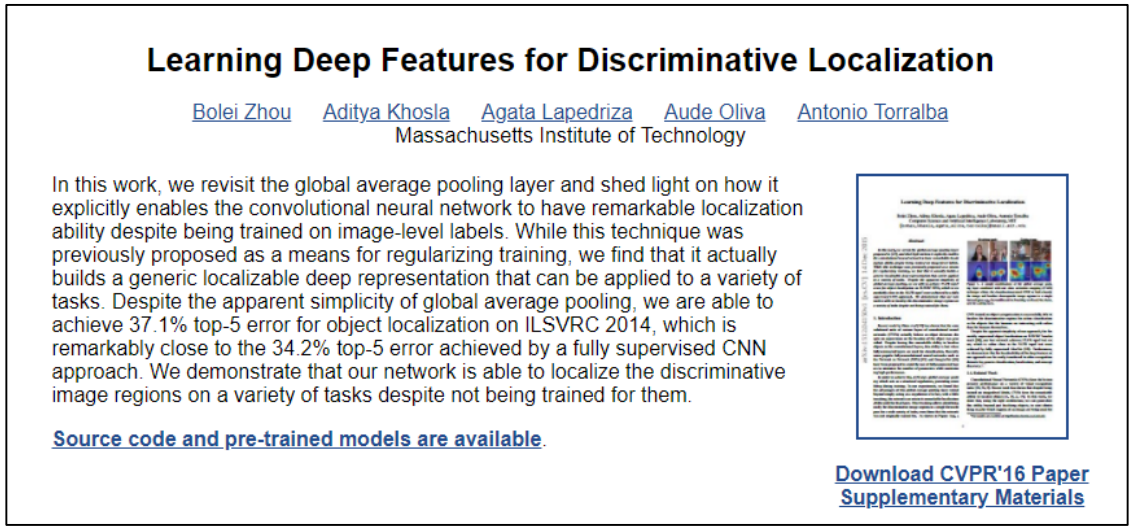
Google Colaboratory

- It appears that the instance will last 12 hours
- It is possible to upload and download files using the google drive
- For example, for every learning epoch, save the .ckpt file on the google drive, and continue to use the saved .ckpt file after 12 hours
- Default is 15gb cloud drive
- By investing about 2,400 won or 11,900 won a month, we can get Tesla- K80(12gb GPU).



Paper Information

- **Title:** Learning Deep Features for Discriminative Localization
- **Author:** Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, Antonio Torralba
- **Affiliation:** Computer Science and Artificial Intelligence Laboratory, MIT
- **Proceeding:** (CVPR 2016) Computer Vision and Pattern Recognition 2016
- http://cnnlocalization.csail.mit.edu/
- https://github.com/metalbubble/CAM



Learning Deep Features for Discriminative Localization

Bolei Zhou Aditya Khosla Agata Lapedriza Aude Oliva Antonio Torralba
Massachusetts Institute of Technology

In this work, we revisit the global average pooling layer and shed light on how it explicitly enables the convolutional neural network to have remarkable localization ability despite being trained on image-level labels. While this technique was previously proposed as a means for regularizing training, we find that it actually builds a generic localizable deep representation that can be applied to a variety of tasks. Despite the apparent simplicity of global average pooling, we are able to achieve 37.1% top-5 error for object localization on ILSVRC 2014, which is remarkably close to the 34.2% top-5 error achieved by a fully supervised CNN approach. We demonstrate that our network is able to localize the discriminative image regions on a variety of tasks despite not being trained for them.

Source code and pre-trained models are available.

Download CVPR'16 Paper
Supplementary Materials

Introduction & Related Works

- Demo video

Introduction & Related Works

- Global Average Pooling
  - Lin, Min, Qiang Chen, and Shuicheng Yan. **"Network in network."** *arXiv preprint arXiv:1312.4400* (2013).

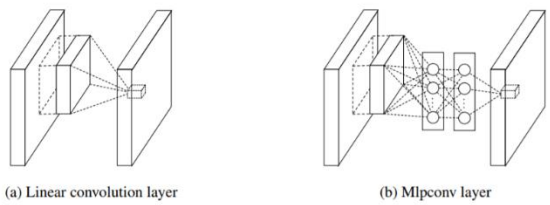


Figure 1: Comparison of linear convolution layer and mlpconv layer. The linear convolution layer includes a linear filter while the mlpconv layer includes a micro network (we choose the multilayer perceptron in this paper). Both layers map the local receptive field to a confidence value of the latent concept.

