Class Activation Map
and
Weakly-Supervised Learning

Taeoh Kim

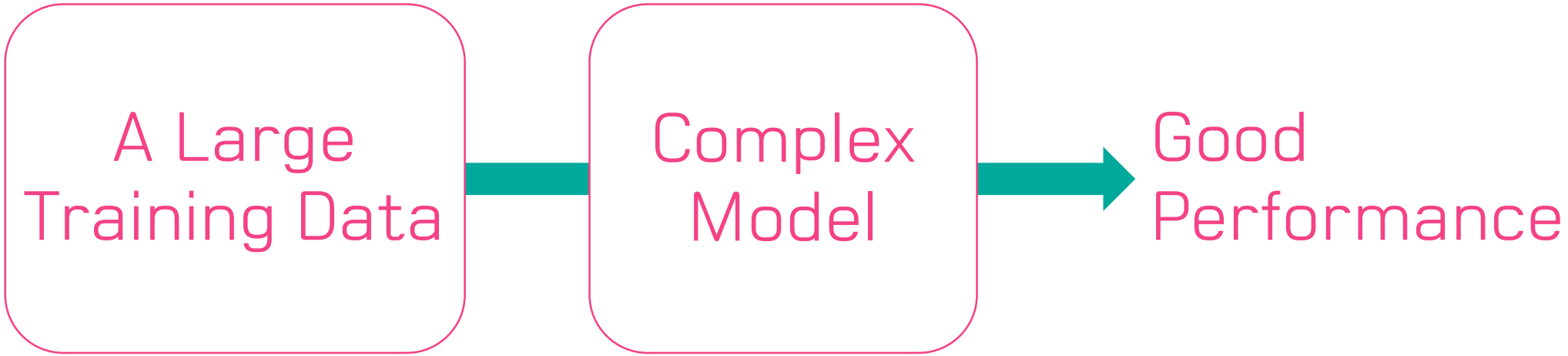
MVP-SEMINAR-18-1
2018. 01. 05

Today

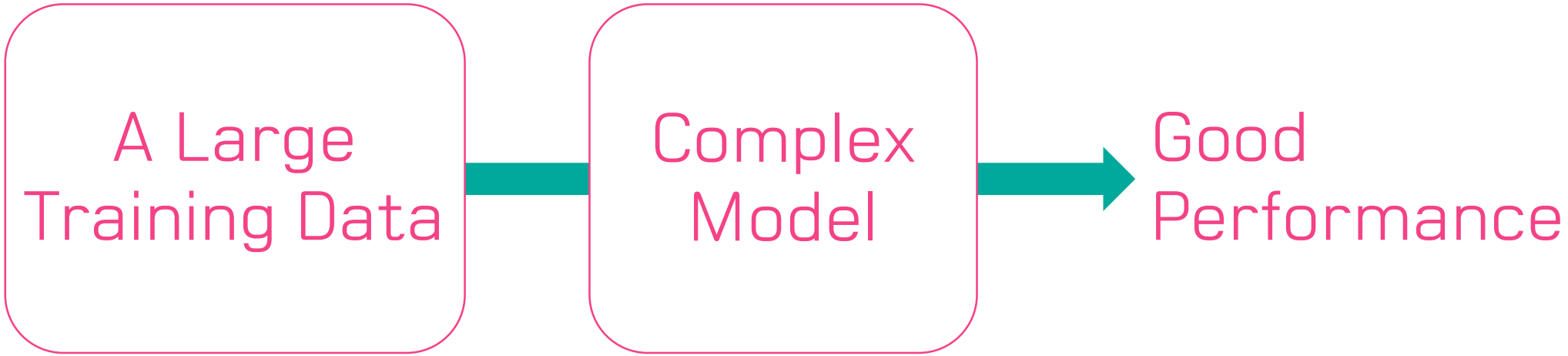
- Deep Learning Interpretation/Visualization
- CAM
- Weakly-Supervised Learning
- Grad-CAM
- YOLO 9000
- MaskX R-CNN

Deep Learning

Deep Learning = Black Box Model?



Deep Learning = Black Box Model?



내부를 분석하기에는
너무 복잡하다

여러 시도들

- Activation: Deconvolution

Visualizing and Understanding Convolutional Networks

- Activation: AllConvNet + Guided BackProp

Striving for Simplicity: The All Convolutional Net

- Model = f (Training)

Understanding Black-box Predictions via Influence Functions

- Explaining Model

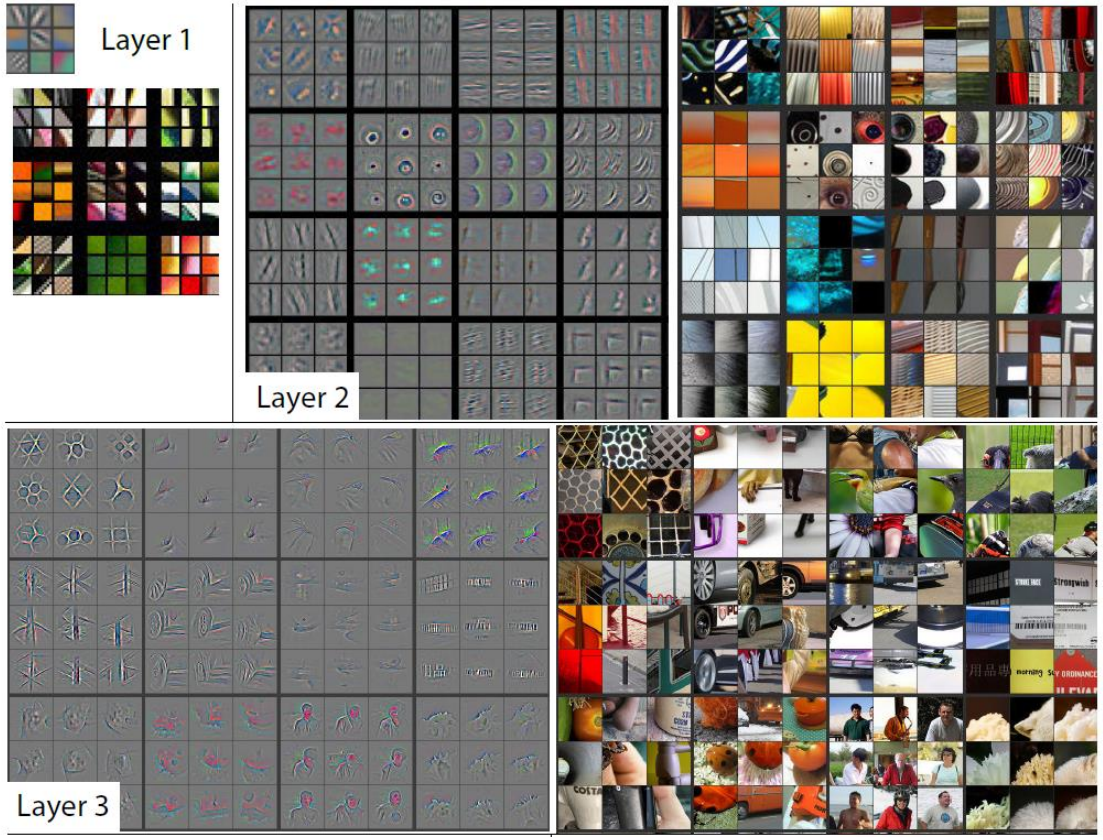
“Why Should I Trust You?” Explaining the Predictions of Any Classifier

- Activation: CAM (Class-Specific) Today

여러 시도들

- Activation: Deconvolution

Visualizing and Understanding Convolutional Networks



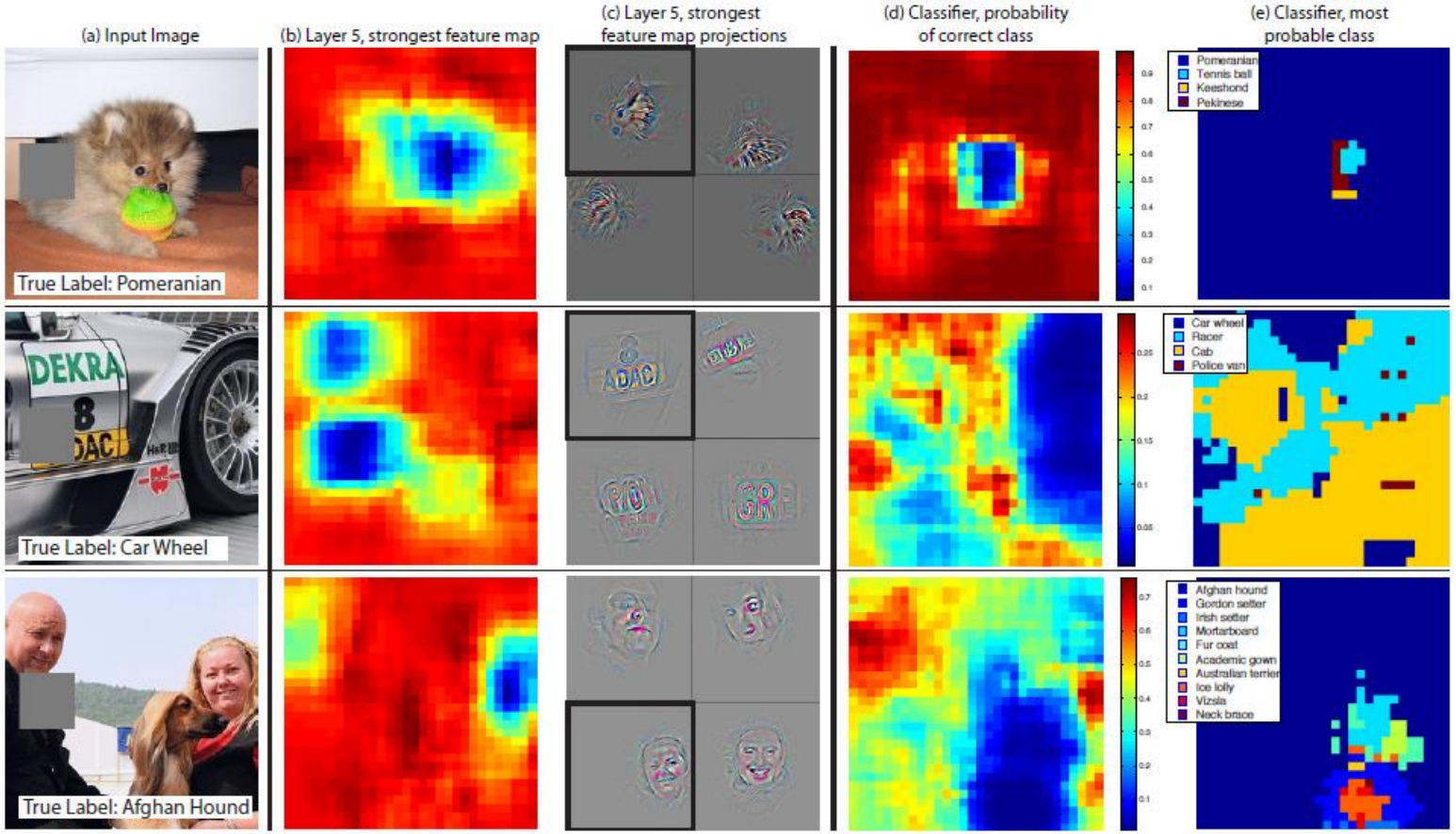
Deconvolution:

특정 CNN Layer에서 Input Space까지
UnPool, UnReLU, Deconvolution
하여 Visualization
(학습 x)

여러 시도들

- Activation: Deconvolution

Visualizing and Understanding Convolutional Networks



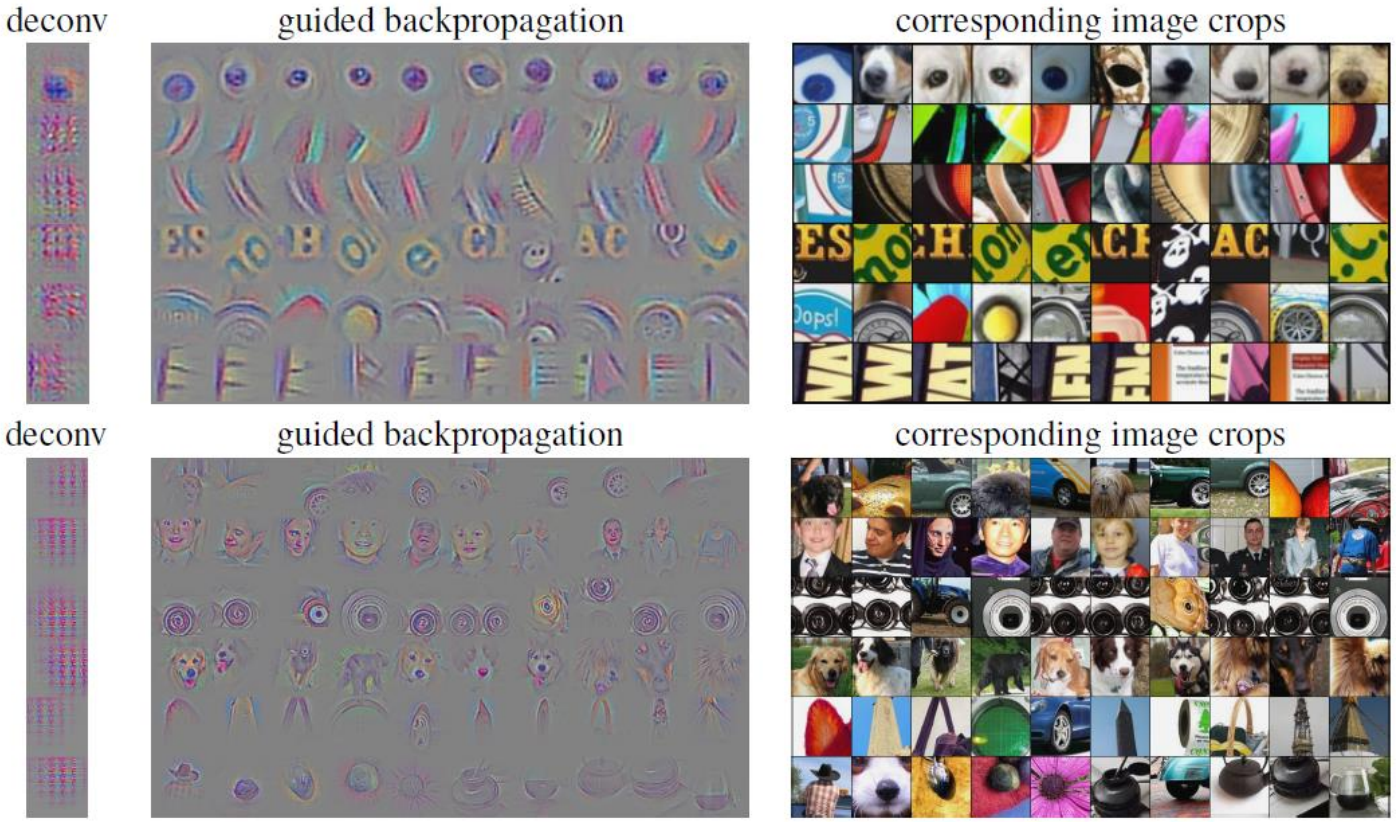
Occlusion Sensitivity:

특정 부분을 가렸을 경우
Class Softmax 값에 미치는 영향

여러 시도들

- Activation: AllConvNet + Guided BackProp

Striving for Simplicity: The All Convolutional Net



All-Conv Net을 사용
Max-pooling을 Conv로 대체

Guided BackProp.:
Unpooling + BackProp.를 제안

여러 시도들

- Model = f (Training)

Understanding Black-box Predictions via Influence Functions

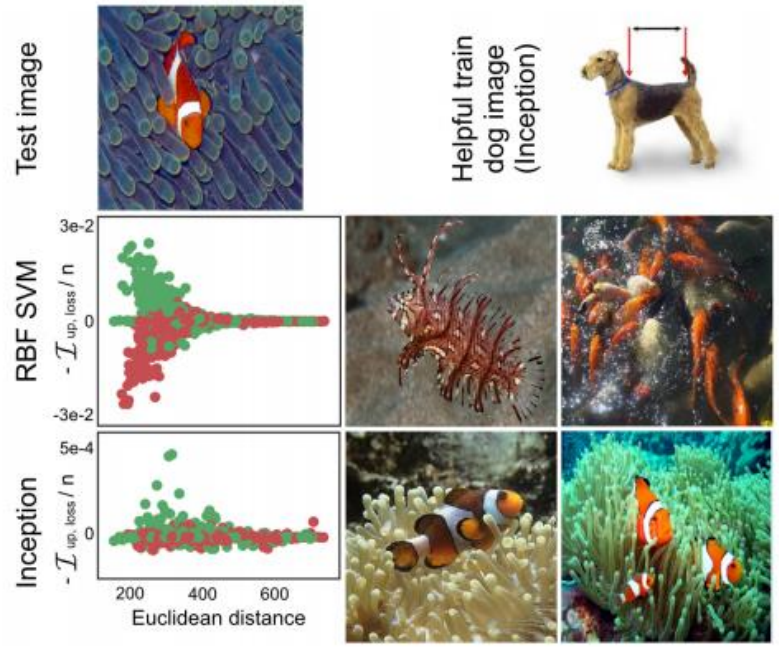


Figure 4. **Inception vs. RBF SVM. Bottom left:** $-\mathcal{I}_{\text{up,loss}}(z, z_{\text{test}})$ vs. $\|z - z_{\text{test}}\|_2^2$. Green dots are fish and red dots are dogs. **Bottom right:** The two most helpful training images, for each model, on the test. **Top right:** An image of a dog in the training set that helped the Inception model correctly classify the test image as a fish.

특정 Training Sample의
영향력을 분석 가능

여러 시도들

- Explaining Model

"Why Should I Trust You?" Explaining the Predictions of Any Classifier

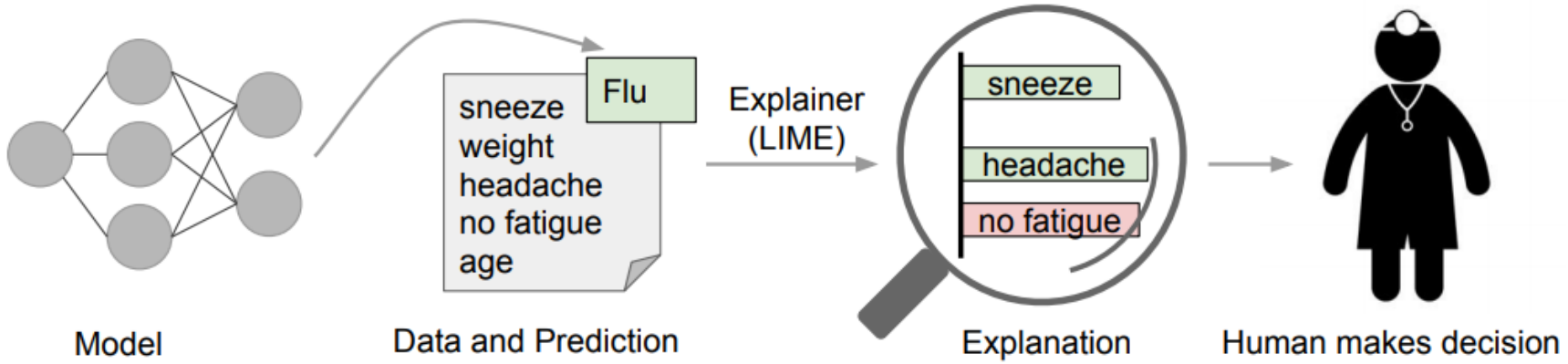
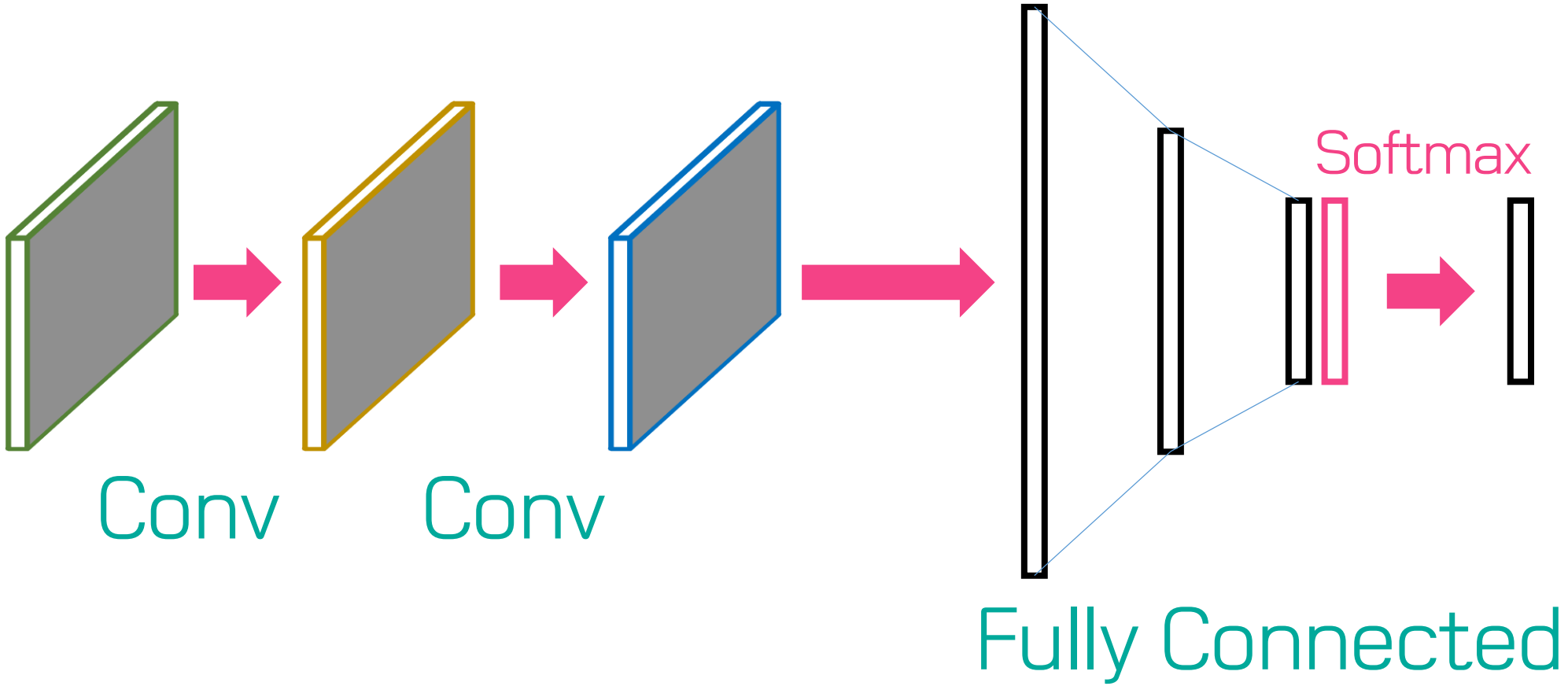


Figure 1: Explaining individual predictions. A model predicts that a patient has the flu, and LIME highlights the symptoms in the patient's history that led to the prediction. Sneeze and headache are portrayed as contributing to the "flu" prediction, while "no fatigue" is evidence against it. With these, a doctor can make an informed decision about whether to trust the model's prediction.