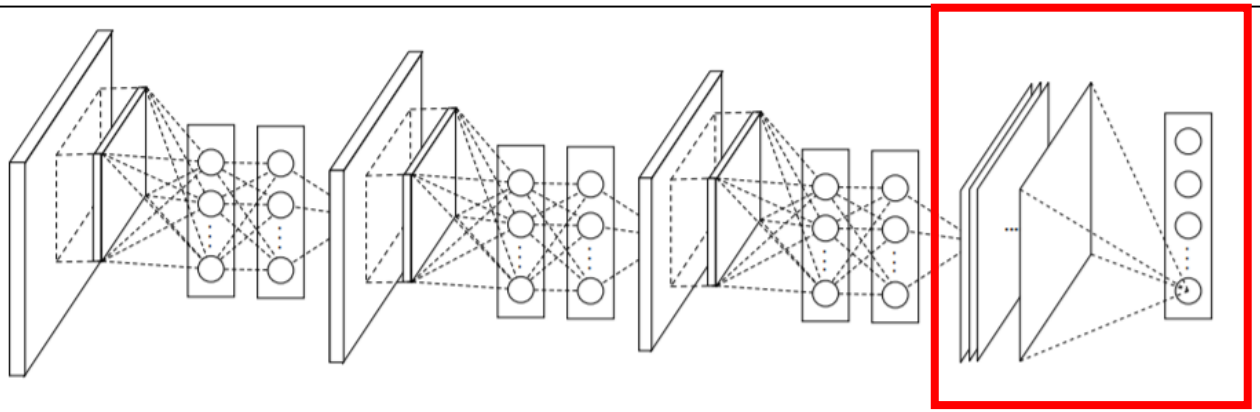
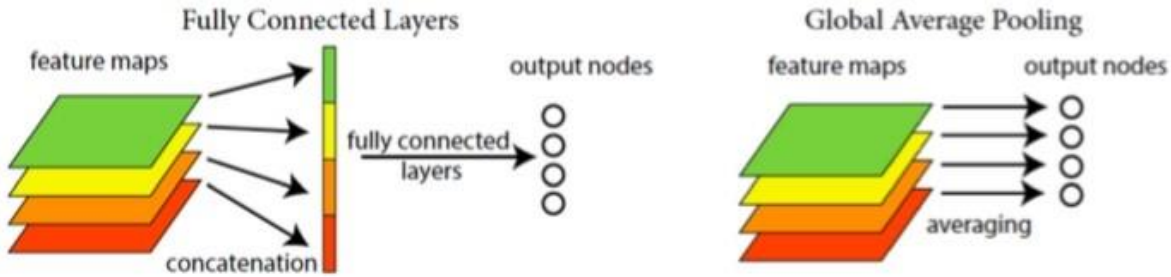
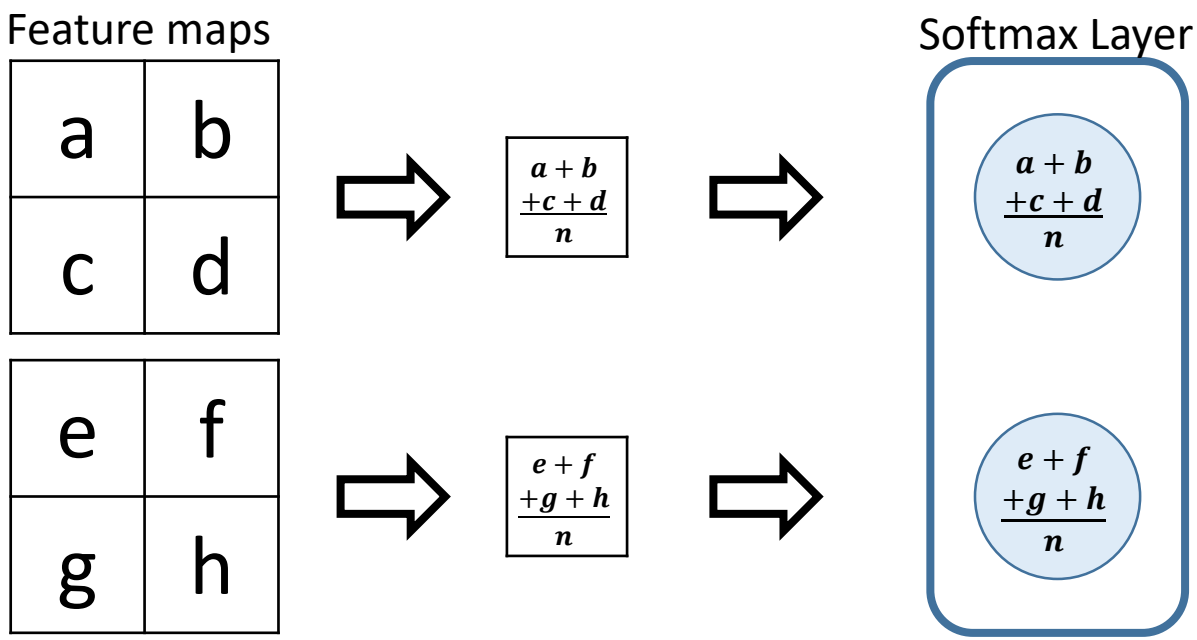