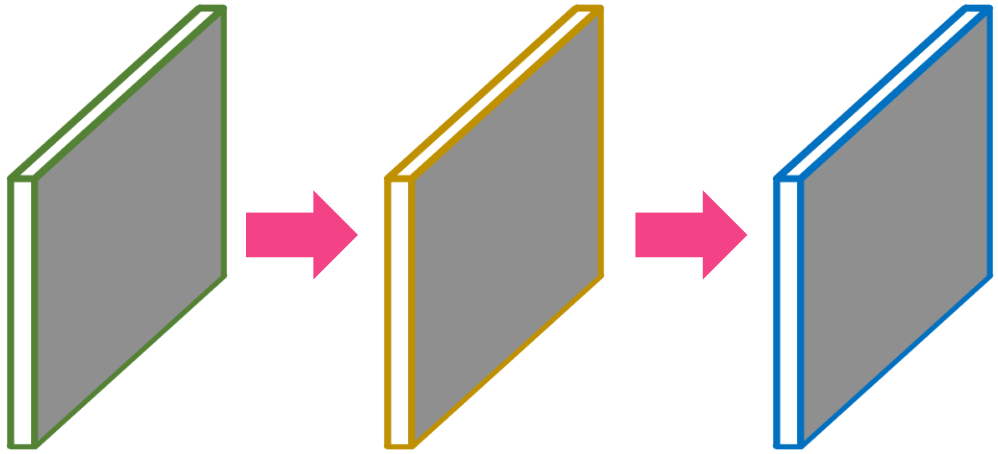
Class Activation Map

Learning Deep Features for Discriminative Localization, CVPR 2016

CNN + FC Network



CNN + FC Network



Conv Conv

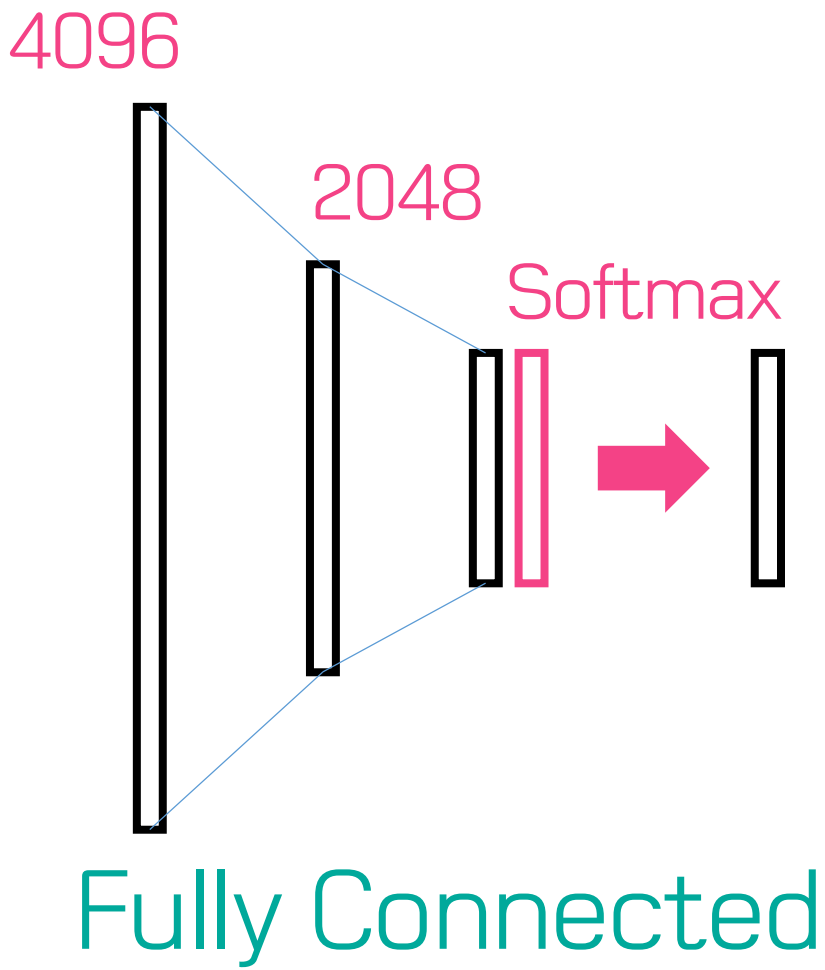
3x3 Conv
512 Channel

3x3x512 = 4608 Parameters

CNN + FC Network

4096x2048

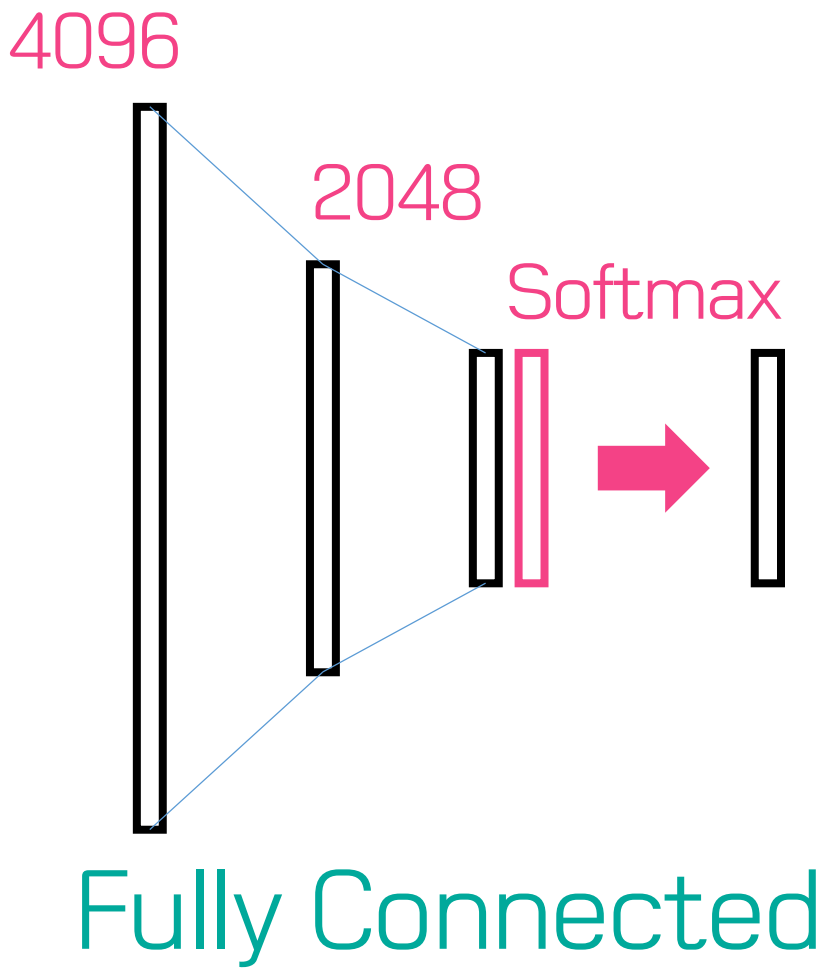
8388608 Parameters



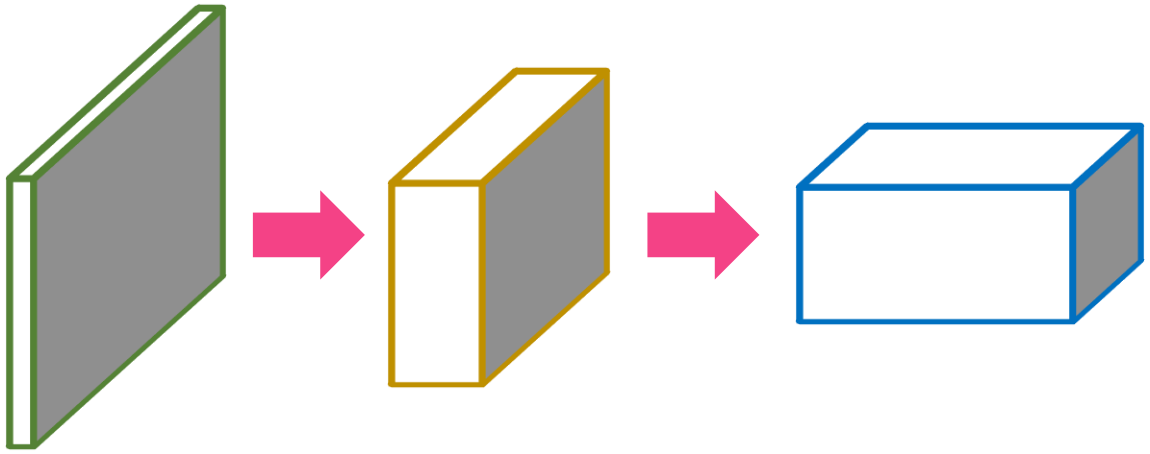
Fully Connected

CNN + FC Network

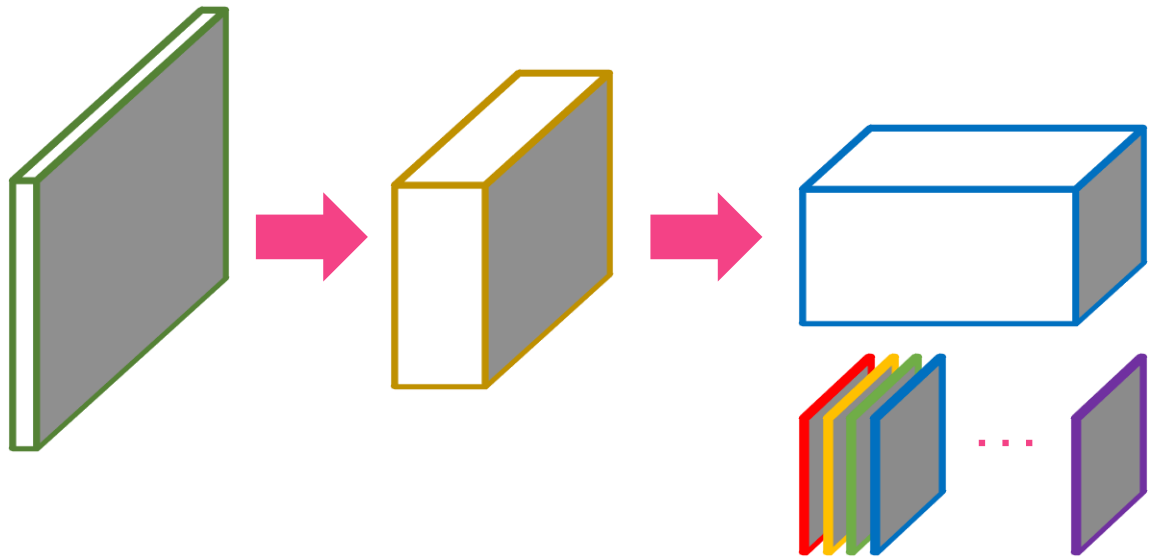
위치 정보도 분실
많은 Parameter → Overfitting
고정된 Size만 사용 가능



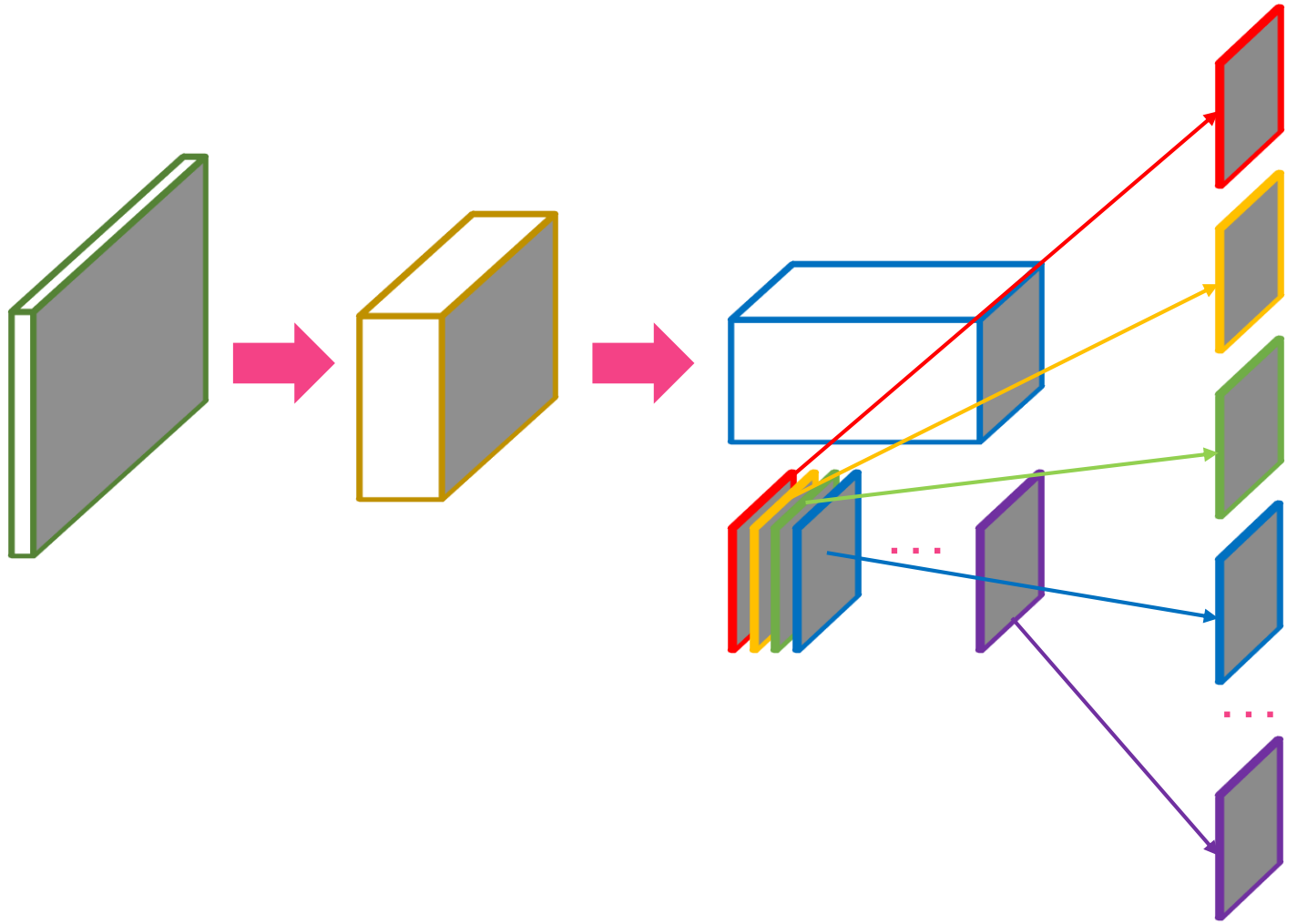
GAP: Global Average Pooling



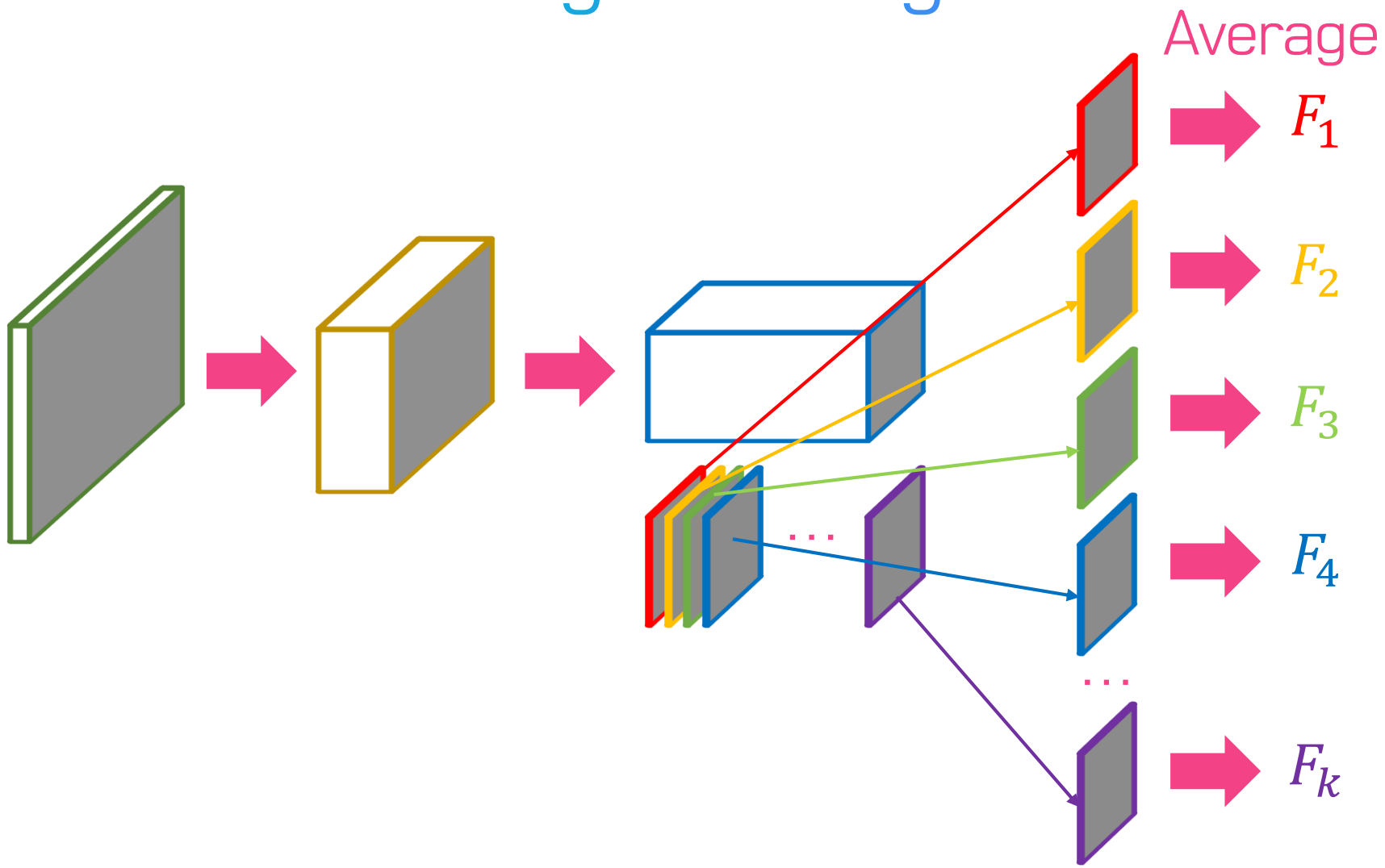
GAP: Global Average Pooling



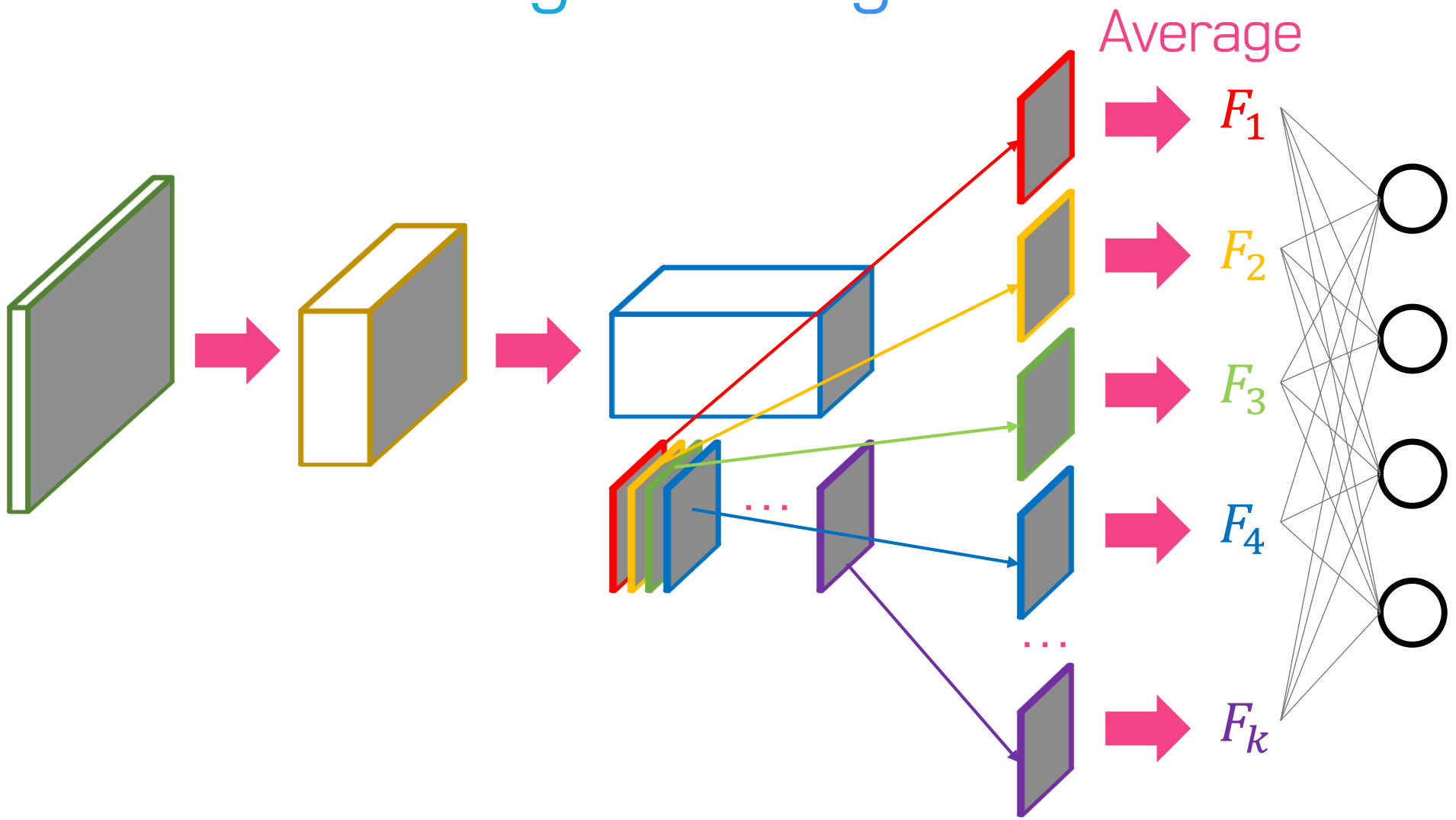
GAP: Global Average Pooling



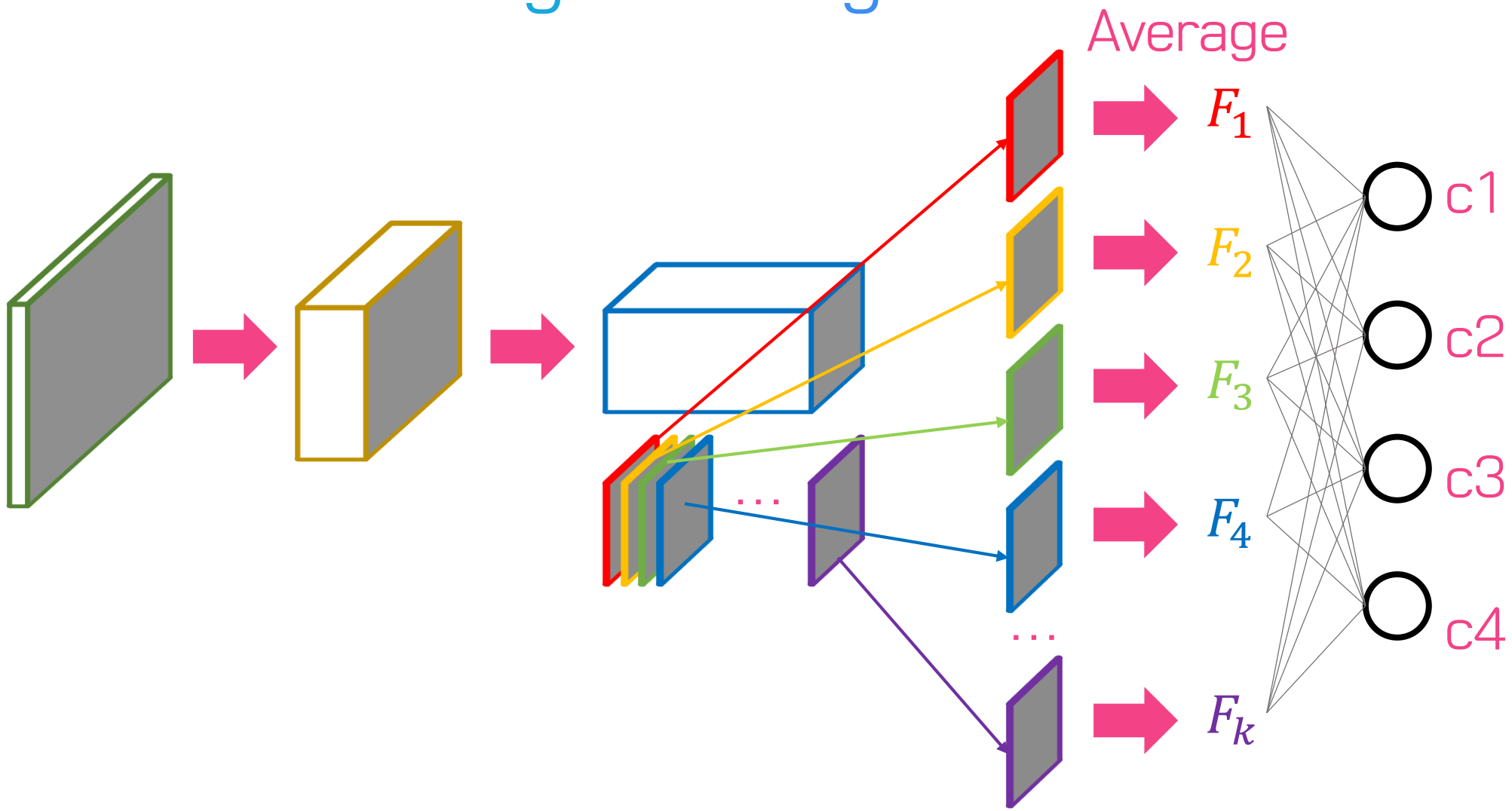
GAP: Global Average Pooling



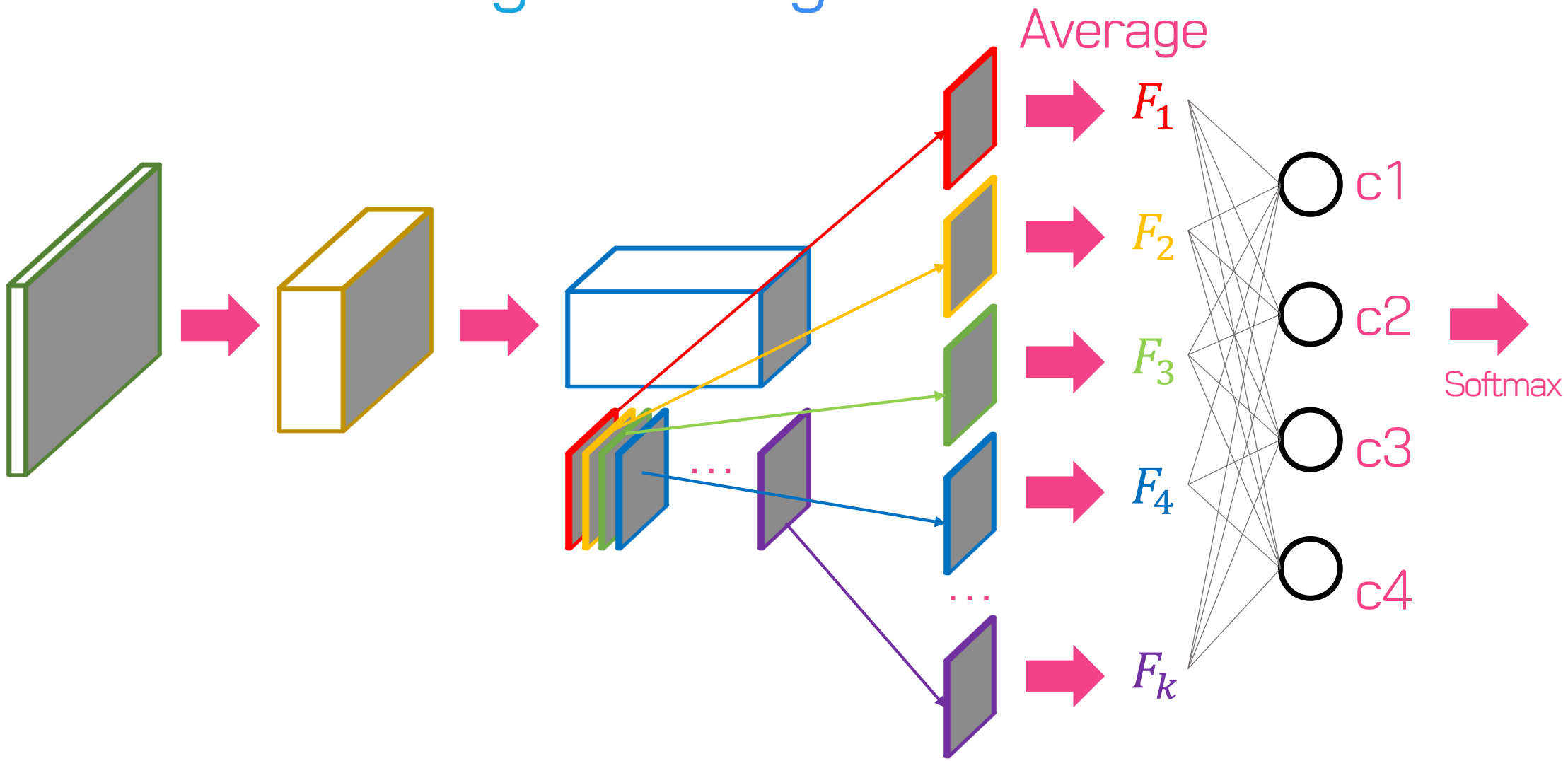
GAP: Global Average Pooling



GAP: Global Average Pooling



GAP: Global Average Pooling



GAP: Global Average Pooling

왜 했나

- Fully Connected Layer를 제거하기 위해서
- 뒤에 나올 CAM을 위해서

GAP: Global Average Pooling

왜 했나

- Fully Connected Layer를 제거하기 위해서
- 뒤에 나올 CAM을 위해서
- 일단, 성능에는 문제가 없나?