Figure 2: The overall structure of Network In Network. In this paper the NINs include the stacking of three mlpconv layers and one global average pooling layer.

Introduction & Related Works

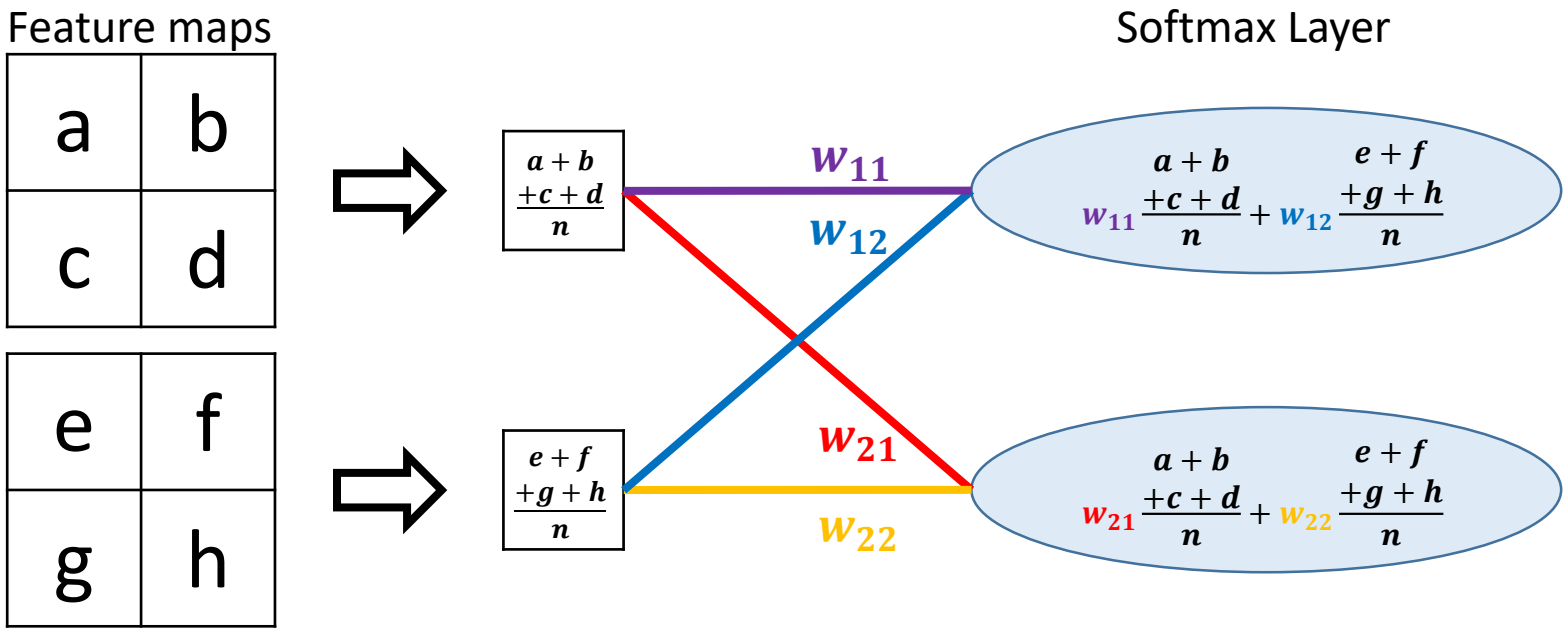
- Golobal Average Pooling
  - Without FC layer?