Replace FC Layer → GAP

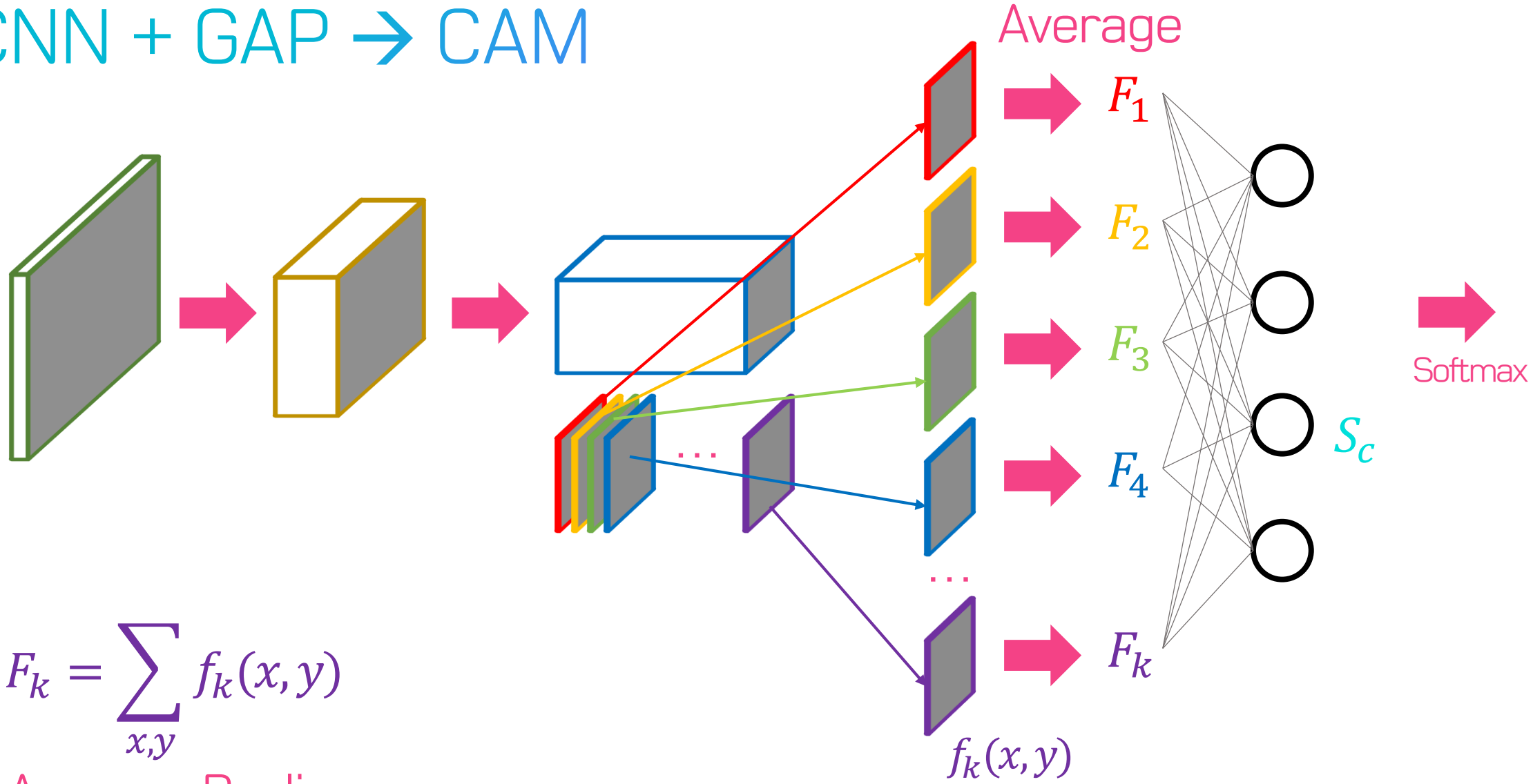
Table 1. Classification error on the ILSVRC validation set.

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

FC Layer 제거하고
GAP으로 대체했더니

성능에 큰 문제는 X

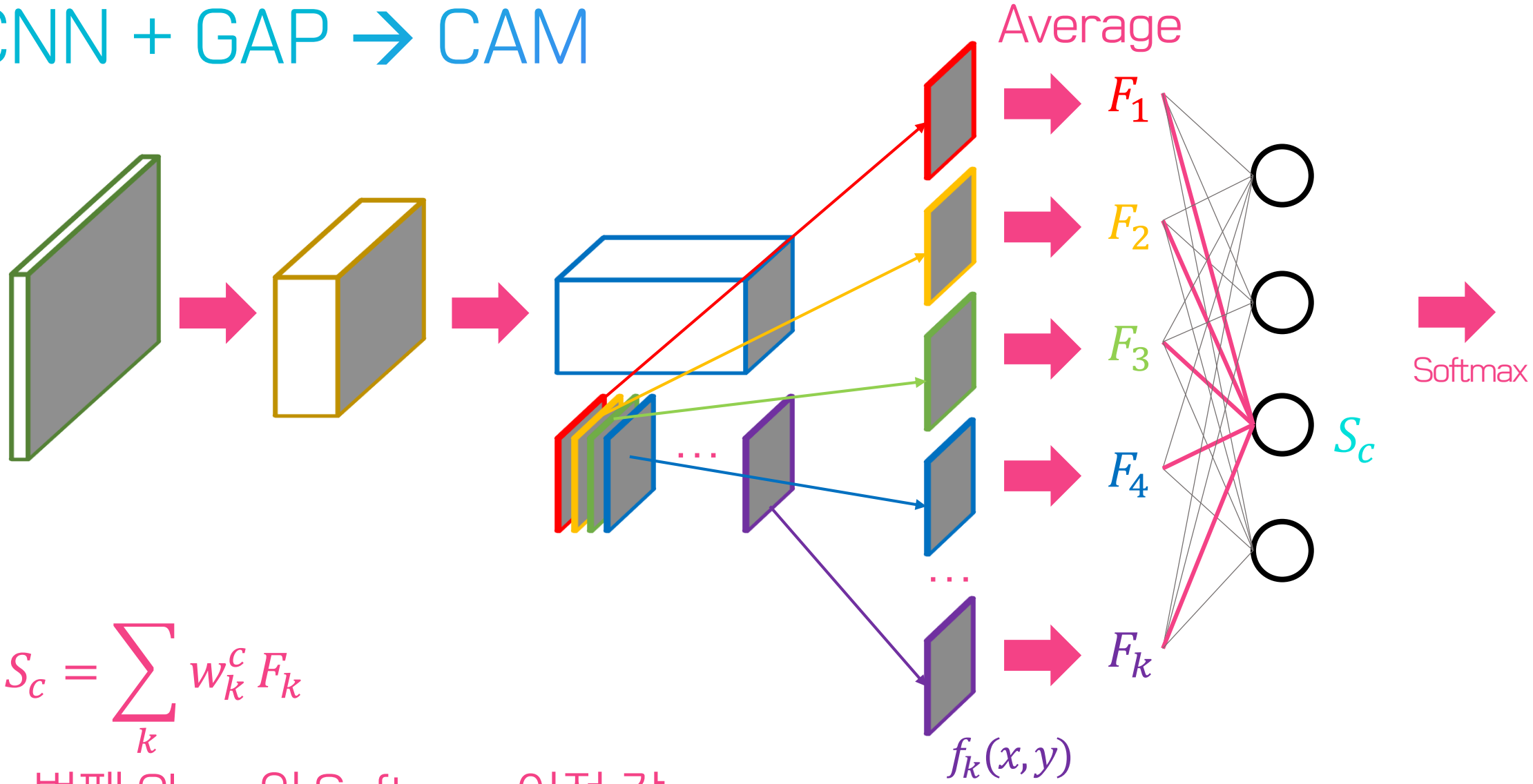
CNN + GAP → CAM



$$F_k = \sum_{x,y} f_k(x,y)$$

Average Pooling

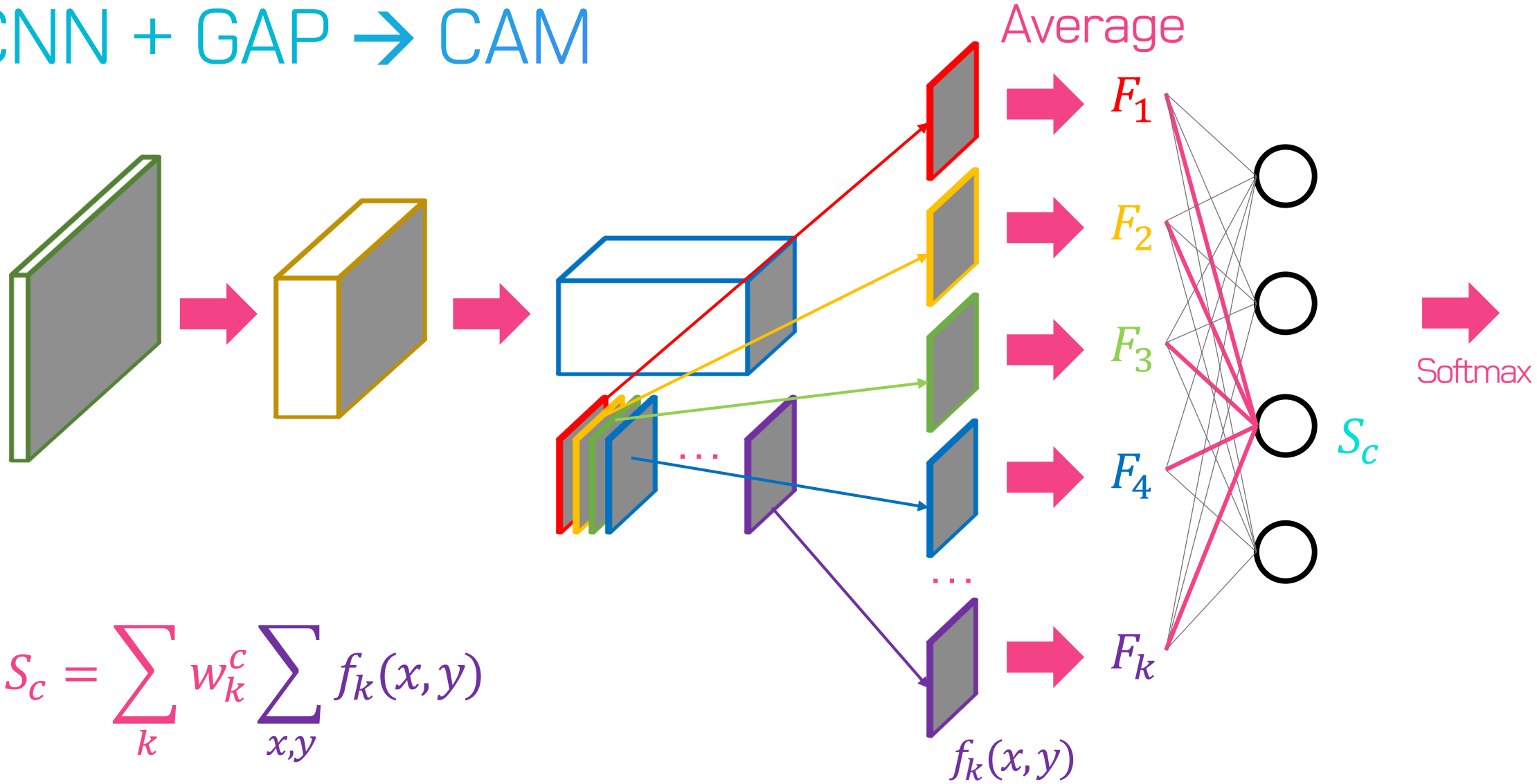
CNN + GAP → CAM



$$S_c = \sum_k w_k^c F_k$$

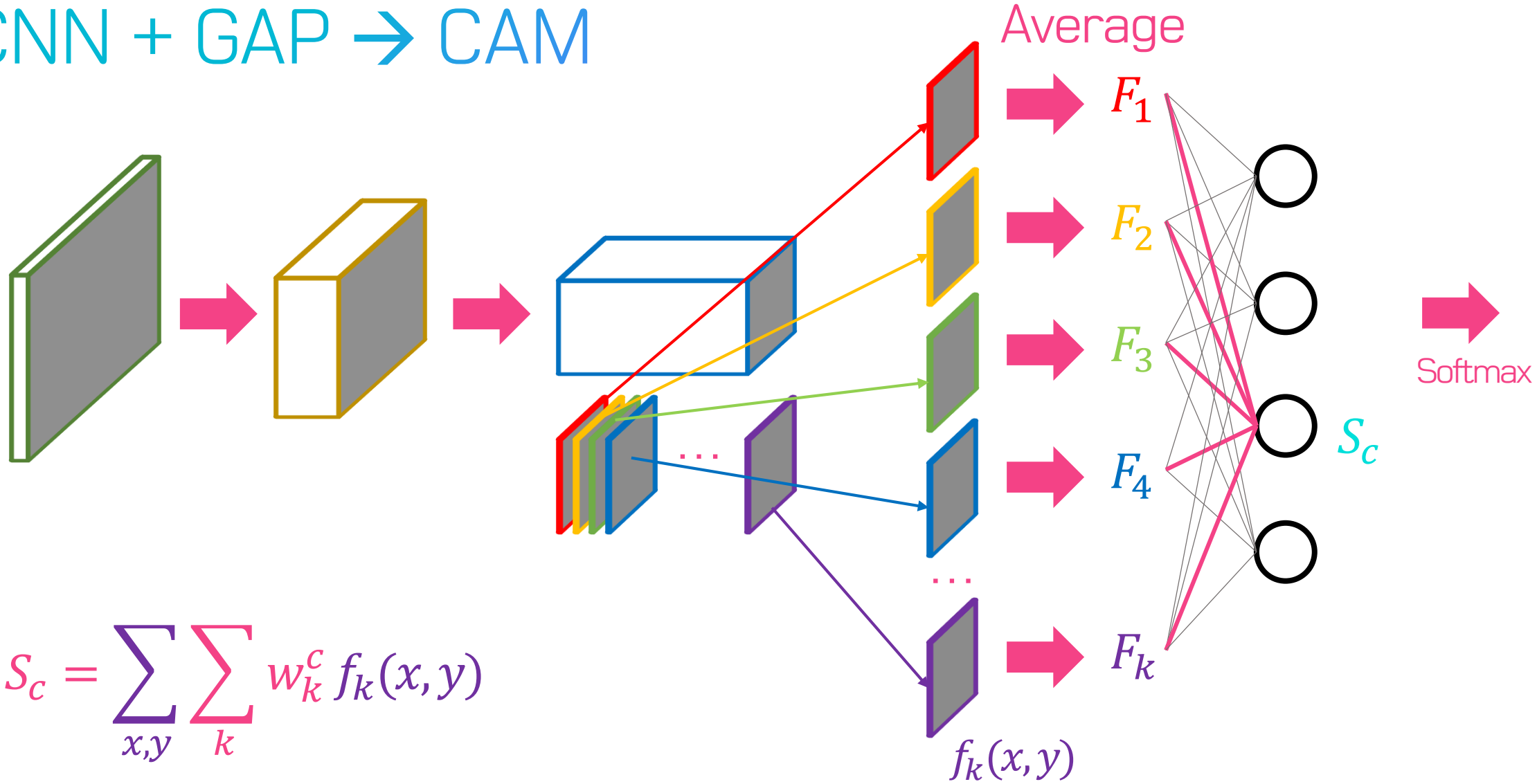
c번째 Class의 Softmax 이전 값

CNN + GAP → CAM



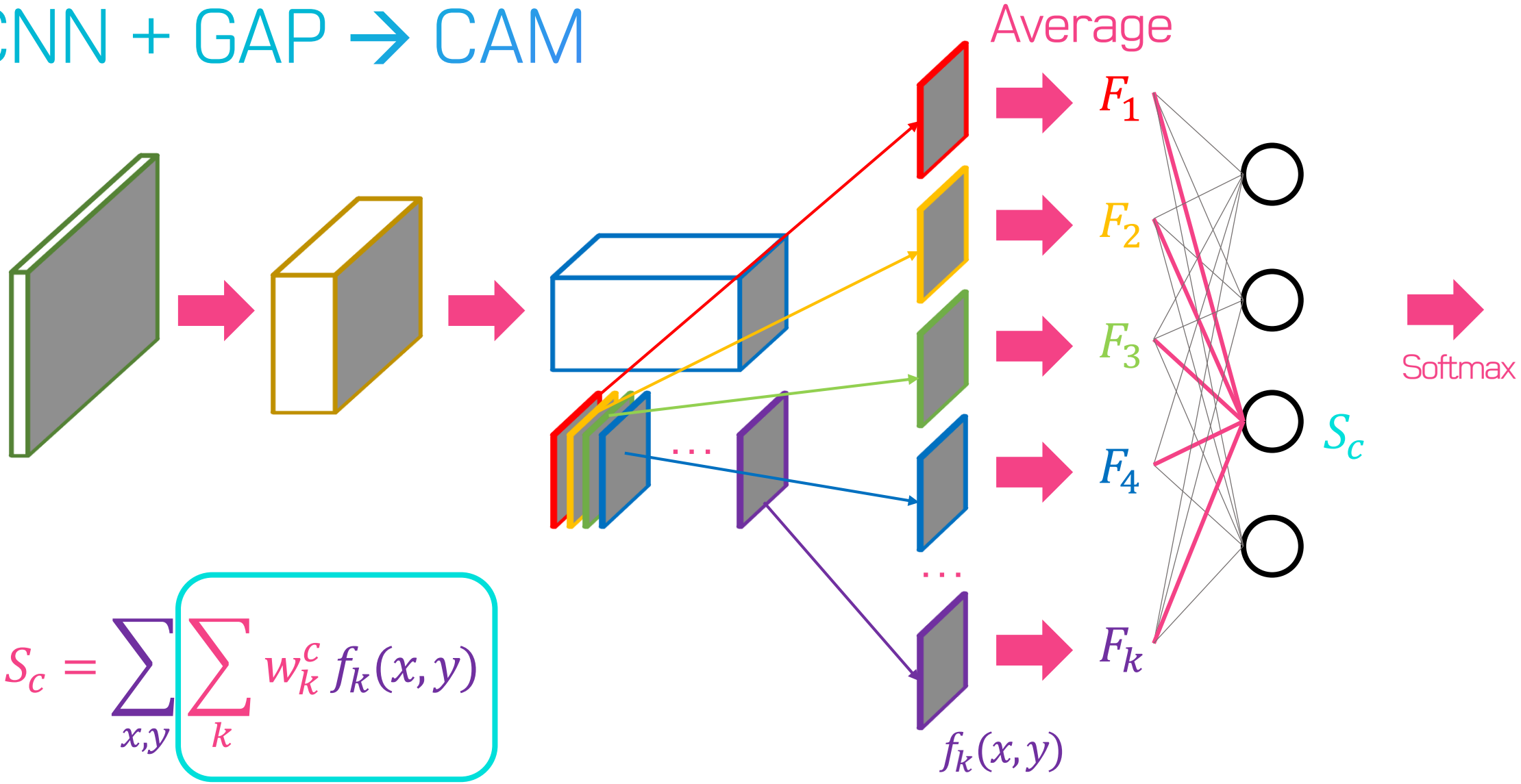
$$S_c = \sum_k w_k^c \sum_{x,y} f_k(x, y)$$