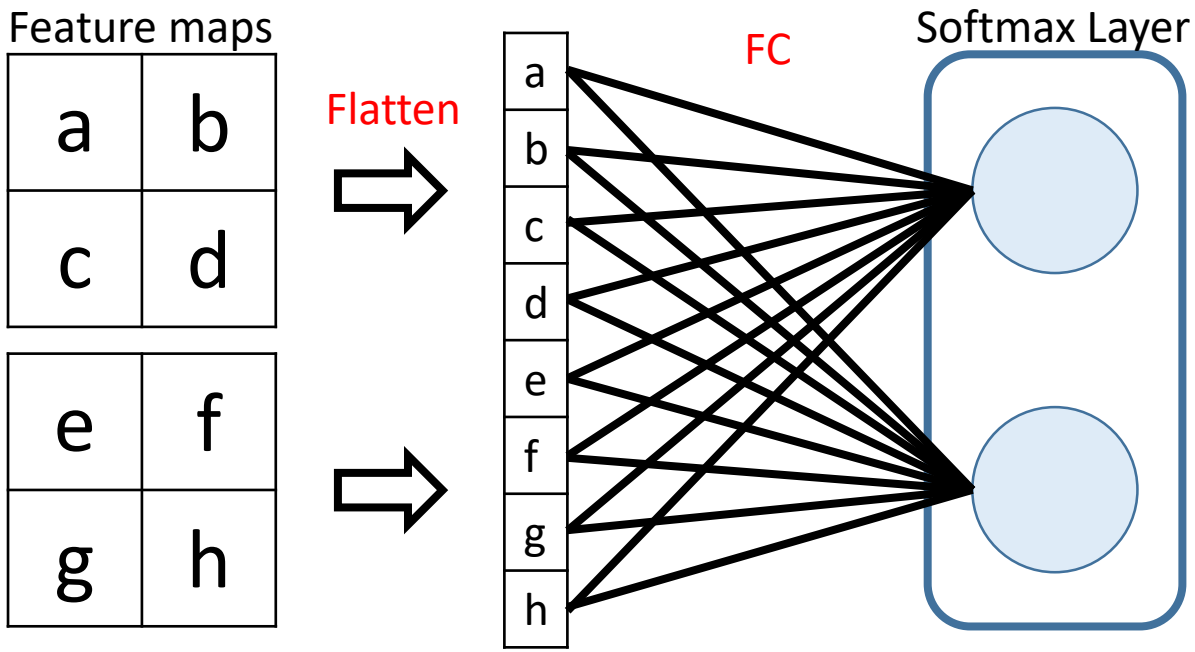
Introduction & Related Works

- Golobal Average Pooling
  - With FC layer
  - The size of feature map: [2, 2, 2] → [1, 1, 2] = 2
  - The number of parameter: [2, 2] = 4



Introduction & Related Works

- Without Global Average Pooling
- The size of feature map: [2, 2, 2] → [1, 8]
- The number of parameter: [8, 2] → 16

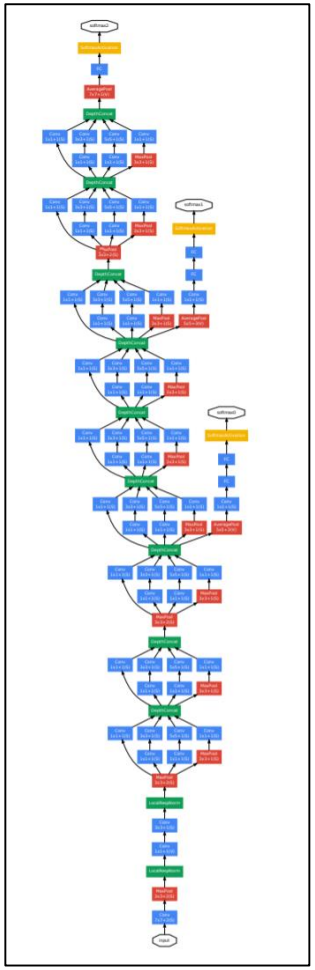


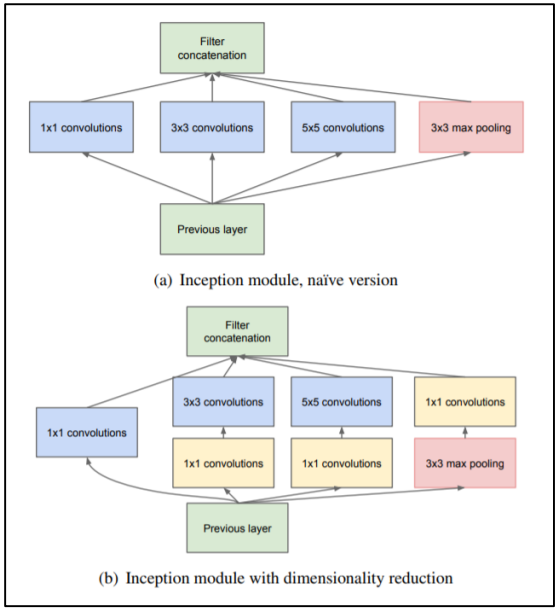
Introduction & Related Works

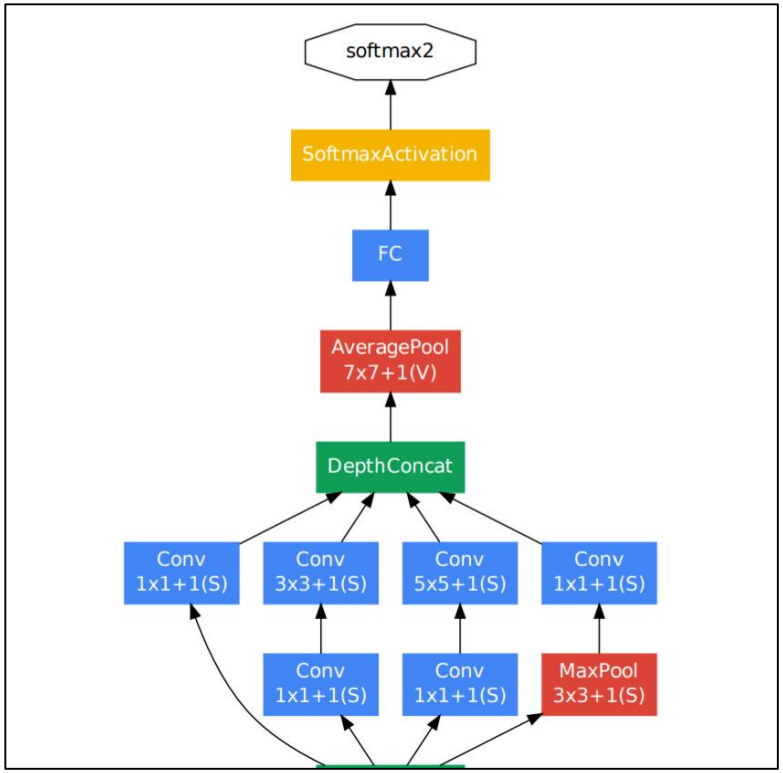
- GAP vs GAP with FC vs Without GAP
- GAP
  - Feature map: [W, H, C] → [1, 1, C] = F
  - The number of parameter: 0
- GAP with FC
  - Feature map: [W, H, C] → [1, 1, C] = F
  - The number of parameter: F * # of output
- Without GAP
  - Feature map: [W, H, C] → [1, W*H*C] = F
  - The number of parameter: F * # of output
- GAP with FC : without GAP ∝ 1: W*H
- If W=10, H=10 → the number of Parameters are 100 Times Different

Introduction & Related Works

- Google Net & GAP with FC







type	patch size/ stride	output size	depth	#1×1	#3×3 reduce	#3×3	#5×5 reduce	#5×5	pool proj	params	ops
avg pool	7×7/1	1×1×1024	0								
dropout (40%)		1×1×1024	0								
linear		1×1×1000	1							1000K	1M
softmax		1×1×1000	0								

Table 1: GoogLeNet incarnation of the Inception architecture.