CNN + GAP → CAM



$$S_c = \sum_{x,y} \sum_{k} w_k^c \, f_k(x,y)$$

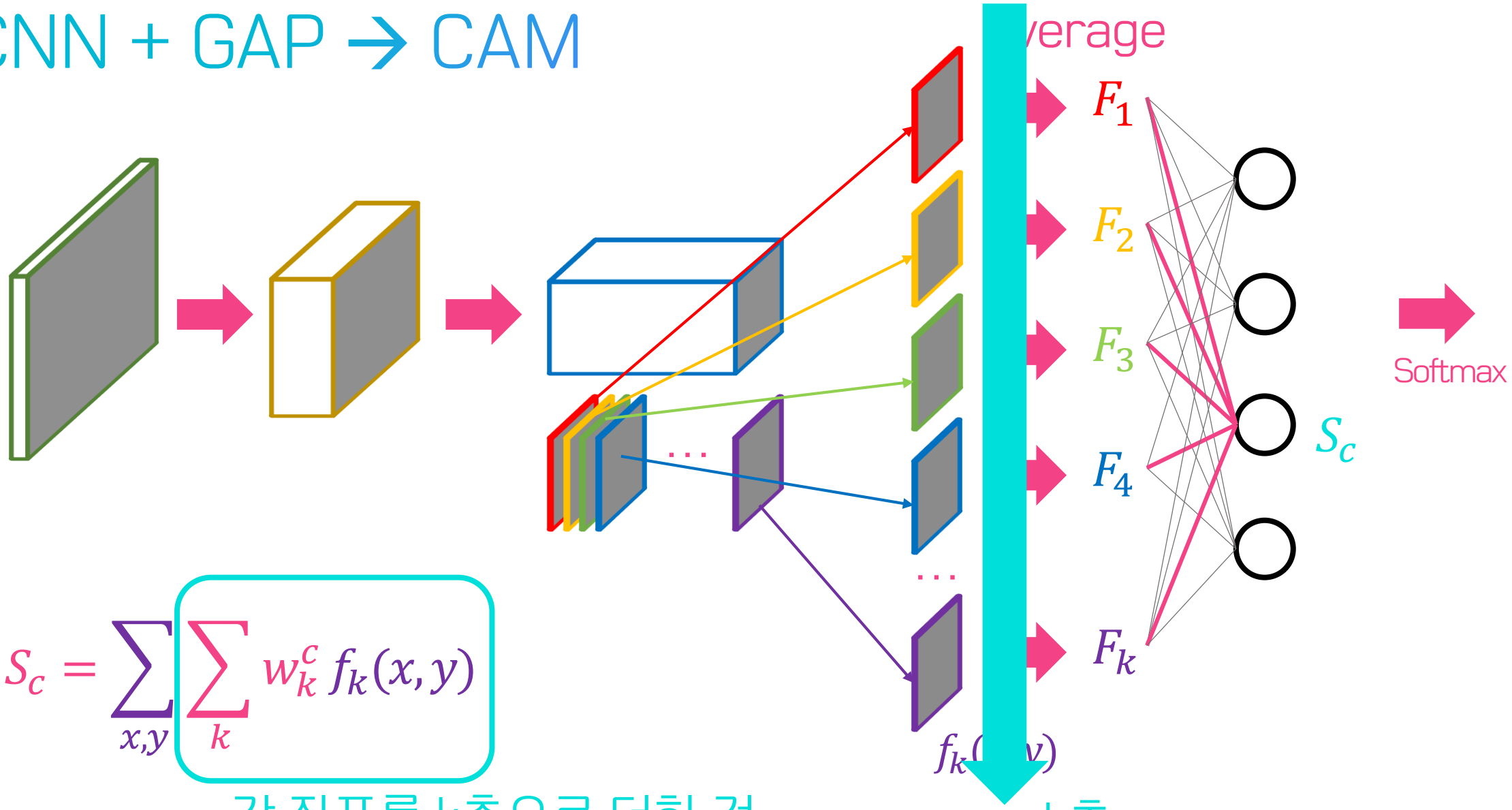
CNN + GAP → CAM



$$S_c = \sum_{x,y} \sum_{k} w_k^c \, f_k(x,y)$$

각 좌표를 k축으로 더한 것

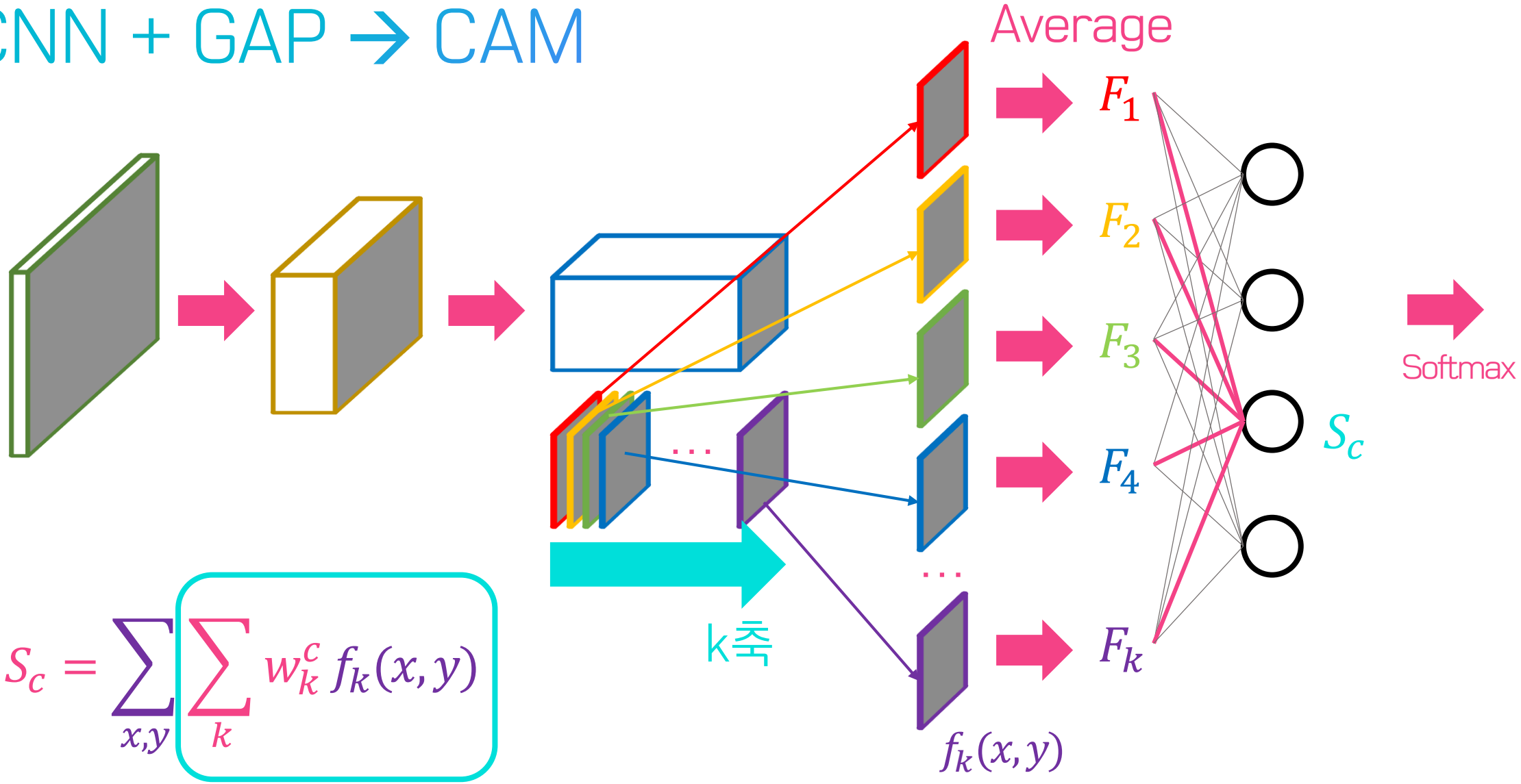
CNN + GAP → CAM



$$S_c = \sum_{x,y} \sum_{k} w_k^c \, f_k(x,y)$$

각 좌표를 k축으로 더한 것

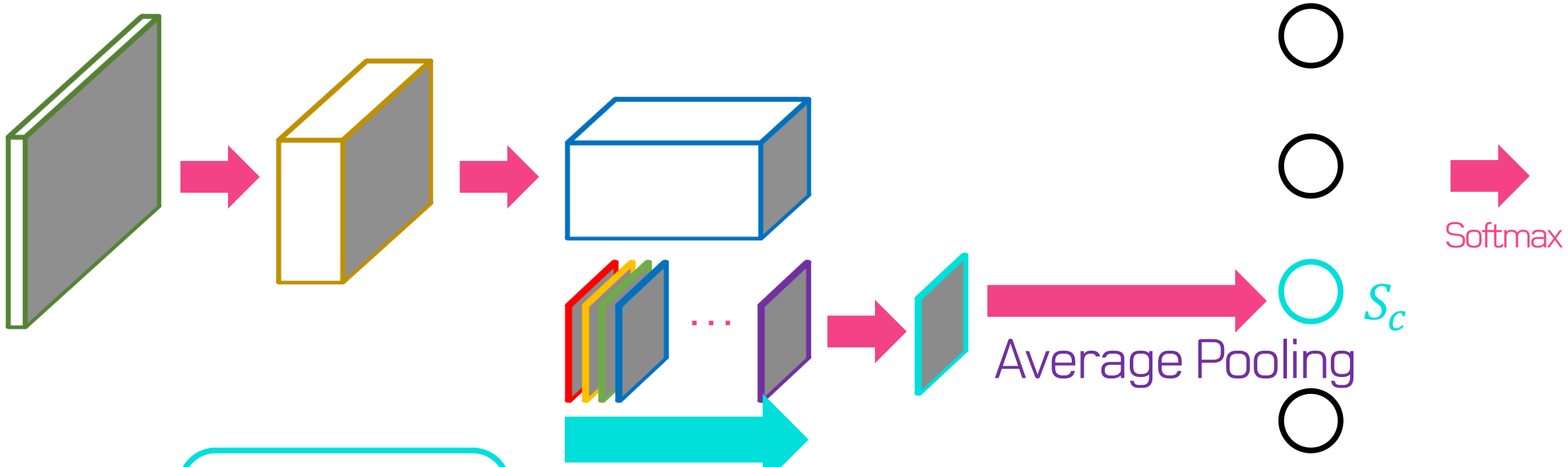
CNN + GAP → CAM



$$S_c = \sum_{x,y} \sum_{k} w_k^c \, f_k(x,y)$$

각 좌표를 k축으로 더한 것

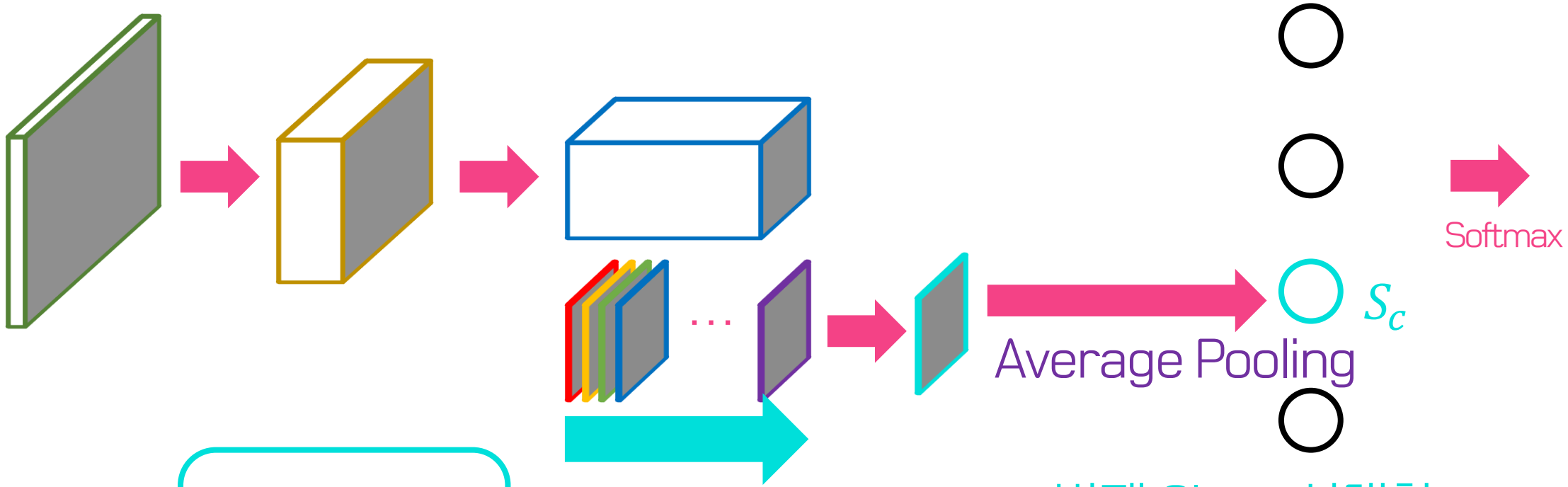
CNN + GAP → CAM



$$S_c = \sum_{x,y} \left[\sum_{k} w_k^c f_k(x,y) \right]$$

각 좌표를 k축으로 더한 것

CNN + GAP → CAM



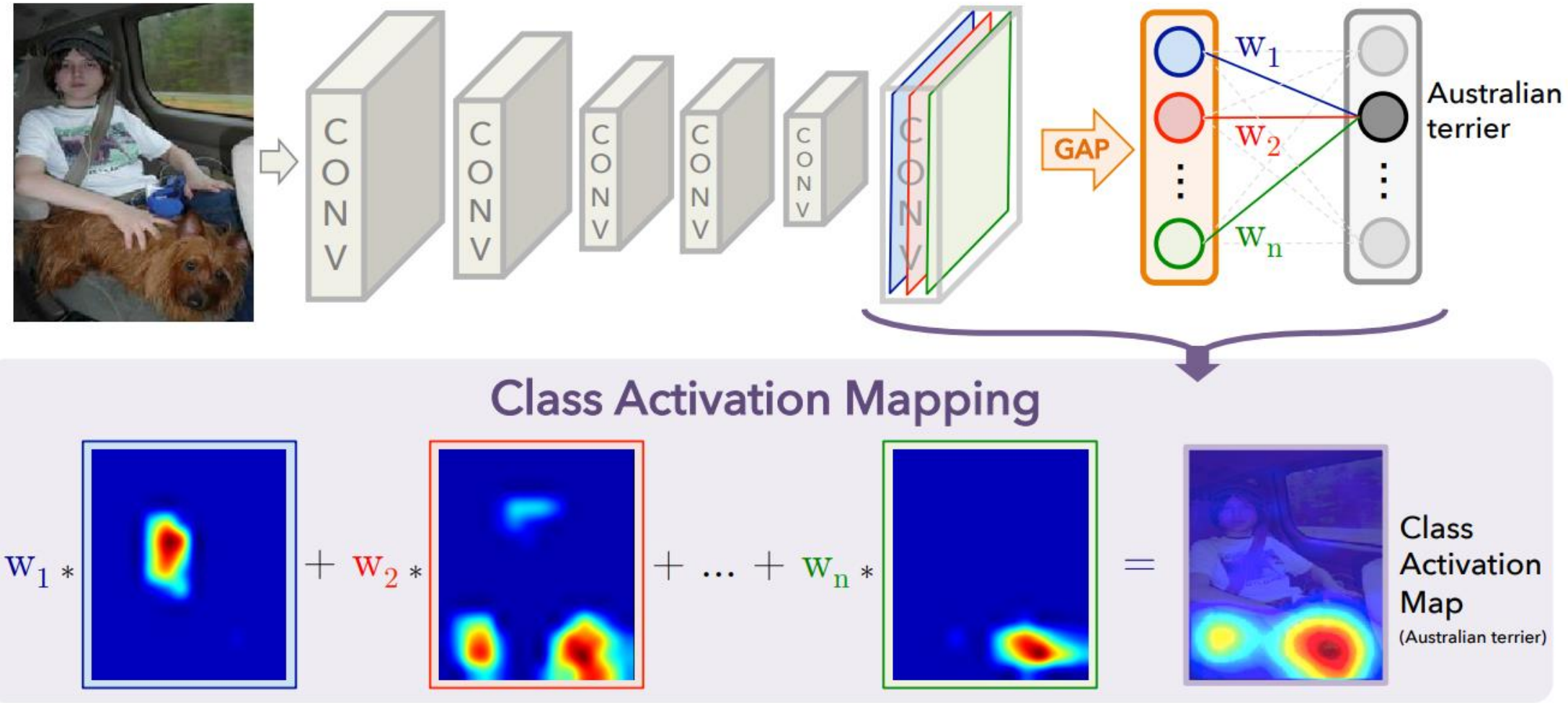
$$S_c = \sum_{x,y} \sum_{k} w_k^c f_k(x,y)$$

각 좌표를 k축으로 더한 것

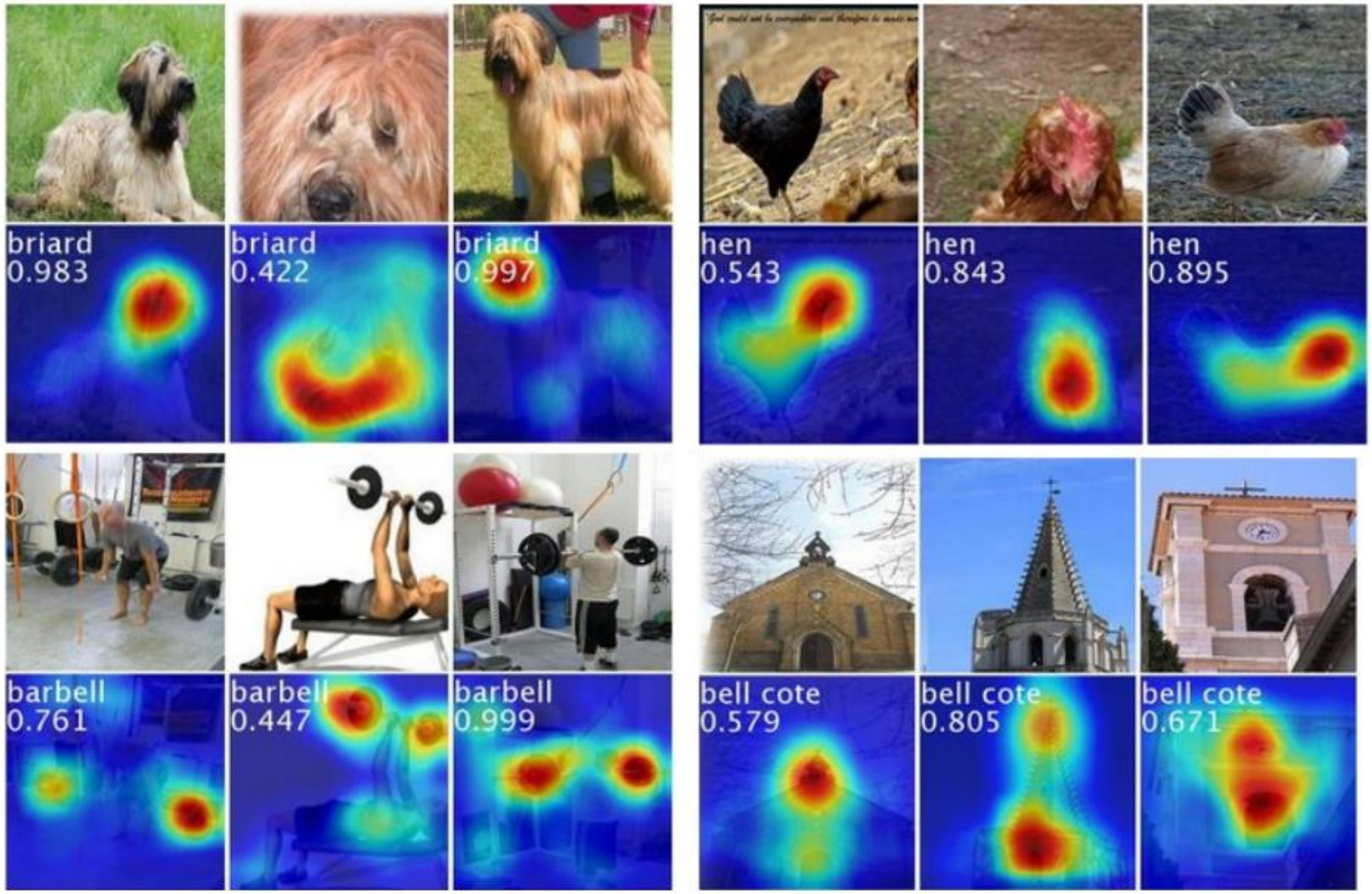
c번째 Class 선택한
Activation Map
= Class Activation Map

CNN + GAP → CAM

Weighted Sum of Feature Map For Each Class

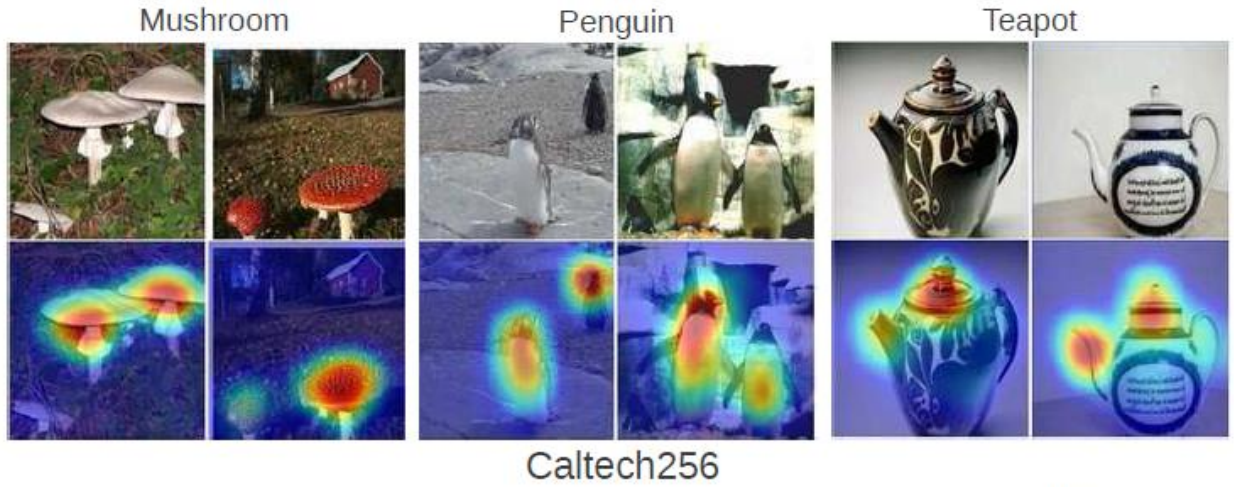


CAM Results

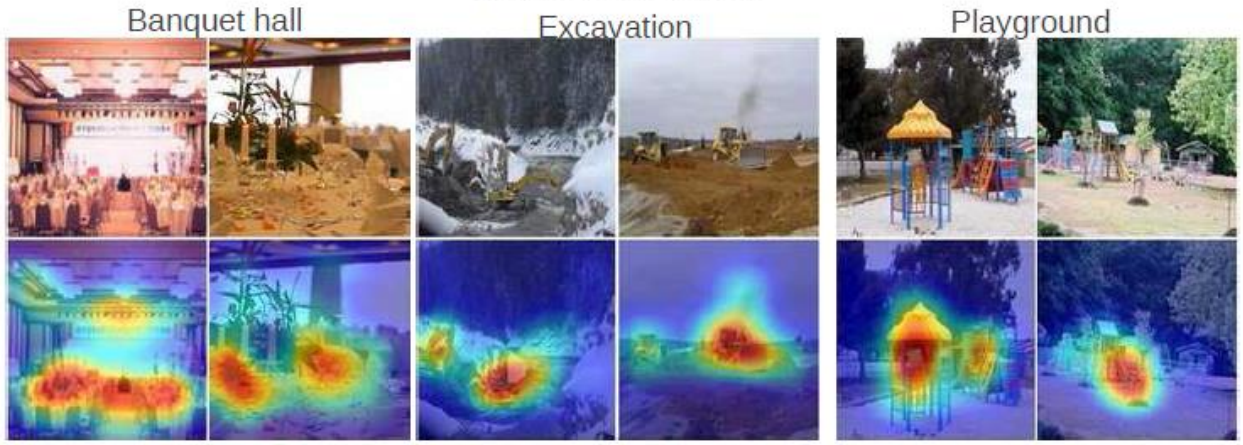

briard
0.983
briard
0.422
briard
0.997
hen
0.543
hen
0.843
hen
0.895
barbell
0.761
barbell
0.447
barbell
0.999
bell cote
0.579
bell cote
0.805
bell cote
0.671

CAM Results

Stanford Action40



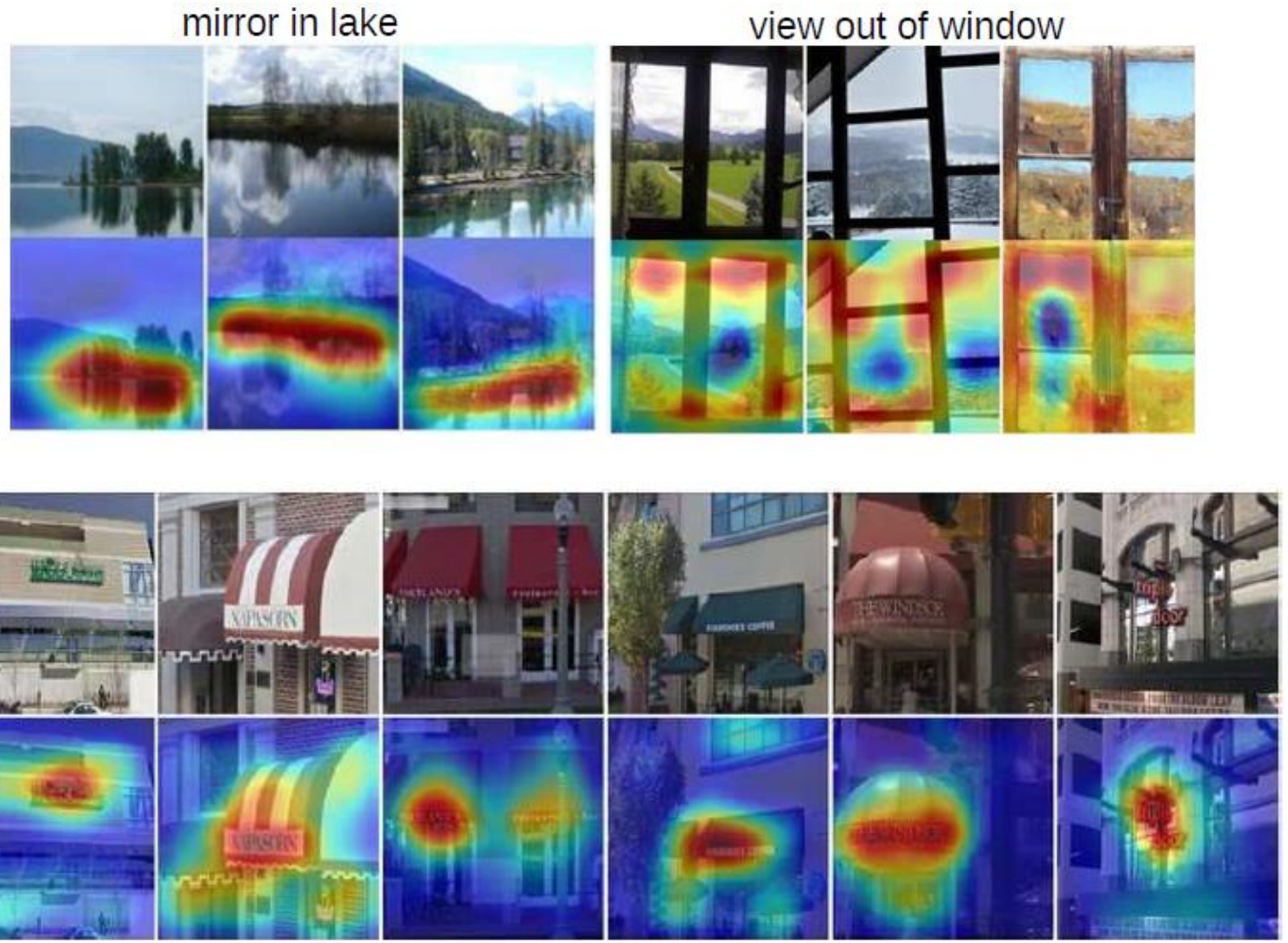
Caltech256



SUN397

UIUC Event8

CAM Results



Weakly Supervised Learning

- Weakly Supervised Learning
  - 쉬운 DB로 학습 (ex. Image Classification)
  - 어려운 문제에 적용 (ex. Object Detection)
  - Class Activation Map, Grad-CAM

- Partially Supervised Learning
  - 많은 쉬운 DB + 적은 어려운 DB로 학습
  - YOLO 9000, MaskX R-CNN

- Semi-Supervised Learning
  - 라벨이 있는 DB와 없는 DB를 이용해서 학습

CAM Can be Object Detection!

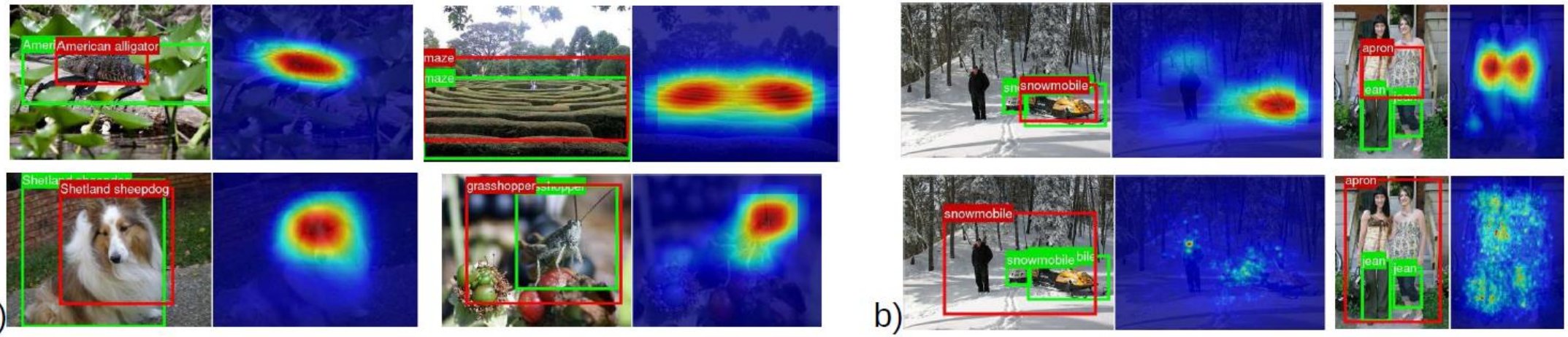
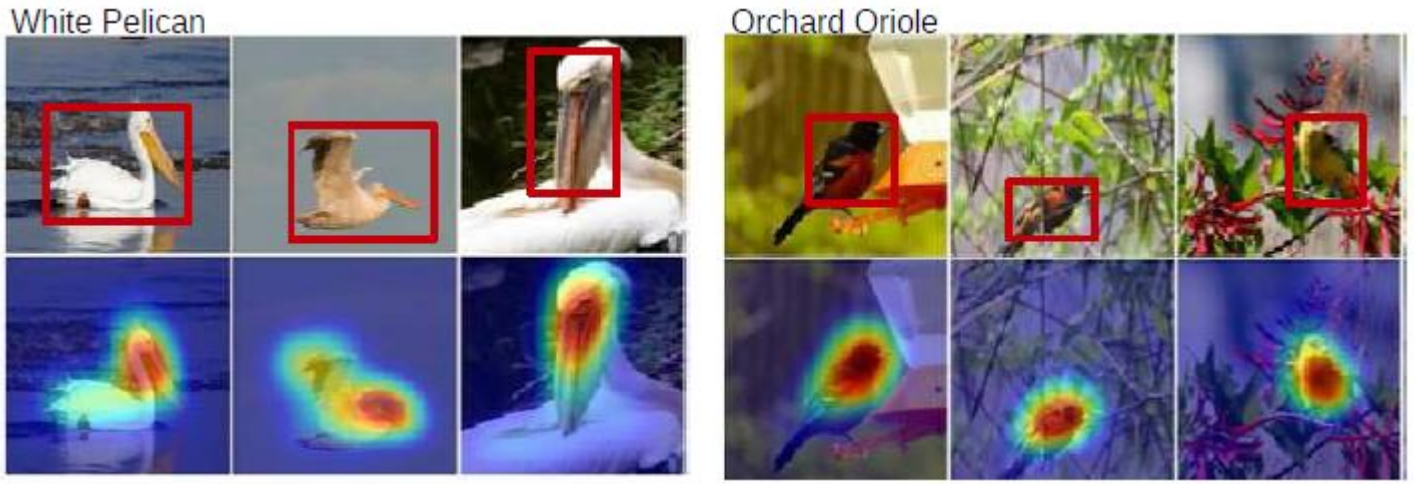


Figure 6. a) Examples of localization from GoogleNet-GAP. b) Comparison of the localization from GooleNet-GAP (upper two) and the backpropagation using AlexNet (lower two). The ground-truth boxes are in green and the predicted bounding boxes from the class activation map are in red.



Maximum Activation
에서 Threshold를 잡아서
Object Detection 가능

Classicifation도 당연히 가능

CAM as Object Detection

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

Grad-CAM

Grad-CAM: Visual Explanations from Deep Networks
via Gradient-based Localization, ICCV 2017

Recall: CAM

CAM의 단점

- 마지막 Conv Layer에서 GAP이 필요하다
- 일반화가 부족!
- 어떻게 일반화해서 모든 Network, 모든 Conv-Layer에서 CAM을 쓸 수 있을까? → Grad-CAM

Recall: CAM

Feature Map

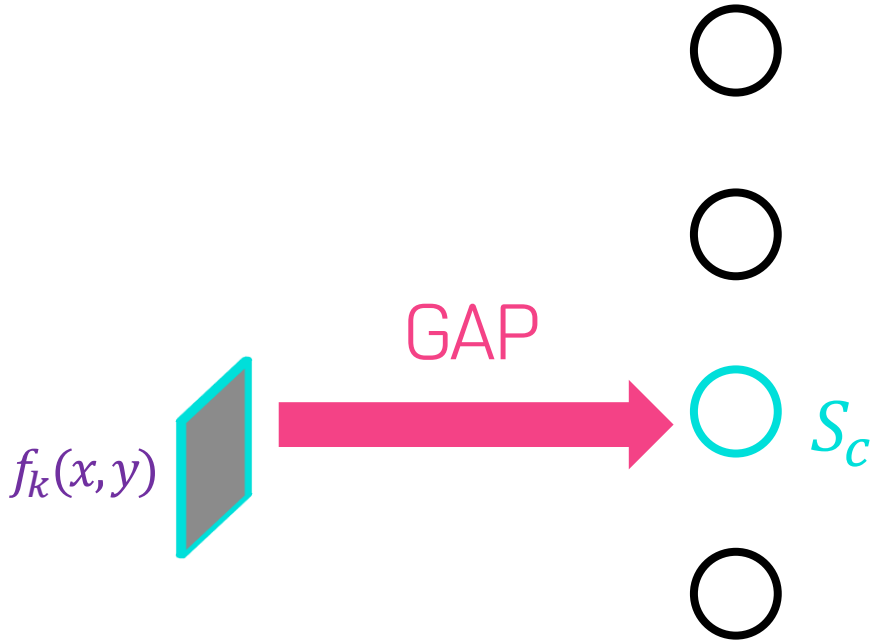
$$S_c = \sum_k w_k^c F_k$$

Recall: CAM

Feature Map

$$S_c = \sum_k w_k^c F_k$$

$$F_k = \sum_{x,y} f_k(x,y)$$



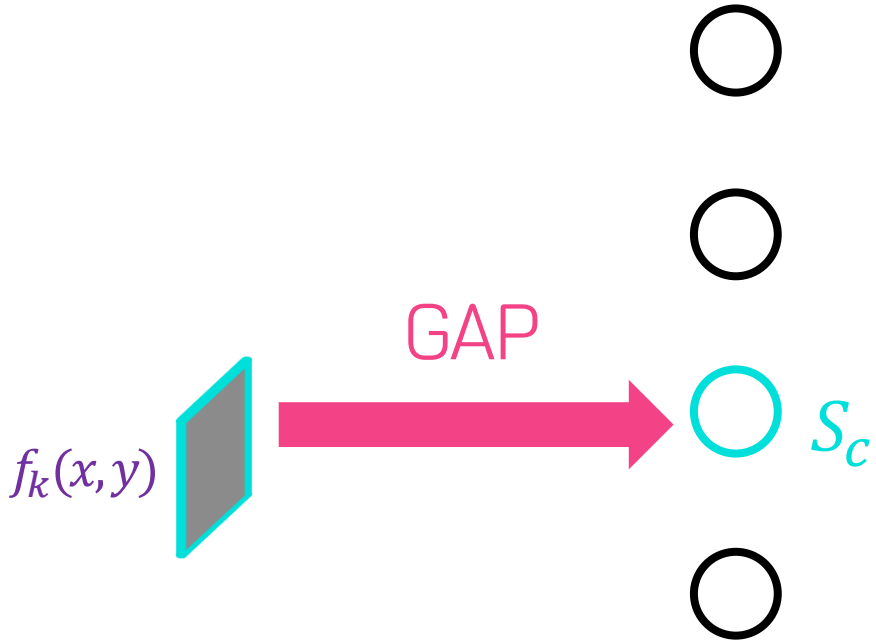
Recall: CAM

Feature Map

$$S_c = \sum_k w_k^c F_k$$

$$F_k = \sum_{x,y} f_k(x,y)$$

$$\frac{\partial S_c}{\partial F_k} = \frac{\frac{\partial S_c}{\partial f_k(x,y)}}{\frac{\partial F_k}{\partial f_k(x,y)}} = w_k^c$$

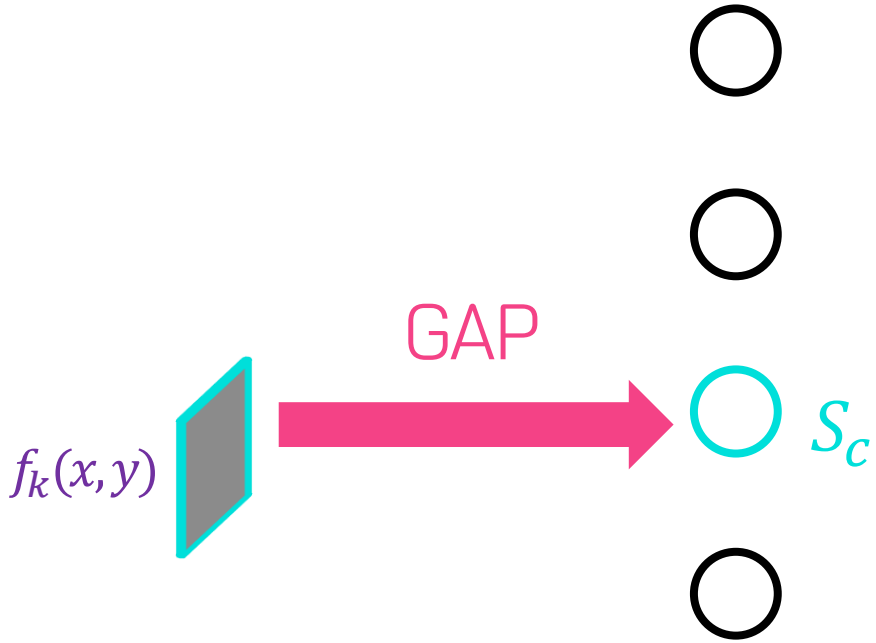


Recall: CAM

Feature Map

$$S_c = \sum_k w_k^c F_k$$

$$F_k = \sum_{x,y} f_k(x,y)$$



$$\frac{\partial S_c}{\partial F_k} = \frac{\frac{\partial S_c}{\partial f_k(x,y)}}{\frac{\partial F_k}{\partial f_k(x,y)}} = w_k^c$$

이걸 알면 GAP으로부터 자유로움

Recall: CAM

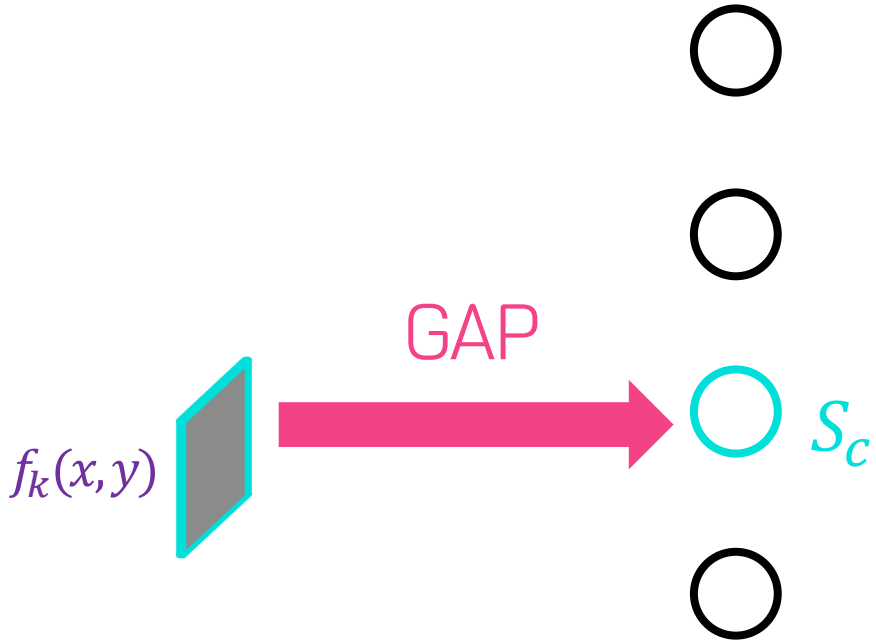
Feature Map

$$S_c = \sum_k w_k^c F_k$$

$$F_k = \sum_{x,y} f_k(x,y)$$

$$\frac{\partial S_c}{\partial F_k} = \frac{\frac{\partial S_c}{\partial f_k(x,y)}}{\frac{\partial F_k}{\partial f_k(x,y)}} = w_k^c$$

1



Recall: CAM

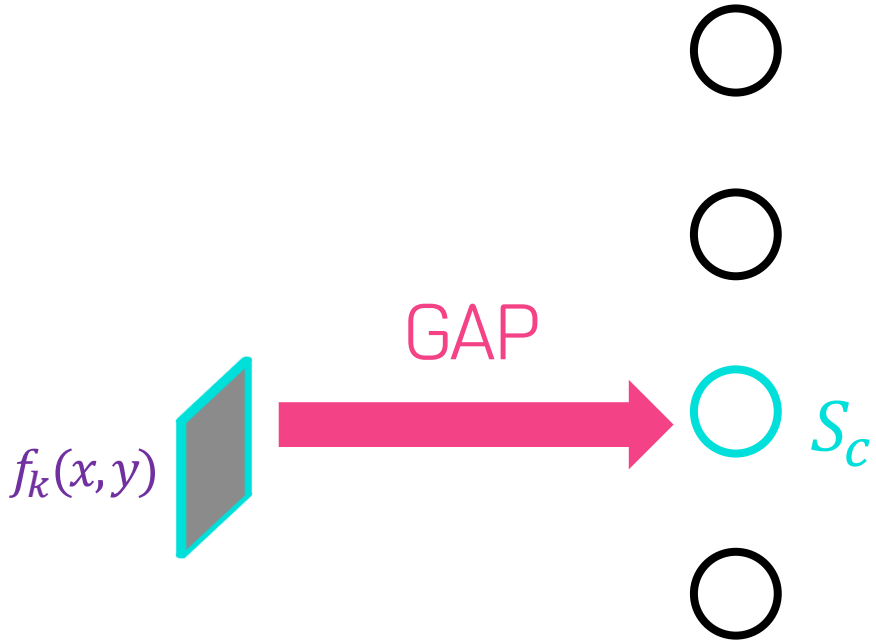
Feature Map

$$S_c = \sum_k w_k^c F_k$$

$$F_k = \sum_{x,y} f_k(x,y)$$

$$\frac{\partial S_c}{\partial F_k} = \frac{\frac{\partial S_c}{\partial f_k(x,y)}}{\frac{\partial F_k}{\partial f_k(x,y)}} = w_k^c$$

1



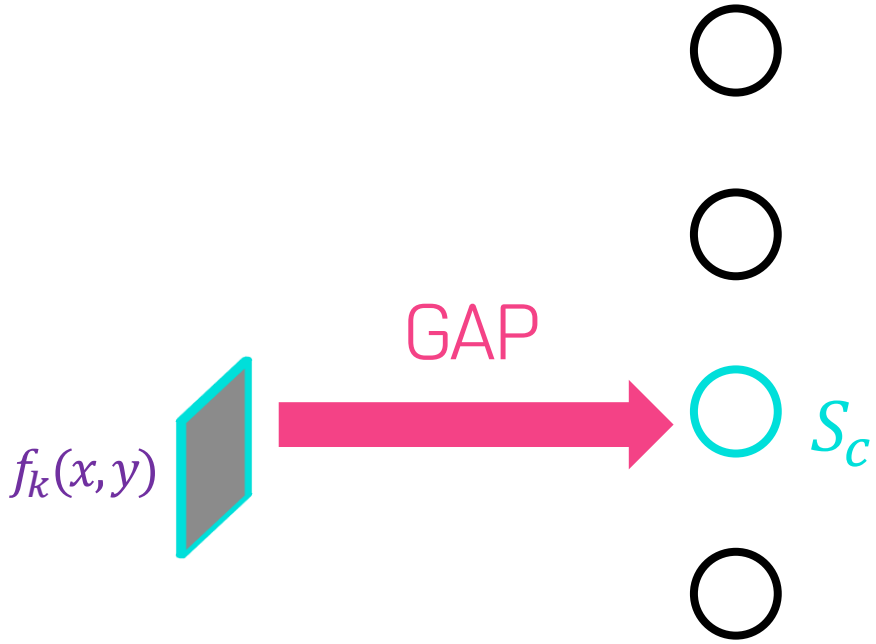
$$\frac{\partial S_c}{\partial f_k(x,y)} = w_k^c$$

Recall: CAM

Feature Map

$$S_c = \sum_k w_k^c F_k$$

$$F_k = \sum_{x,y} f_k(x,y)$$



$$\frac{\partial S_c}{\partial F_k} = \frac{\frac{\partial S_c}{\partial f_k(x,y)}}{\frac{\partial F_k}{\partial f_k(x,y)}} = w_k^c$$

1

$$\frac{\partial S_c}{\partial f_k(x,y)} = w_k^c$$

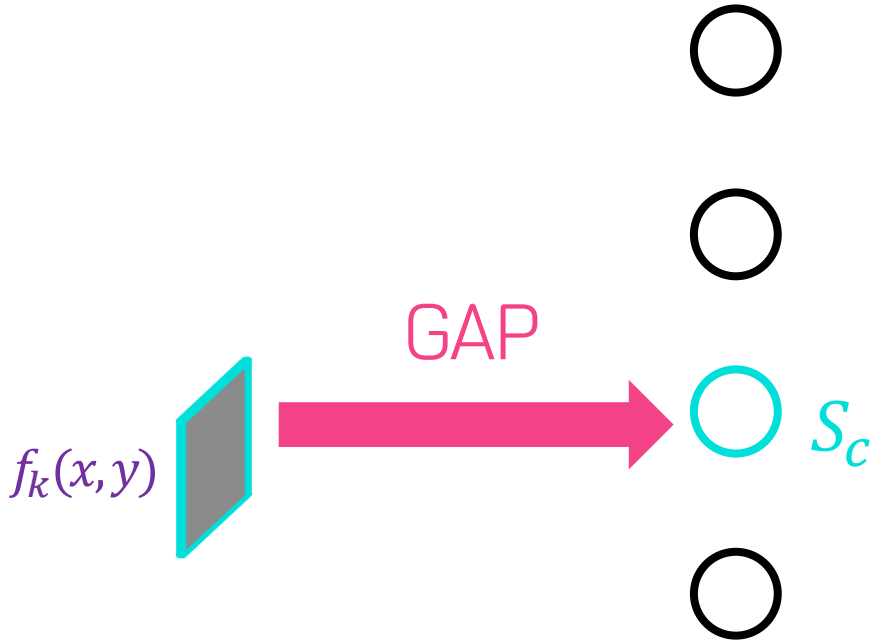
$$\sum_{x,y} \frac{\partial S_c}{\partial f_k(x,y)} = \sum_{x,y} w_k^c$$

Recall: CAM

Feature Map

$$S_c = \sum_k w_k^c F_k$$

$$F_k = \sum_{x,y} f_k(x,y)$$



$$\frac{\partial S_c}{\partial F_k} = \frac{\frac{\partial S_c}{\partial f_k(x,y)}}{\frac{\partial F_k}{\partial f_k(x,y)}} = w_k^c$$

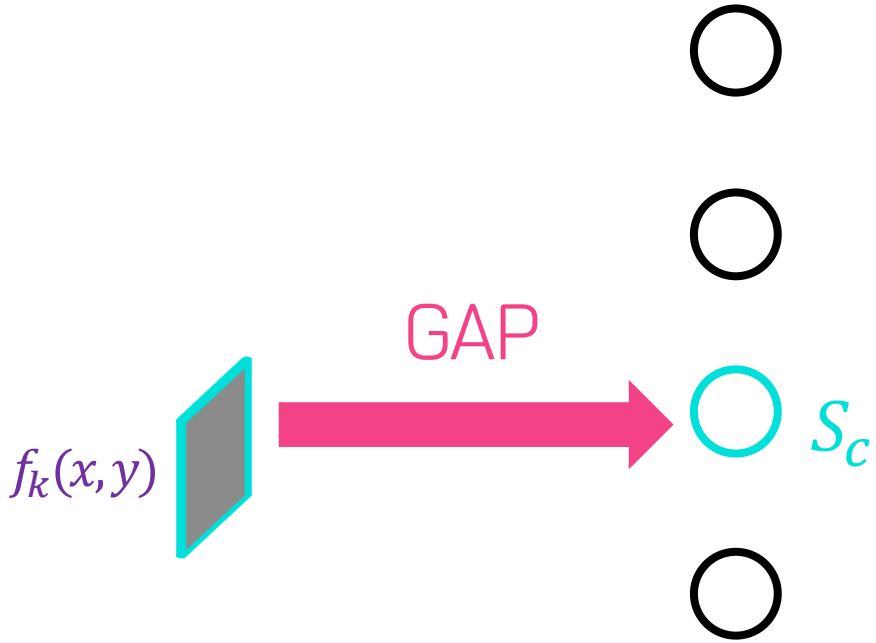
$$\frac{1}{N}\sum_{x,y} \frac{\partial S_c}{\partial f_k(x,y)} = w_k^c$$

Grad-CAM

$$\frac{1}{N}\sum_{x,y}\frac{\partial S_c}{\partial f_k(x,y)} = w_k^c$$

이걸 다르게 해석하면
(중간에 F_k가 없다)

S_c로부터 $f_k(x,y)$까지 온
Gradient를
Average Pooling을 하면
그게 w_k^c이다

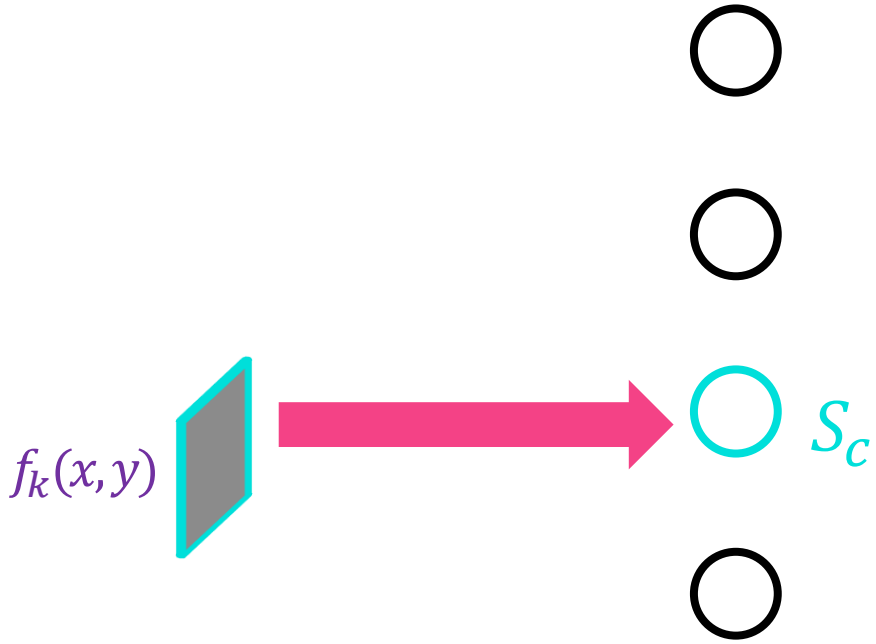


Grad-CAM

$$\frac{1}{N}\sum_{x,y}\frac{\partial S_c}{\partial f_k(x,y)} = w_k^c$$

이걸 다르게 해석하면
(중간에 F_k가 없다)

S_c로부터 $f_k(x,y)$까지 온
Gradient를
Average Pooling을 하면
그게 w_k^c이다

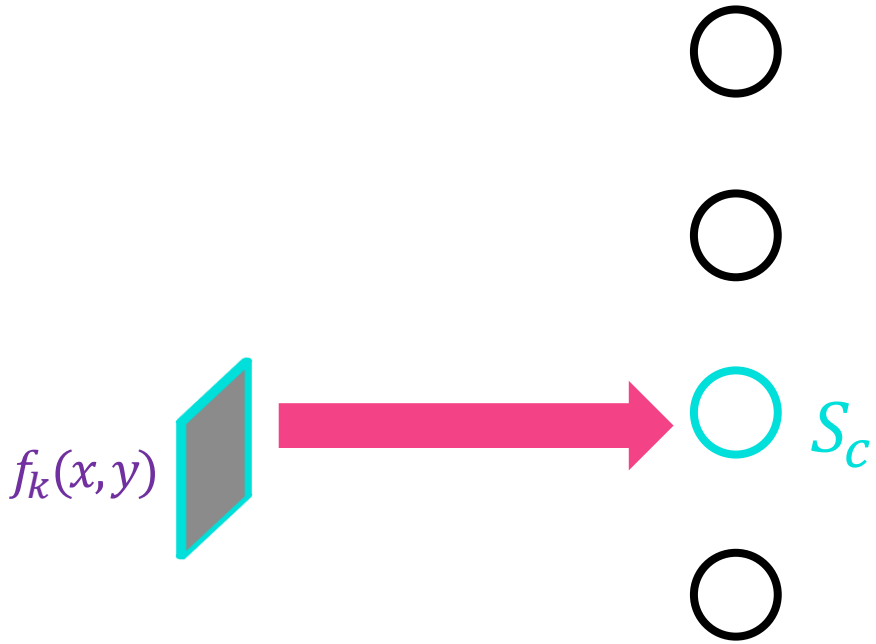


w_k^c는 결국 현재 (k번째)
Feature Map이 최종 결정에
미치는 중요도라고 볼 수 있다.

Grad-CAM

$$\frac{1}{N}\sum_{x,y}\frac{\partial S_c}{\partial f_k(x,y)} = w_k^c$$

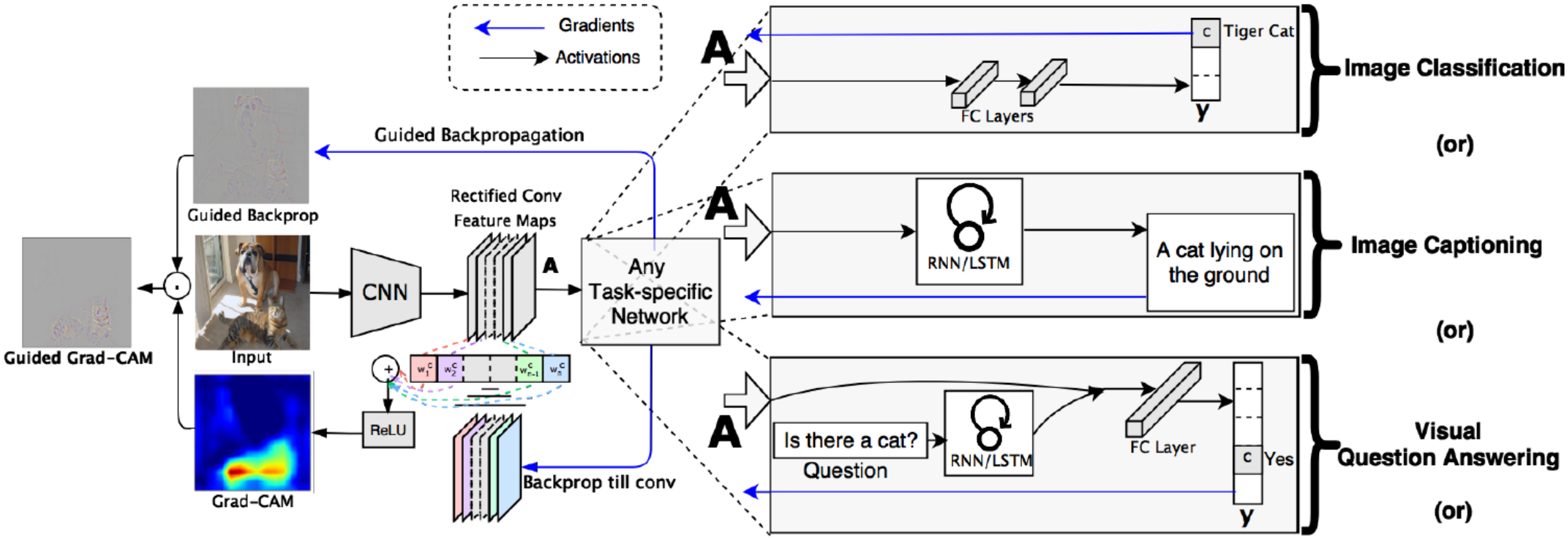
$$L_{grad} = ReLU(\sum_k w_k^c f_k(x,y))$$



Network에서 Back-prop Gradient를 가지고
Weighted Sum을 하면 Grad-CAM을 얻을 수 있다.

Grad-CAM for *Any* Networks

$$\frac{1}{N}\sum_{x,y}\frac{\partial S_c}{\partial f_k(x,y)} = w_k^c \qquad L_{grad} = ReLU(\sum_k w_k^c f_k(x,y))$$

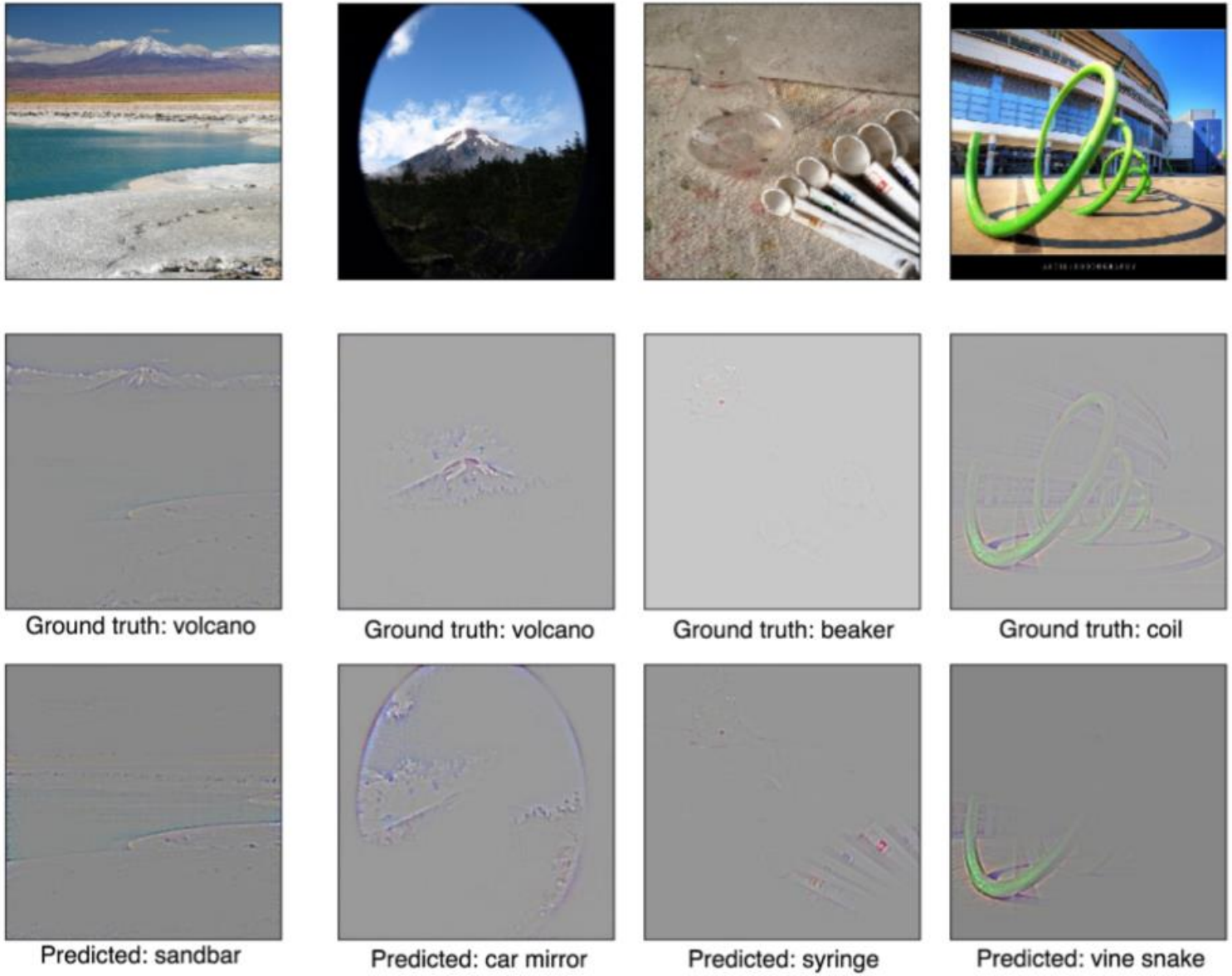


Grad-CAM Results

Method	Top-1 loc error	Top-5 loc error	Top-1 cls error	Top-5 cls error
Backprop on VGG-16 [44]	61.12	51.46	30.38	10.89
c-MWP on VGG-16 [50]	70.92	63.04	30.38	10.89
Grad-CAM on VGG-16 (ours)	56.51	46.41	30.38	10.89
VGG-16-GAP (CAM) [51]	57.20	45.14	33.40	12.20

Table 1: Classification and Localization results on ILSVRC-15 val (lower is better).