Class Activation Mapping

- Feature map → GAP → FC → Class Activation Map

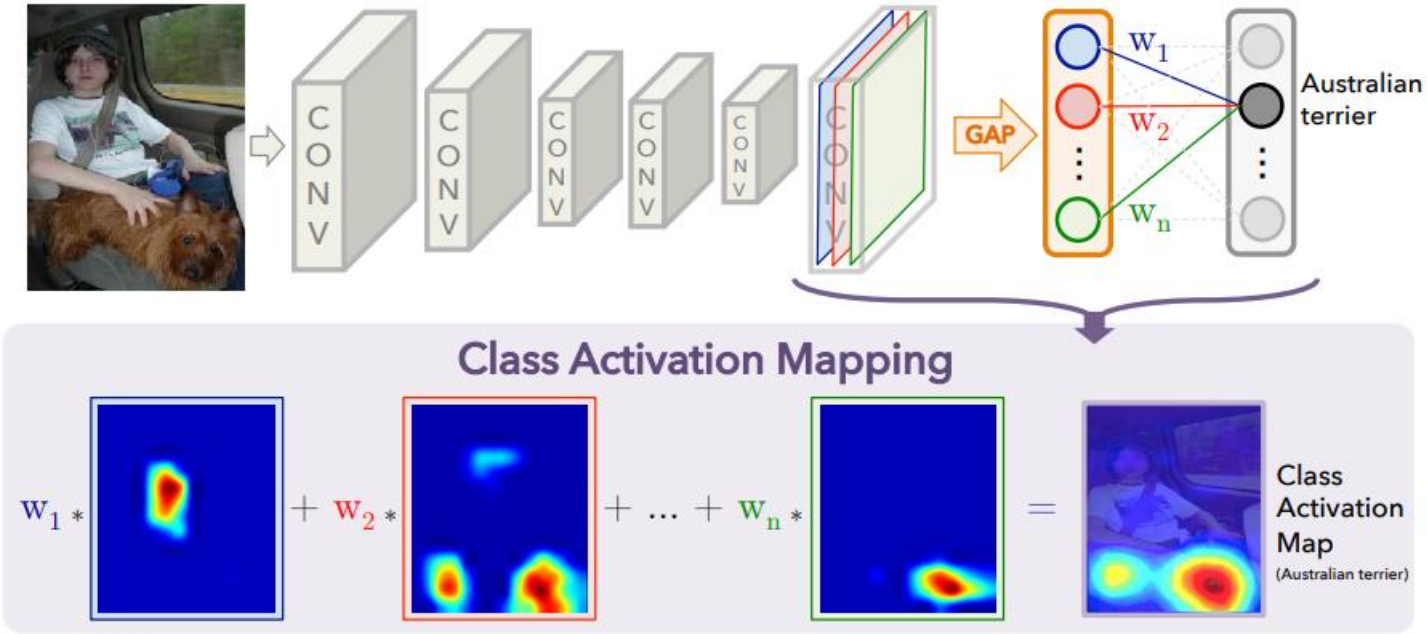


Figure 2. Class Activation Mapping: the predicted class score is mapped back to the previous convolutional layer to generate the class activation maps (CAMs). The CAM highlights the class-specific discriminative regions.

Class Activation Mapping

For a given image, let $f_k(x, y)$ represent the activation of unit k in the last convolutional layer at spatial location (x, y). Then, for unit k, the result of performing global average pooling, F^k is $\sum_{x,y} f_k(x, y)$. Thus, for a given class c, the input to the softmax, S_c, is $\sum_k w_k^c F_k$ where w_k^c is the weight corresponding to class c for unit k. Essentially, w_k^c indicates the *importance* of F_k for class c. Finally the output of the softmax for class c, P_c is given by $\frac{\exp(S_c)}{\sum_c \exp(S_c)}$.

Here we ignore the bias term: we explicitly set the input bias of the softmax to 0 as it has little to no impact on the classification performance.

By plugging $F_k = \sum_{x,y} f_k(x, y)$ into the class score, S_c, we obtain

$$S_c = \sum_k w_k^c \sum_{x,y} f_k(x, y) = \sum_{x,y} \boxed{\sum_k w_k^c f_k(x, y).} \quad (1)$$

We define M_c as the class activation map for class c, where each spatial element is given by

$$\boxed{M_c(x, y) = \sum_k w_k^c f_k(x, y).} \quad (2)$$

Thus, $S_c = \sum_{x,y} M_c(x, y)$, and hence $M_c(x, y)$ directly indicates the importance of the activation at spatial grid (x, y) leading to the classification of an image to class c.