Grad-CAM Results



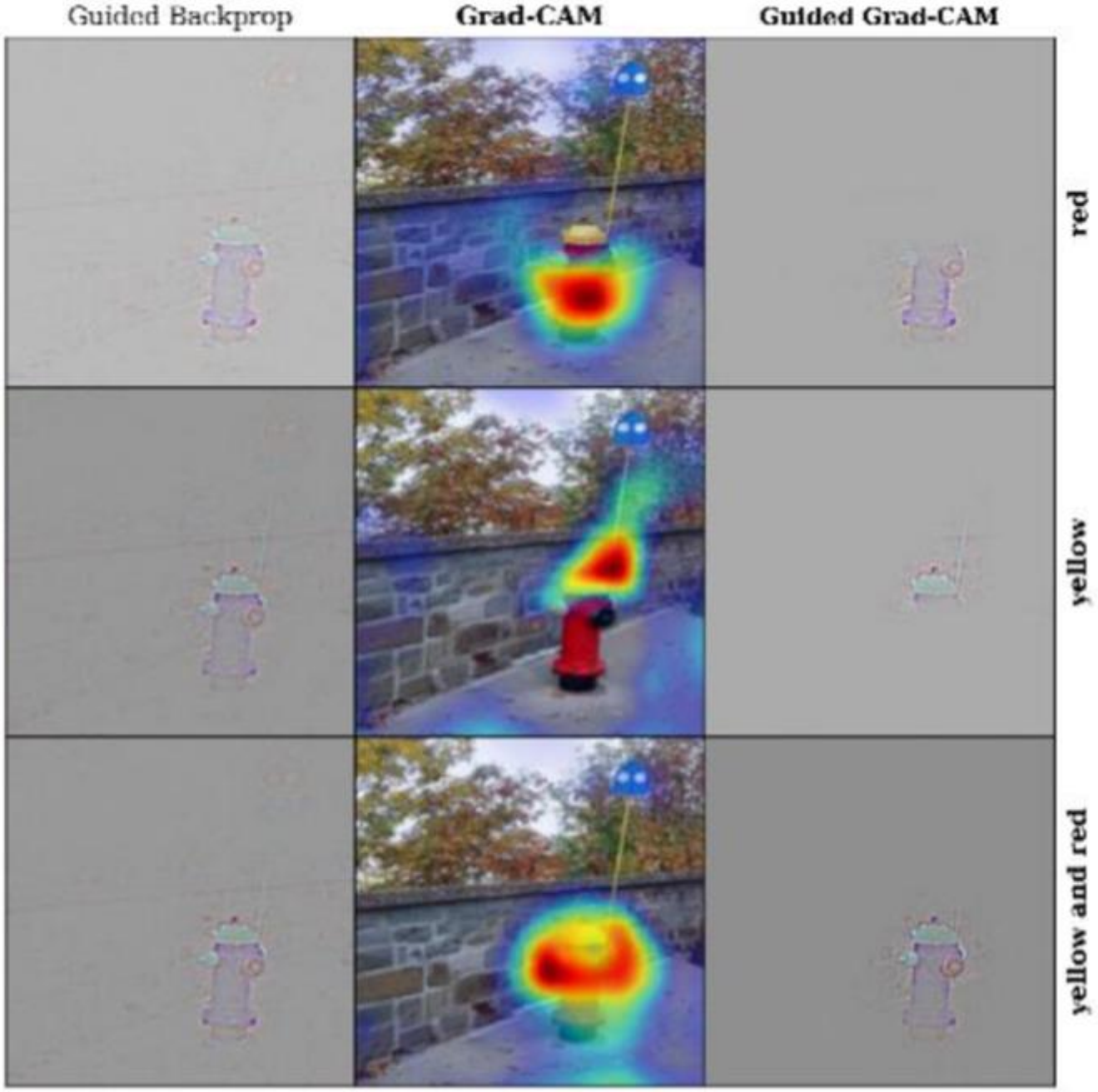
Grad-CAM Results

(a) Image captioning explanations

(b) Comparison to DenseCap

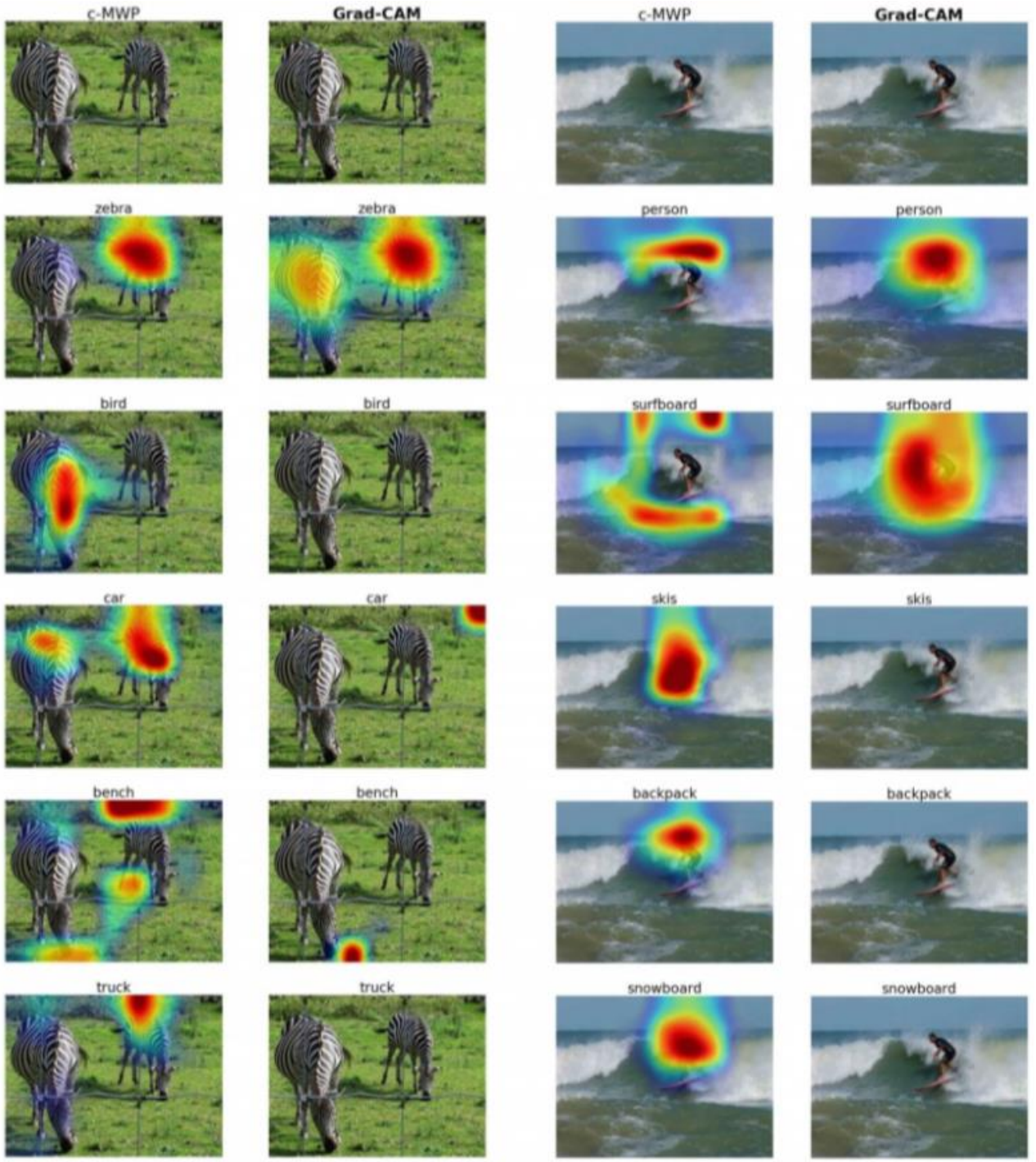
Grad-CAM Results

What color is the firehydrant?



Grad-CAM Results

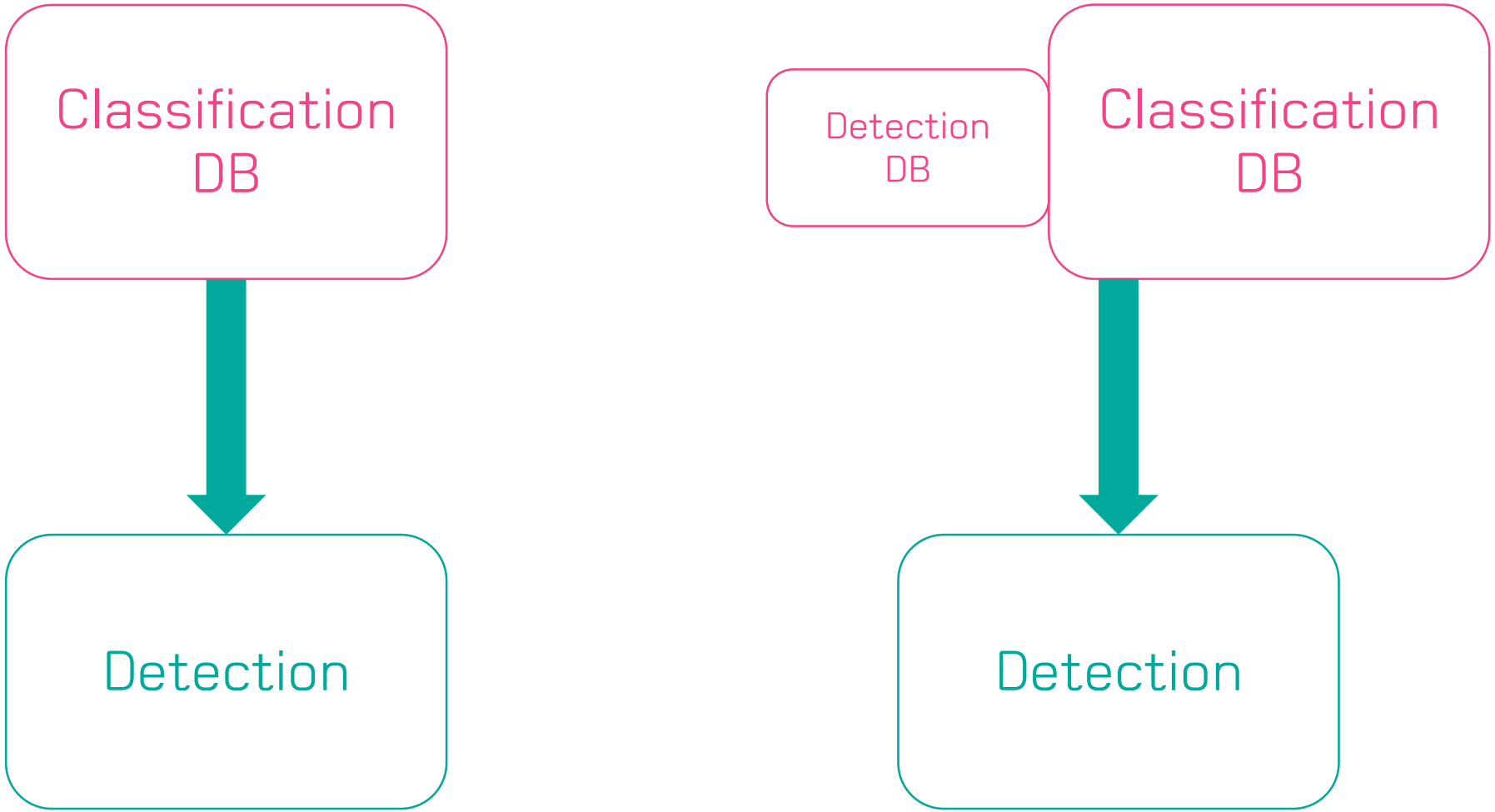
Grad-CAM Results



Bonus!

Partially Supervised Learning

Weakly vs Partially



YOLO 9000

Image Classification

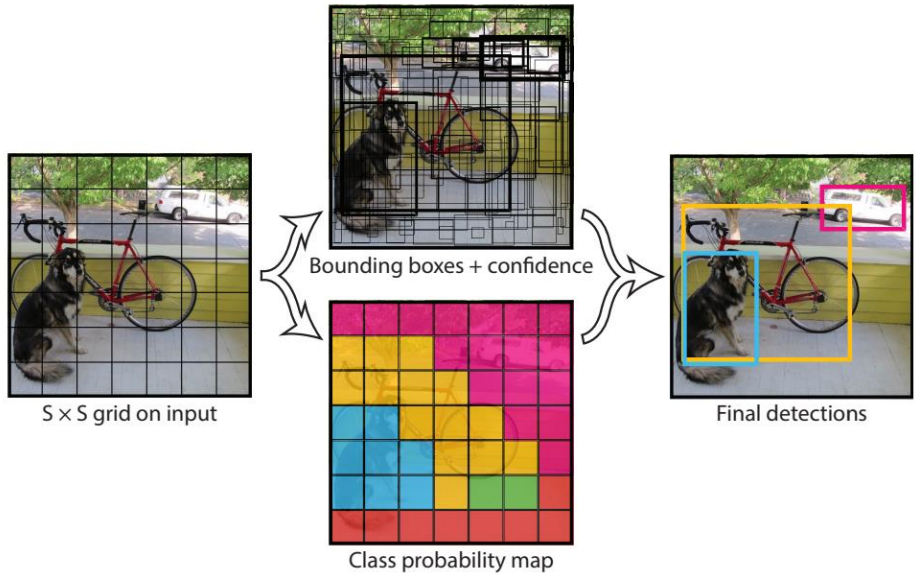
+

Object Detection

YOLO9000: Better, Faster, Stronger, CVPR 2017


JOSEPH REDMON
ALI FARHADI
RETURN IN.....
YOLO9000
Better, Faster,
Stronger
NOW PLAYING IN A DEMO NEAR YOU
@DARKNETFOREVER #YOLO9000
pjreddie.com/yolo
XNOR.AI
musher
malamute

You Only Look Once: 1–Stage Obj. Detector



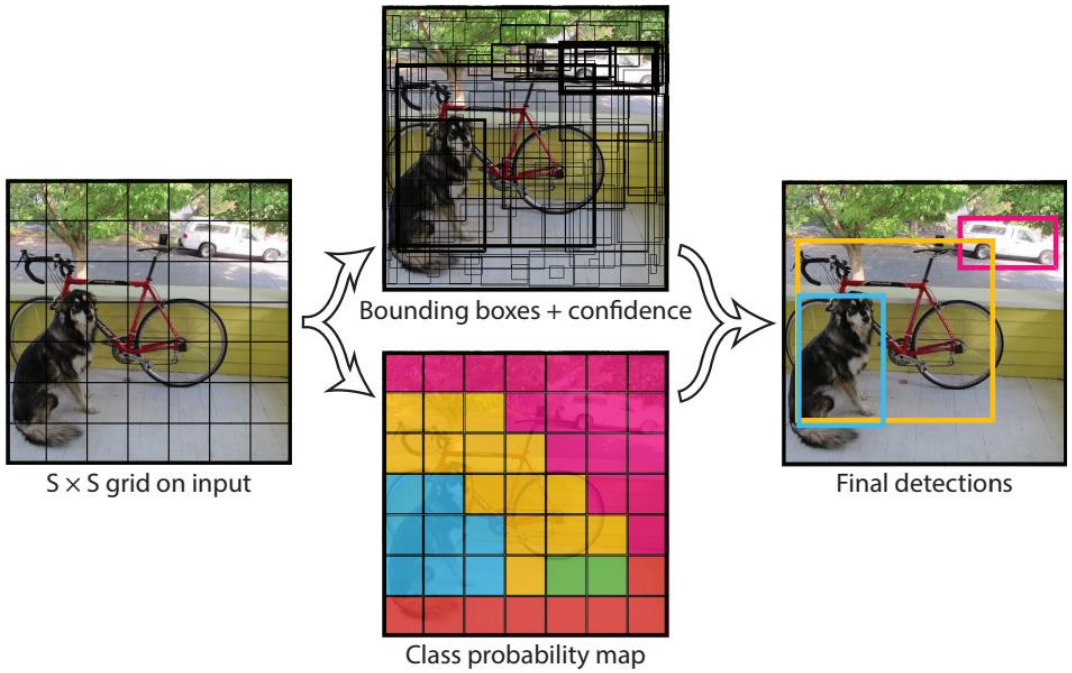
Detection Frameworks	Train	mAP	FPS
Fast R-CNN [5]	2007+2012	70.0	0.5
Faster R-CNN VGG-16[15]	2007+2012	73.2	7
Faster R-CNN ResNet[6]	2007+2012	76.4	5
YOLO [14]	2007+2012	63.4	45
SSD300 [11]	2007+2012	74.3	46
SSD500 [11]	2007+2012	76.8	19
YOLOv2 288 × 288	2007+2012	69.0	91
YOLOv2 352 × 352	2007+2012	73.7	81
YOLOv2 416 × 416	2007+2012	76.8	67
YOLOv2 480 × 480	2007+2012	77.8	59
YOLOv2 544 × 544	2007+2012	**78.6**	40

Table 3: Detection frameworks on PASCAL VOC 2007.

	YOLO							YOLOv2
batch norm?		✓	✓	✓	✓	✓	✓	✓
hi-res classifier?			✓	✓	✓	✓	✓	✓
convolutional?				✓	✓	✓	✓	✓
anchor boxes?				✓	✓			
new network?					✓	✓	✓	✓
dimension priors?						✓	✓	✓
location prediction?						✓	✓	✓
passthrough?							✓	✓
multi-scale?								✓
hi-res detector?								✓
VOC2007 mAP	63.4	65.8	69.5	69.2	69.6	74.4	75.4	76.8	**78.6**

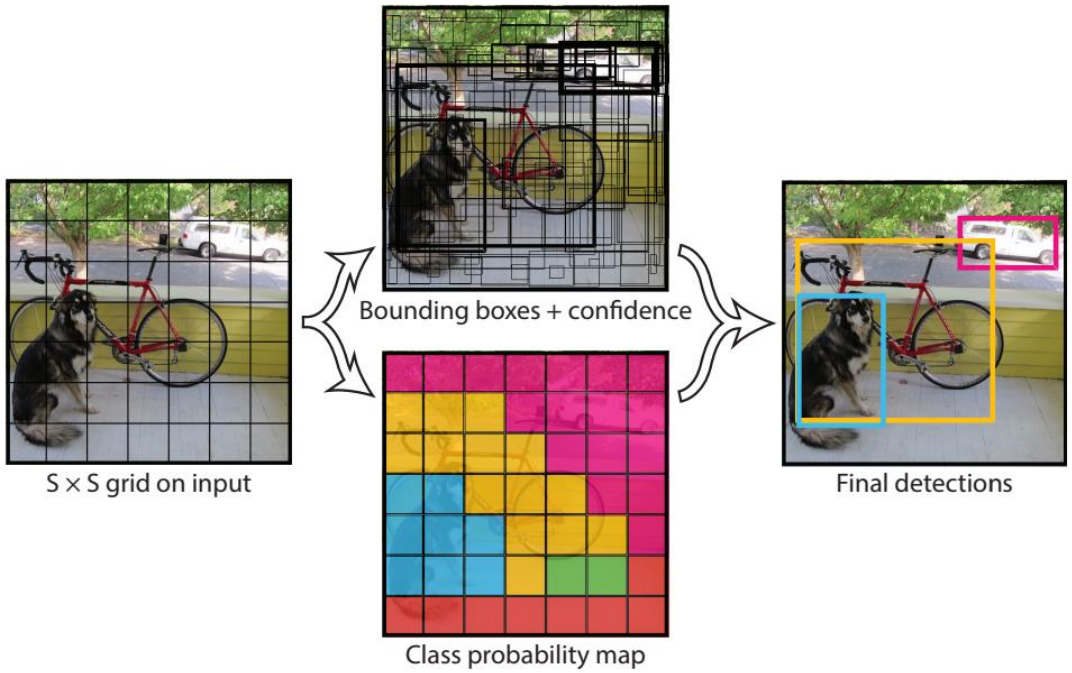
YOLOv1: Unified, Real-time Object Detection, CVPR 2016
YOLO9000: Better, Faster, Stronger, CVPR 2017

You Only Look Once: 1–Stage Obj. Detector Brief Review



Deep Network

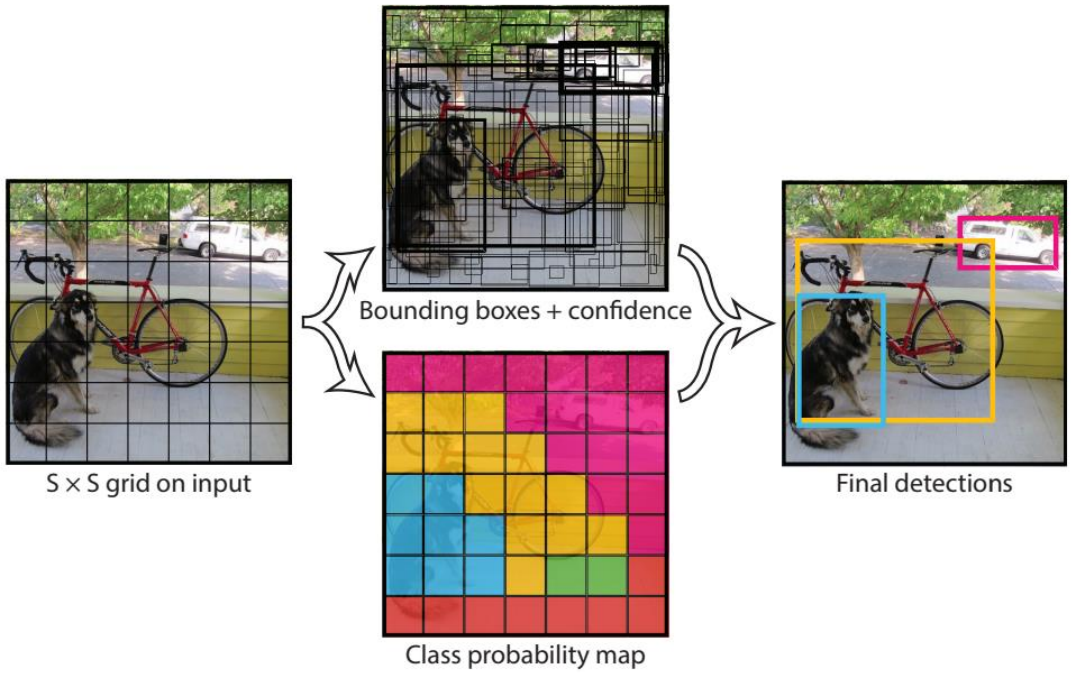
You Only Look Once: 1–Stage Obj. Detector Brief Review



Deep Network

Predict BBoxes

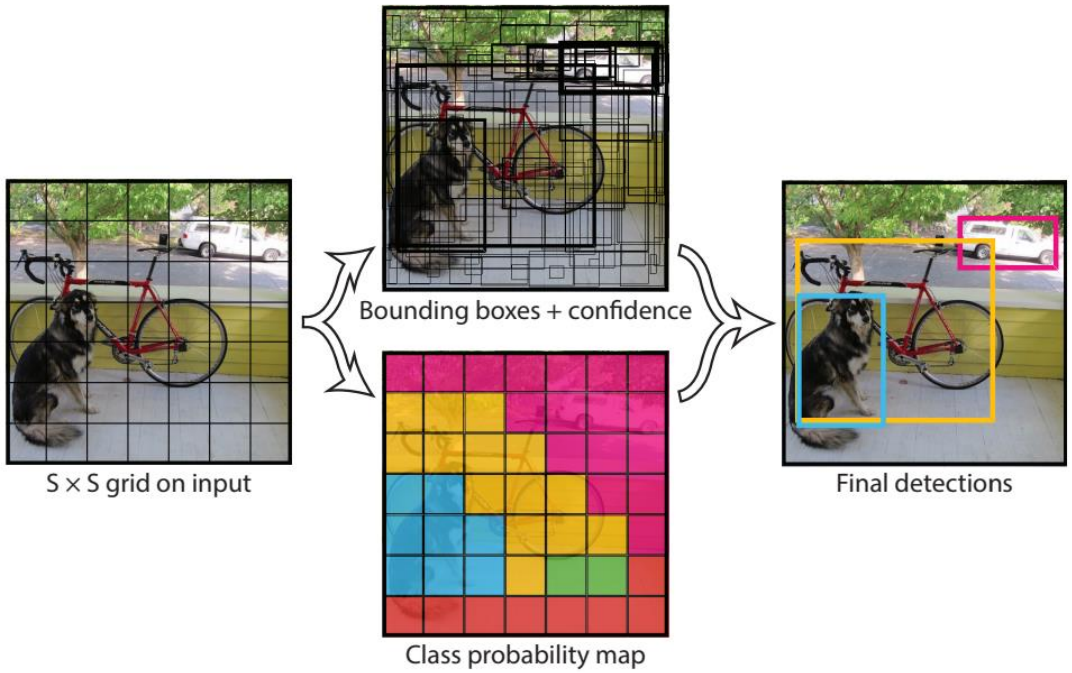
You Only Look Once: 1–Stage Obj. Detector Brief Review



Deep Network

Predict BBoxes
Predict Confidence
Objectness
IoU / From DB

You Only Look Once: 1–Stage Obj. Detector Brief Review



Deep Network

Predict BBoxes
Predict Confidence
Objectness
IoU / From DB
Predict Class

YOLO 9000

- Better
- Faster
- Stronger

YOLO 9000

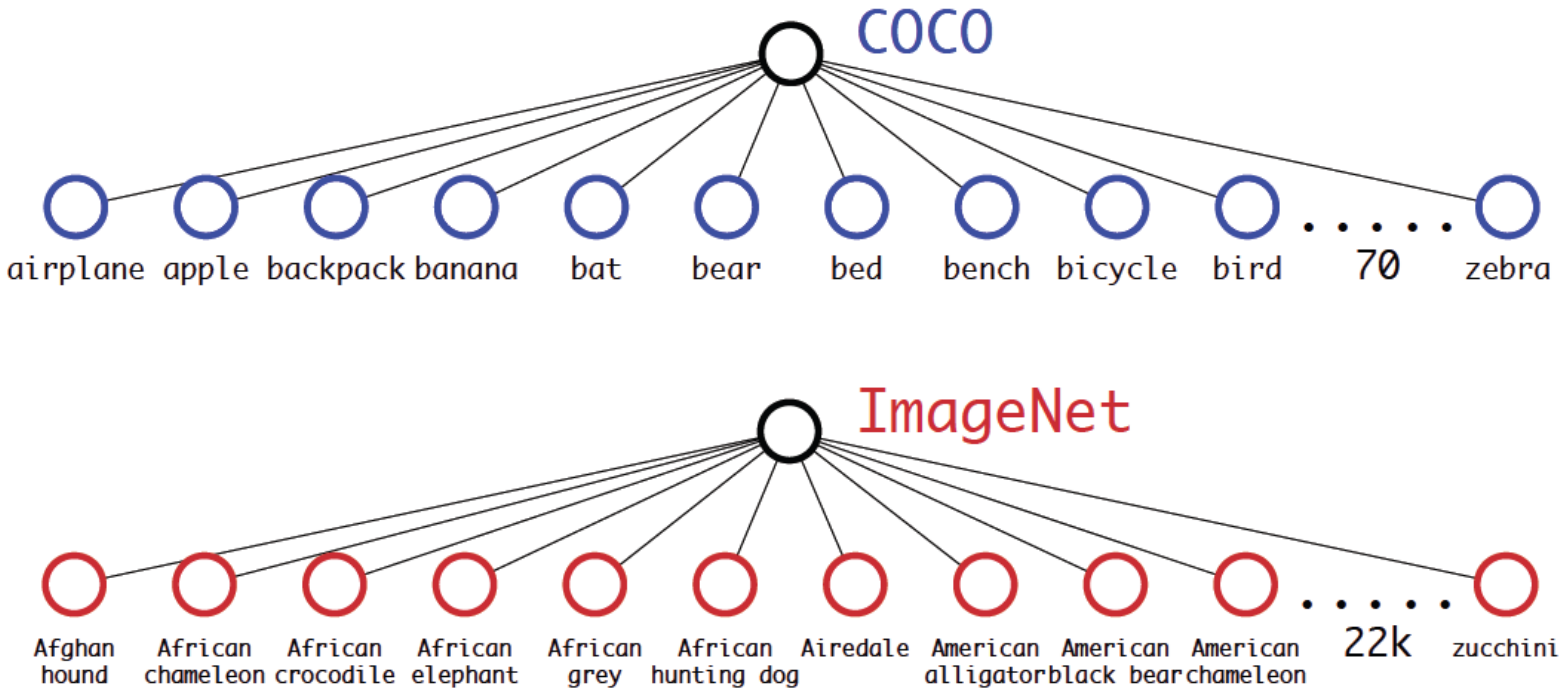
- Better YOLO v2
- Faster YOLO v2
- Stronger

YOLO 9000

- Better YOLO v2
- Faster YOLO v2
- Stronger YOLO 9000

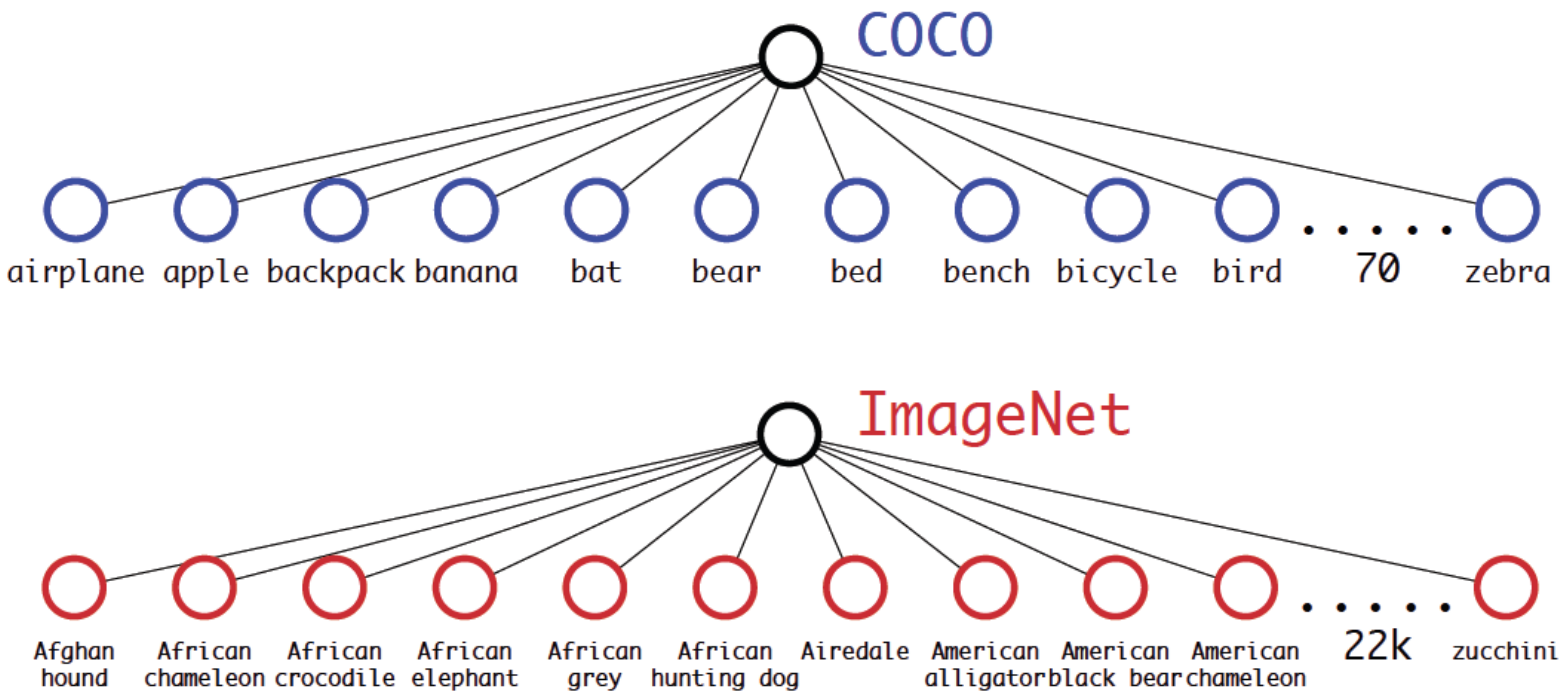
Problem: Dataset

- COCO (Object Detection)
- ImageNet (Classification)



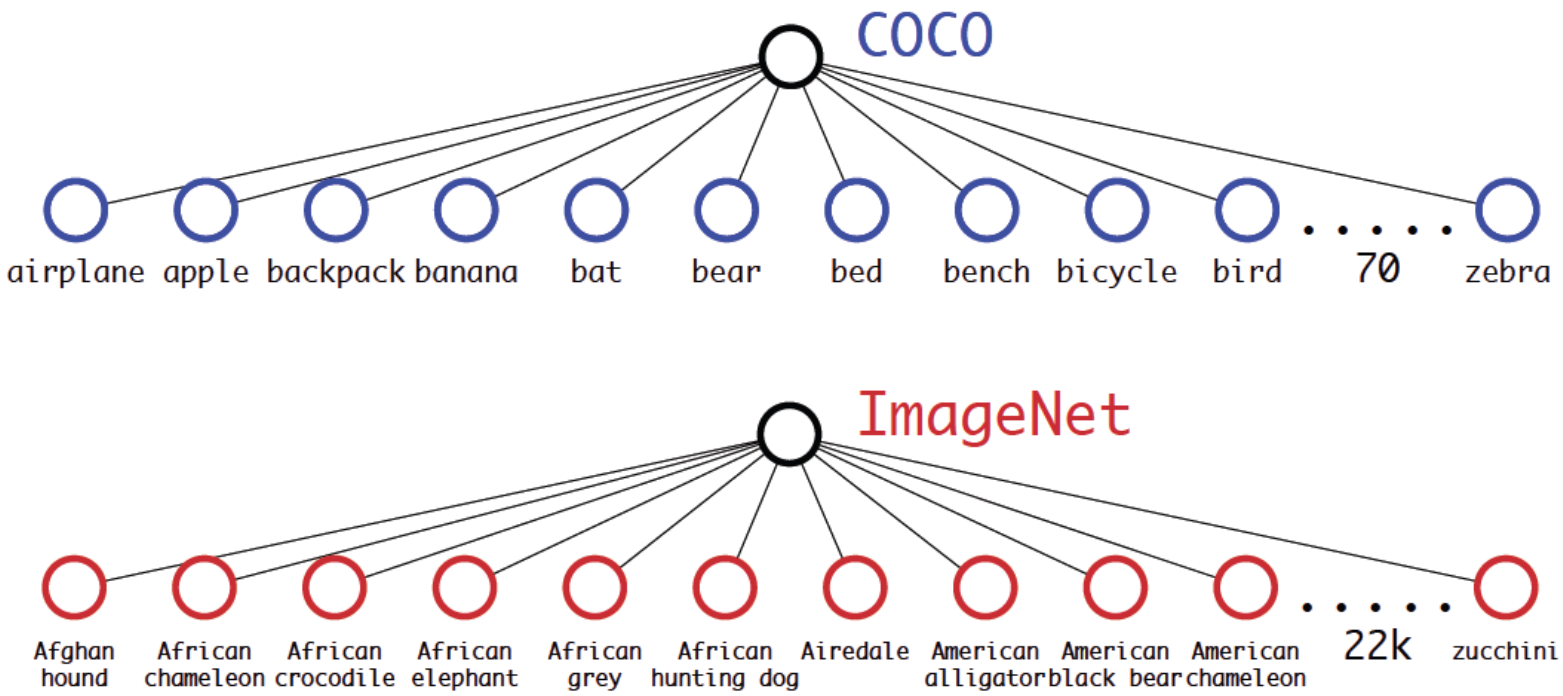
Problem: Dataset

- COCO (Object Detection)
- ImageNet (Classification): Object Location 정보는 없다



Problem: Dataset

Goal: 수천개의 Label로 Object Detection을 하자



Training YOLO-9000

Training COCO Detection Dataset
: Backprop BBox / Objectness / Class

Training YOLO-9000

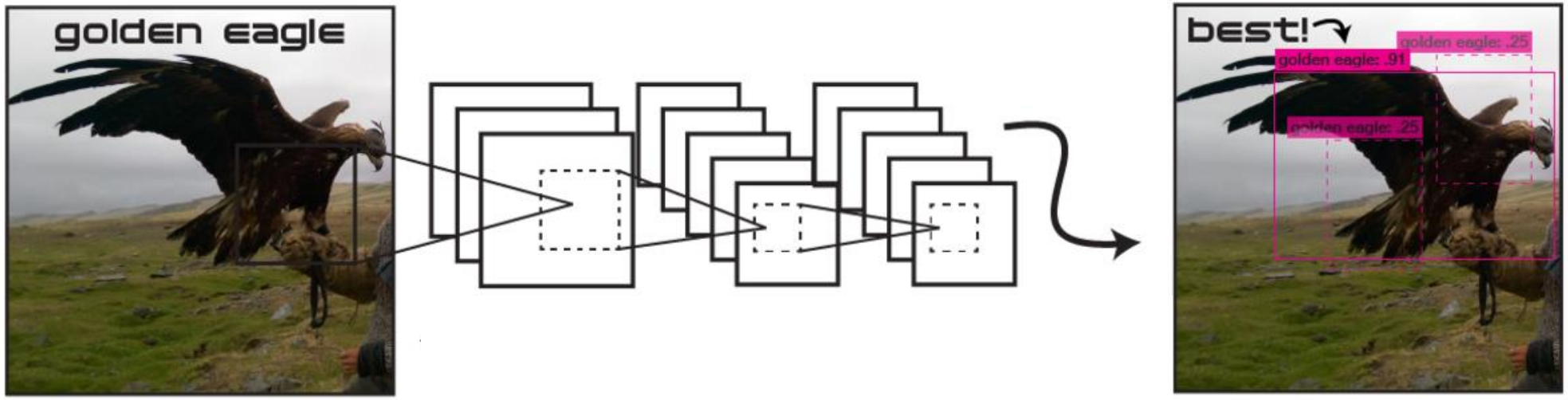
Training COCO Detection Dataset
: Backprop BBox / Objectness / Class

Training ImageNet Classification Dataset
: Run YOLO

Training YOLO-9000

Training COCO Detection Dataset
: Backprop BBox / Objectness / Class

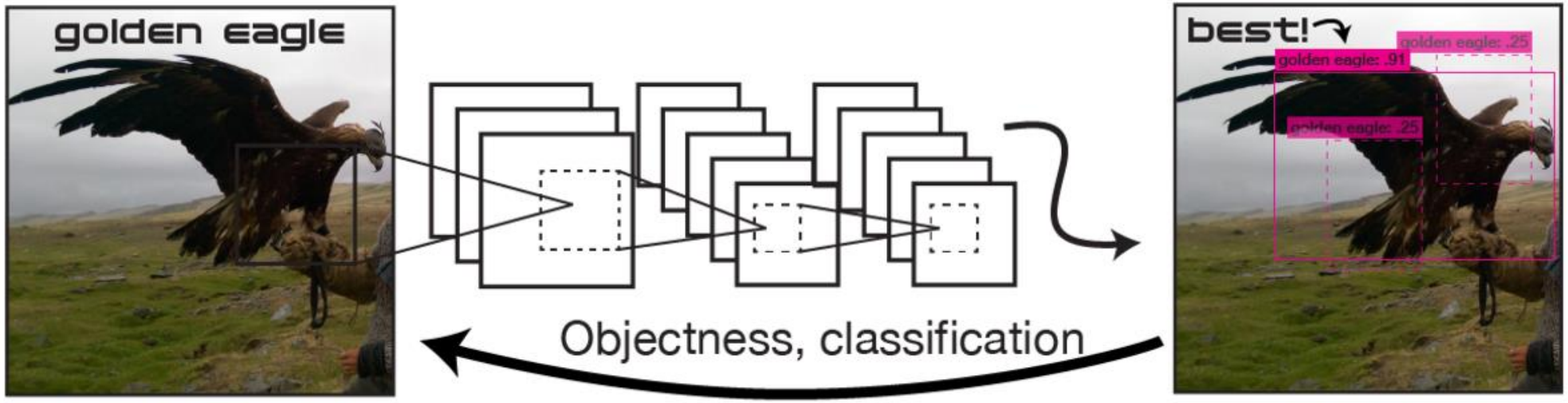
Training ImageNet Classification Dataset
: Run YOLO (Without Class) + Pick Best Objectness



Training YOLO-9000

Training COCO Detection Dataset
: Backprop BBox / Objectness / Class

Training ImageNet Classification Dataset
: Backprop Objectness / Class

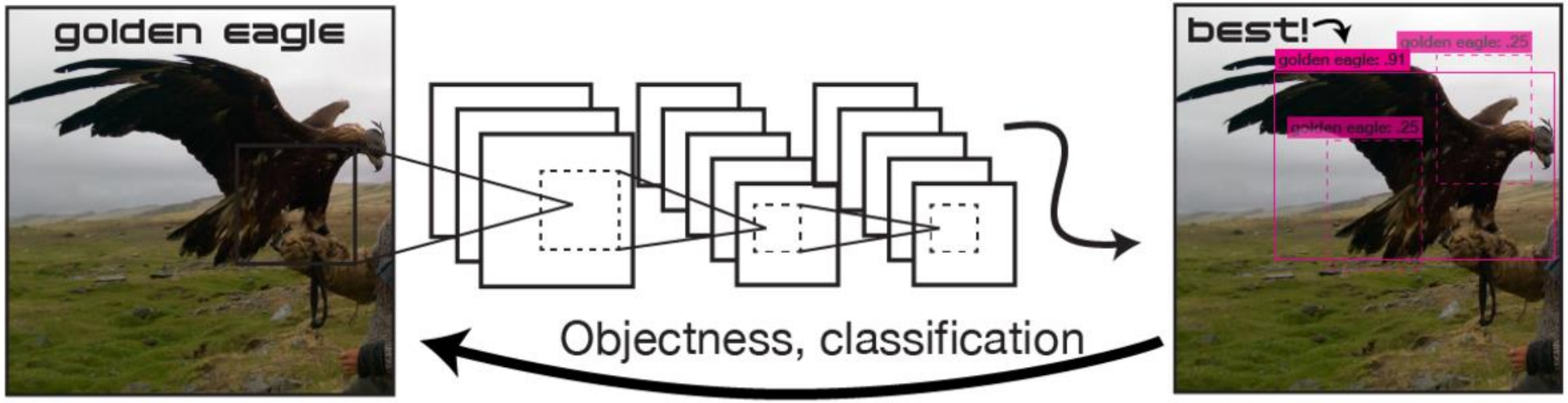


Training YOLO-9000

Training COCO Detection Dataset
: Backprop BBox / Objectness / Class

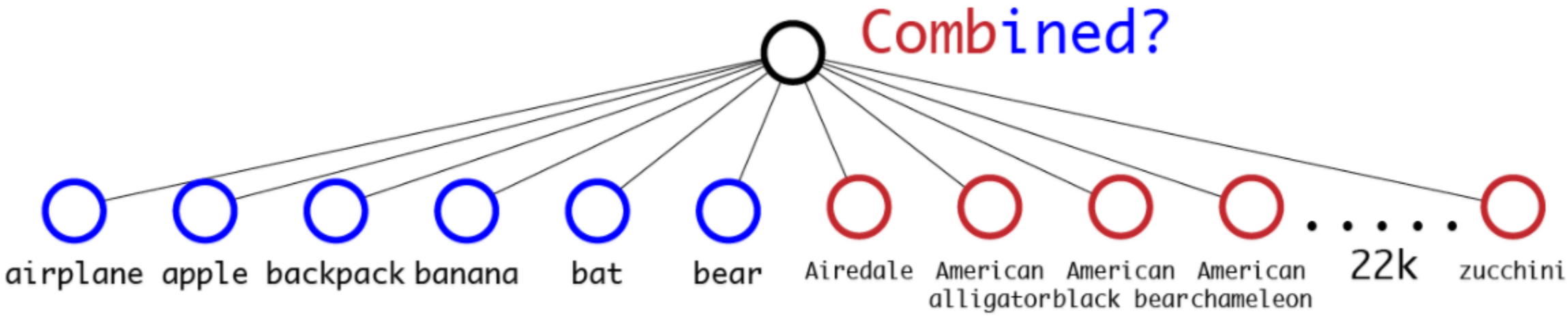
Oversample!

Training ImageNet Classification Dataset
: Backprop Objectness / Class



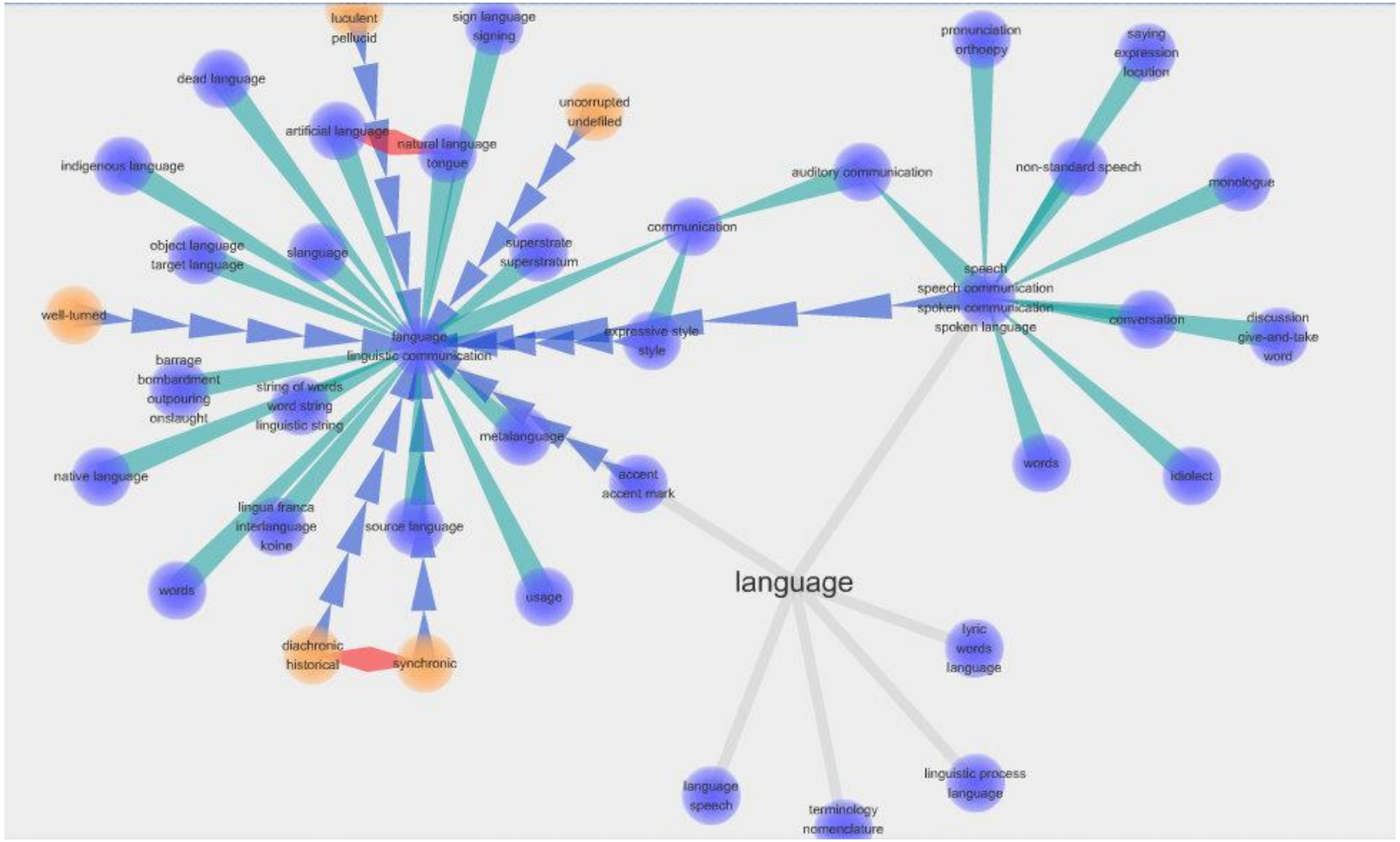
How to train with “LABEL”???

단순히 합쳐서? 그건 좀...