Weakly-supervised Object Localization

- Classification

Table 1. Classification error on the ILSVRC validation set.

Networks	top-1 val. error	top-5 val. error
VGGnet-GAP	33.4	12.2
GoogLeNet-GAP	35.0	13.2
AlexNet*-GAP	44.9	20.9
AlexNet-GAP	51.1	26.3
GoogLeNet	31.9	11.3
VGGnet	31.2	11.4
AlexNet	42.6	19.5
NIN	41.9	19.6
GoogLeNet-GMP	35.6	13.9

- Localization

Table 2. Localization error on the ILSVRC validation set. *Back-prop* refers to using [23] for localization instead of CAM.

Method	top-1 val.error	top-5 val. error
GoogLeNet-GAP	**56.40**	**43.00**
VGGnet-GAP	57.20	45.14
GoogLeNet	60.09	49.34
AlexNet*-GAP	63.75	49.53
AlexNet-GAP	67.19	52.16
NIN	65.47	54.19
Backprop on GoogLeNet	61.31	50.55
Backprop on VGGnet	61.12	51.46
Backprop on AlexNet	65.17	52.64
GoogLeNet-GMP	57.78	45.26

Table 3. Localization error on the ILSVRC test set for various weakly- and fully- supervised methods.

Method	supervision	top-5 test error
GoogLeNet-GAP (heuristics)	weakly	**37.1**
GoogLeNet-GAP	weakly	42.9
Backprop [23]	weakly	46.4
GoogLeNet [25]	full	26.7
OverFeat [22]	full	29.9
AlexNet [25]	full	34.2

Weakly-supervised Object Localization

- Object detection

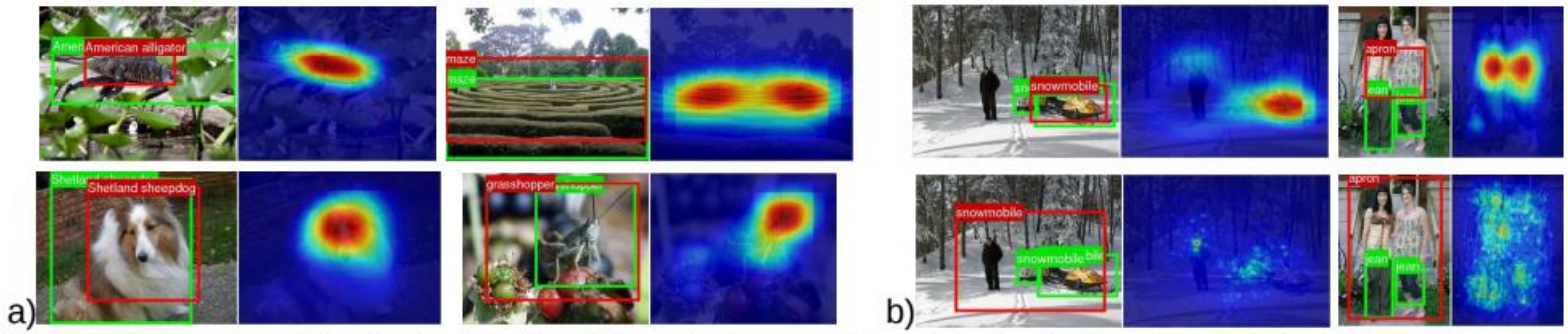


Figure 6. a) Examples of localization from GoogleNet-GAP. b) Comparison of the localization from GooleNet-GAP (upper two) and the backpropagation using AlexNet (lower two). The ground-truth boxes are in green and the predicted bounding boxes from the class activation map are in red.

- Weakly supervised text detector

Figure 11. Learning a weakly supervised text detector. The text is accurately detected on the image even though our network is not trained with text or any bounding box annotations.

Conclusion

- Class Activation Mapping(CAM) with Global Average Pooling(GAP)
- Weakly-supervised object localization → CNNs trained for classification can do object localization
- Various experiments are very important!
- Reducing uncertainty in deep learning (NRF Global ph.d research topic)

Thank you.
Q & A