Focus on “Word”

ImageNet Label은 WordNet 기반으로 Label 되어 있음
WordNet은 매우 복잡한 Graph 구조



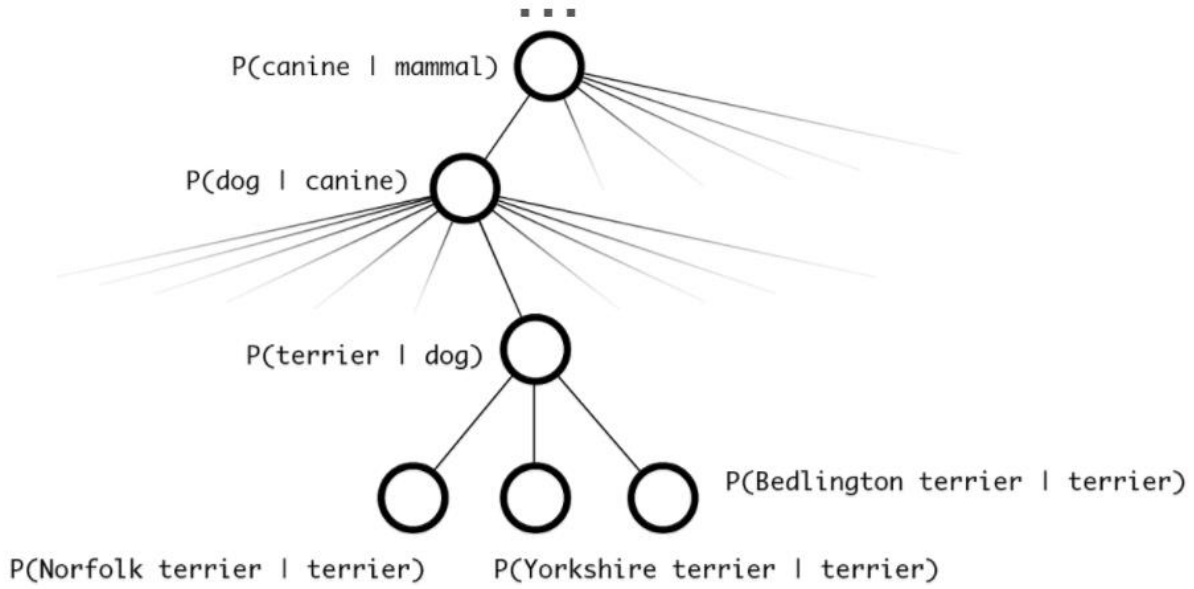
Focus on “Word”

1. ImageNet Label을 본다
   - ImageNet Label은 1000개 이상

Focus on “Word”

1. ImageNet Label을 본다
   - ImageNet Label은 1000개 이상

2. 해당 Label에서 “Physical Object”까지 가는 “가장 짧은 길”을 가지고 Tree 구조를 만든다



Focus on “Word”

1. ImageNet Label을 본다
   - ImageNet Label은 1000개 이상

2. 해당 Label에서 “Physical Object”까지 가는 “가장 짧은 길”을 가지고 Tree 구조를 만든다

3. 모든 Label에 대해서 반복해서 Tree를 완성한다

Focus on “Word”

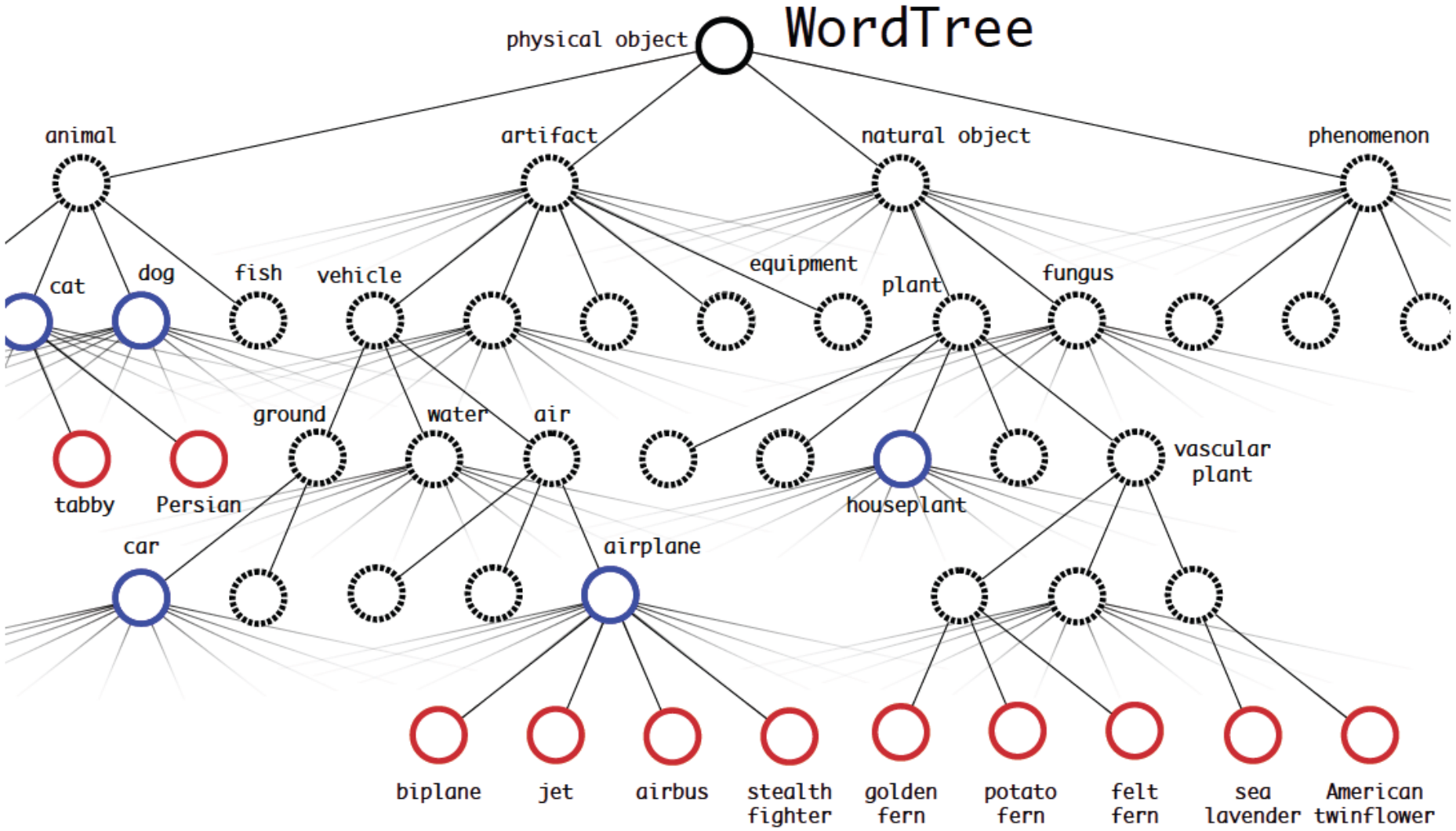
1. ImageNet Label을 본다
   - ImageNet Label은 1000개 이상

2. 해당 Label에서 “Physical Object”까지 가는 “가장 짧은 길”을 가지고 Tree 구조를 만든다

3. 모든 Label에 대해서 반복해서 Tree를 완성한다

4. COCO DB에 대해서도 수행

WordTree


WordTree
physical object
animal
artifact
natural object
phenomenon
cat
dog
fish
vehicle
equipment
plant
fungus
ground
water
air
vascular plant
tabby
Persian
houseplant
car
airplane
biplane
jet
airbus
stealth fighter
golden fern
potato fern
felt fern
sea lavender
American twinflower

Label Propagation

이제 COCO DB와 ImageNet DB의 Label (Ground Truth)를
“다시 한다”

Label Propagation

이제 COCO DB와 ImageNet DB의 Label (Ground Truth)를
“다시 한다”

→ Tree 상에서 해당 Label 위로 모두 1로 Label !

Label Propagation

이제 COCO DB와 ImageNet DB의 Label (Ground Truth)를
"다시 한다"

→ Tree 상에서 해당 Label 위로 모두 1로 Label !

Yorkshire Terrier

Label Propagation

이제 COCO DB와 ImageNet DB의 Label (Ground Truth)를
"다시 한다"

→ Tree 상에서 해당 Label 위로 모두 1로 Label !

Now, Train Again

이제 학습을 하면, Network는 수천개의 Class를 예측

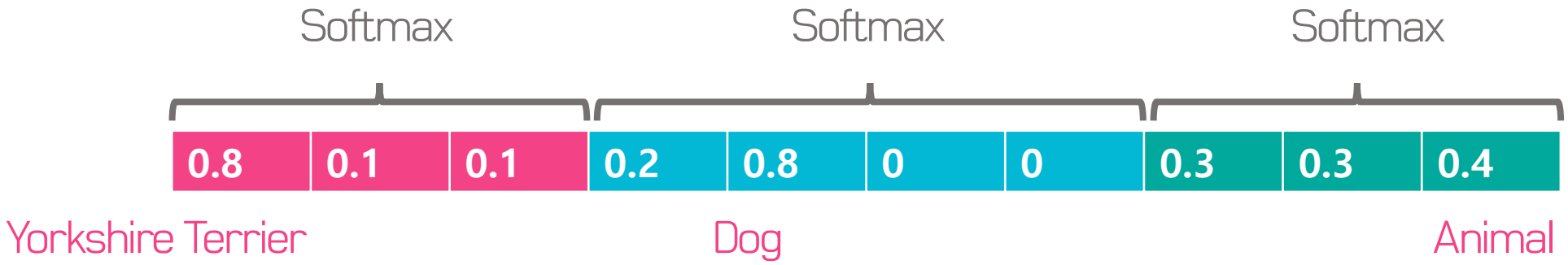
그냥 Softmax 쓰면 안된다. 우리는 Multi-label을 했으니까

Now, Train Again

이제 학습을 하면, Network는 수천개의 Class를 예측

그냥 Softmax 쓰면 안된다. 우리는 Multi-label을 했으니까

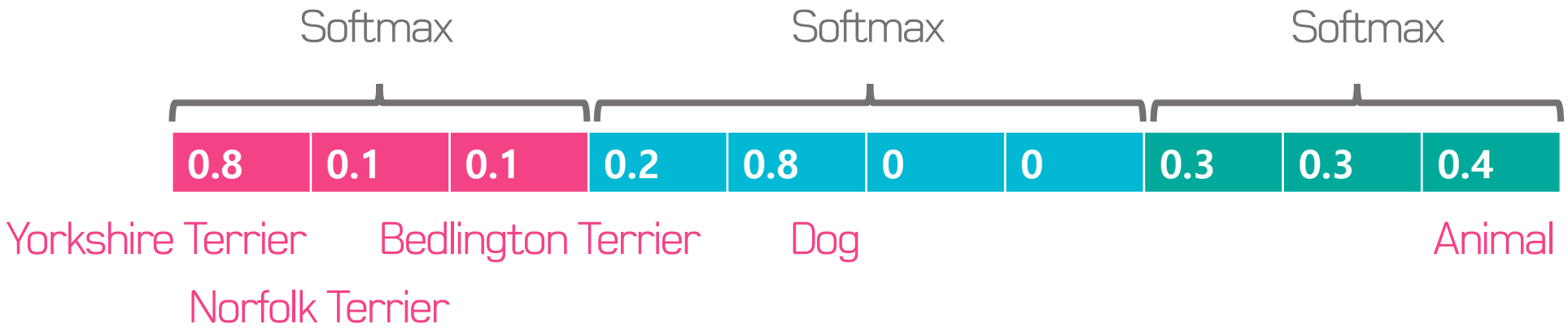
그래서 "동위어" 별 Softmax를 해서 Ground Truth와 비교해서 학습한다



Prediction

이걸 비교해서 예측을 한다

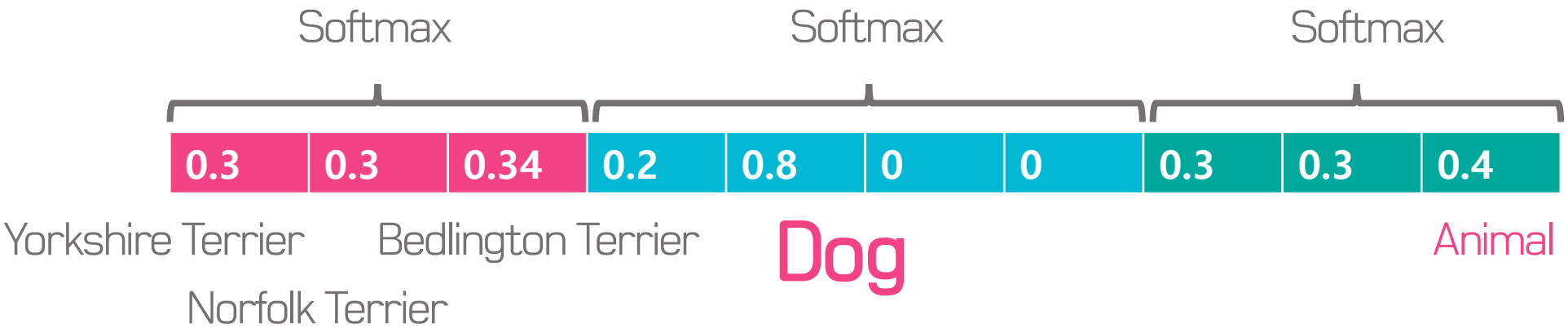
$$Pr(\text{Norfolk terrier}|\text{terrier})$$
$$Pr(\text{Yorkshire terrier}|\text{terrier})$$
$$Pr(\text{Bedlington terrier}|\text{terrier})$$



Prediction

이걸 비교해서 예측을 한다

값이 낮으면 상위 단어로 예측!



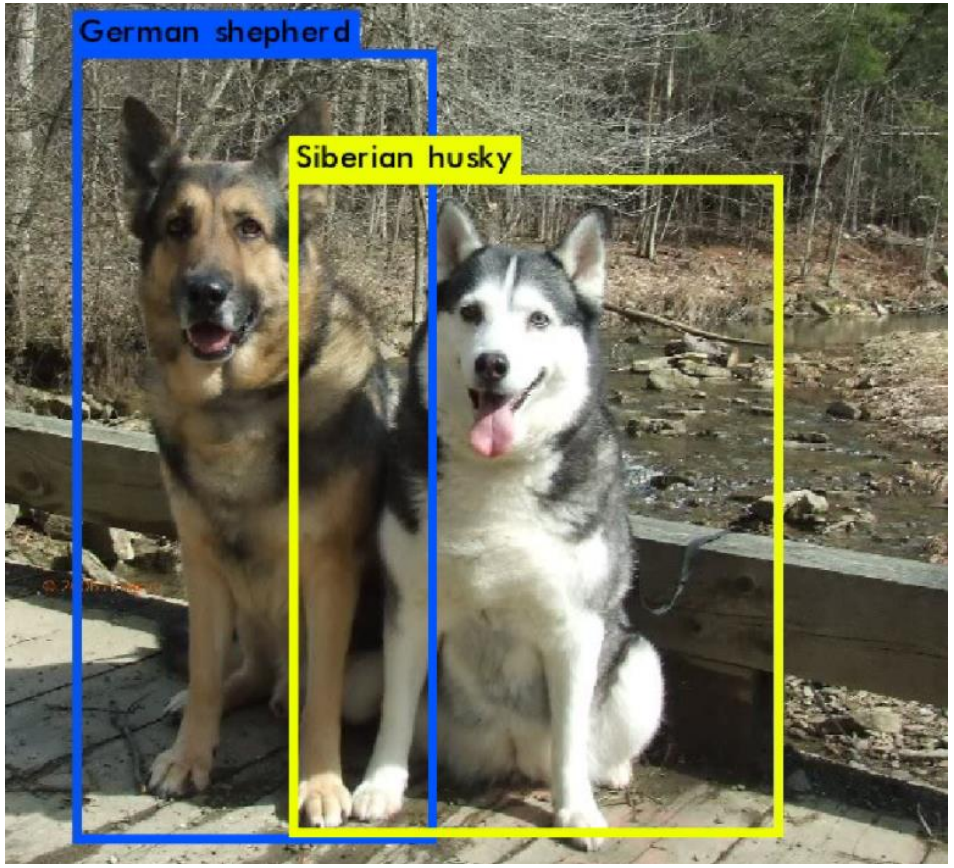
YOLO 9000 Results

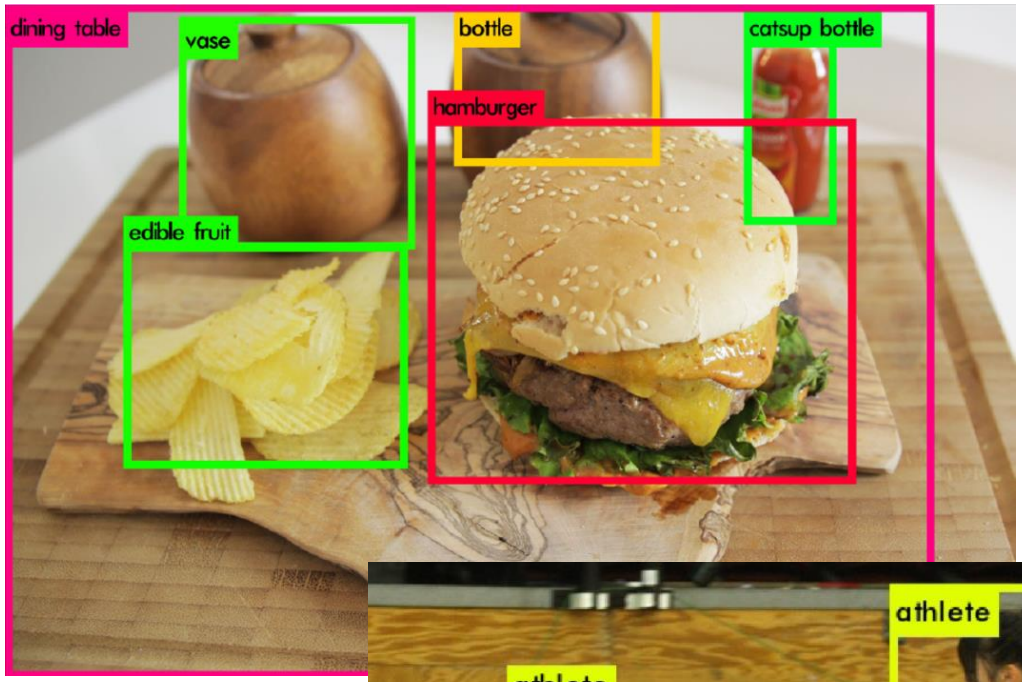
COCO 80개 Class + ImageNet Top 9000개 Class → 9418 Class

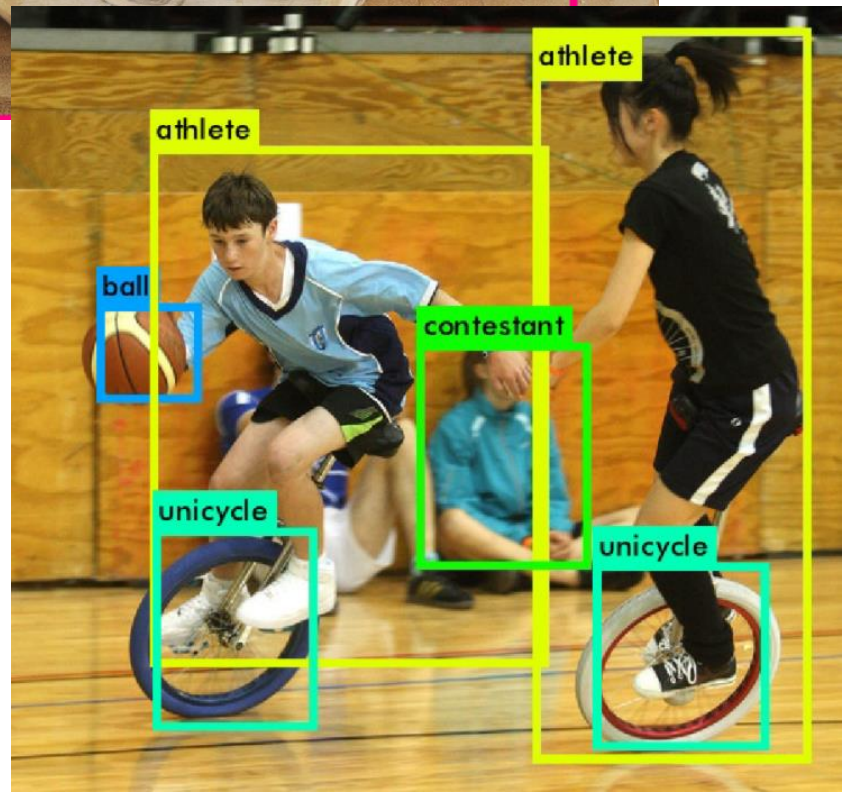
ImageNet Detection에 Test
- 200개 Class
- 156개의 Object Detection Class는 학습 x
  - Classification DB + WordTree로만 학습

- 19.7mAP
- 16.0mAP (156 Class)

YOLO 9000 Results







Mask^{X} R-CNN

Object Detection

+

Instance Segmentation

MaskX R-CNN: Learning to Segment Every Thing, Arxiv 2017.11

2017. 11. 28

Learning to Segment Every Thing

Ronghang Hu[1,2,*] Piotr Dollár[2] Kaiming He[2] Trevor Darrell[1] Ross Girshick[2]

[1]BAIR, UC Berkeley [2]Facebook AI Research (FAIR)

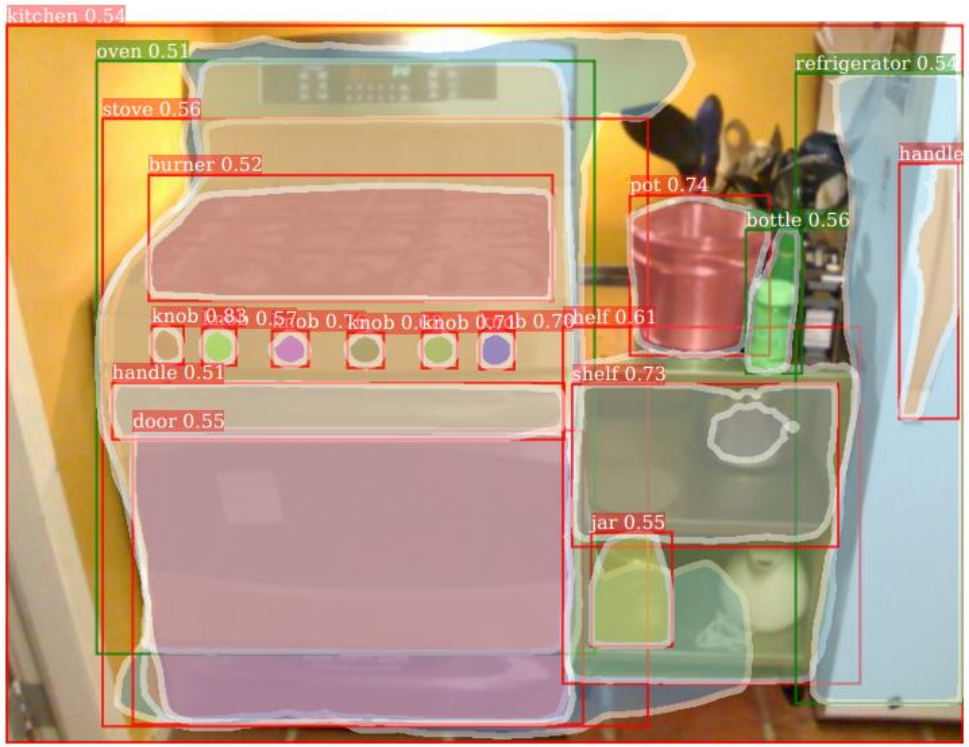


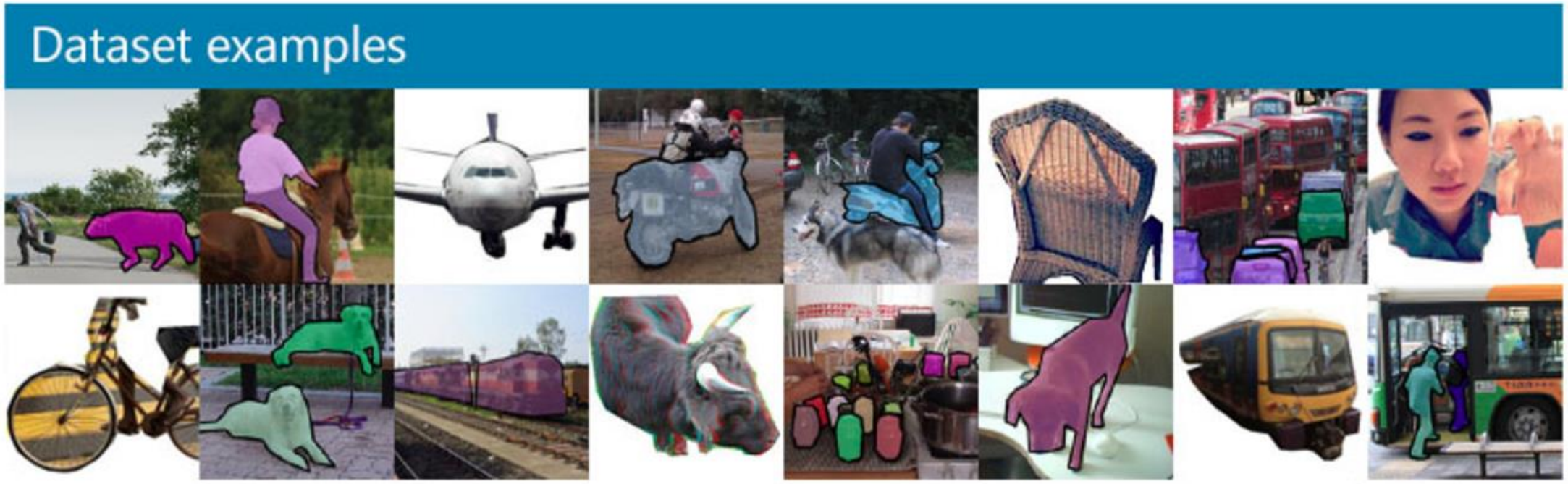
Figure 1. **We explore training instance segmentation models with partial supervision**: a subset of classes (green boxes) have instance mask annotations during training; the remaining classes (red boxes) have only bounding box annotations. This image shows output from our model trained for 3000 classes from Visual Genome, using mask annotations from only 80 classes in COCO.

Mask X R-CNN

Instance Segmentation using Detection (BBox) Dataset

Dataset

MS COCO: 80 Classes
with Segmentation Label

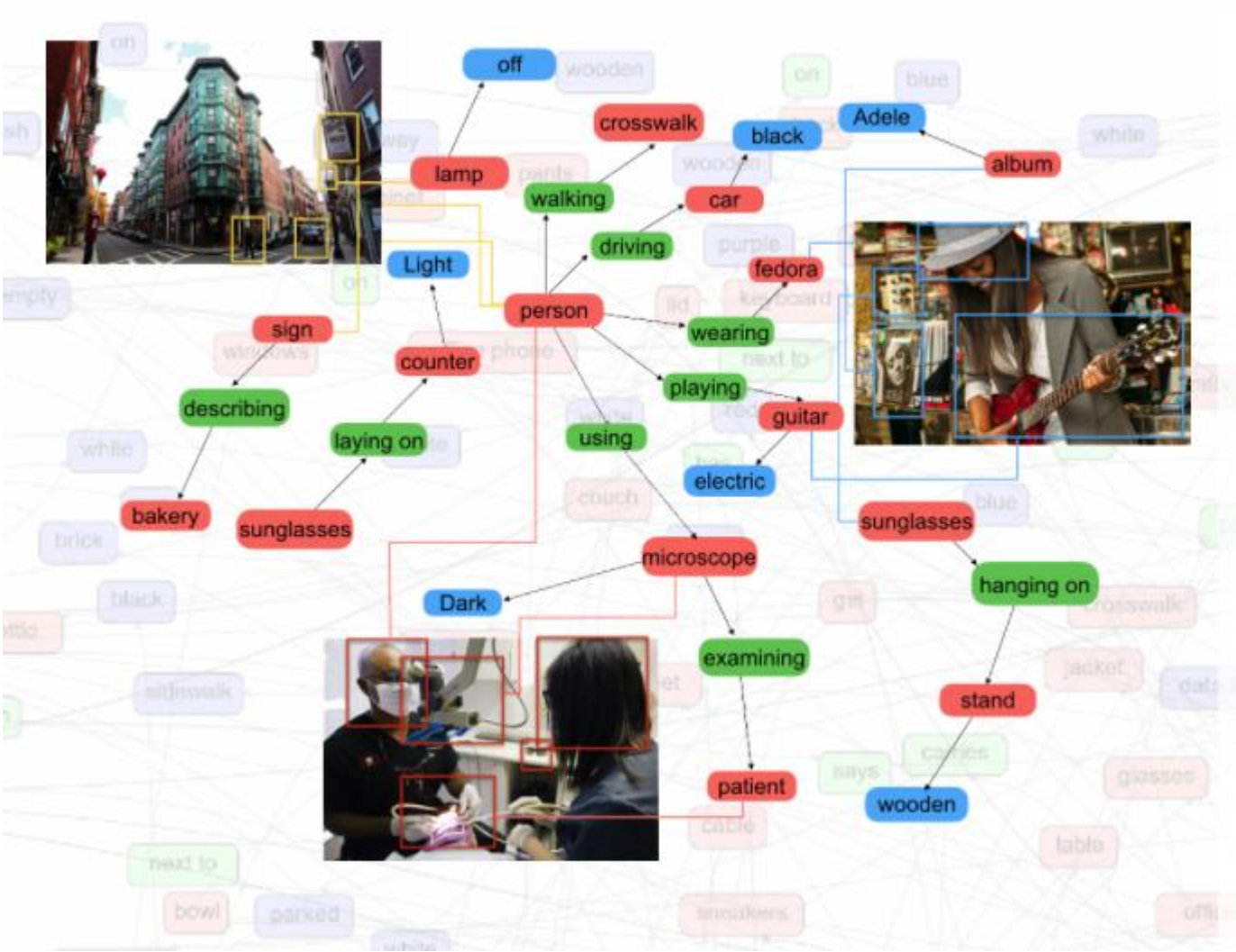


Dataset

Visual Genome

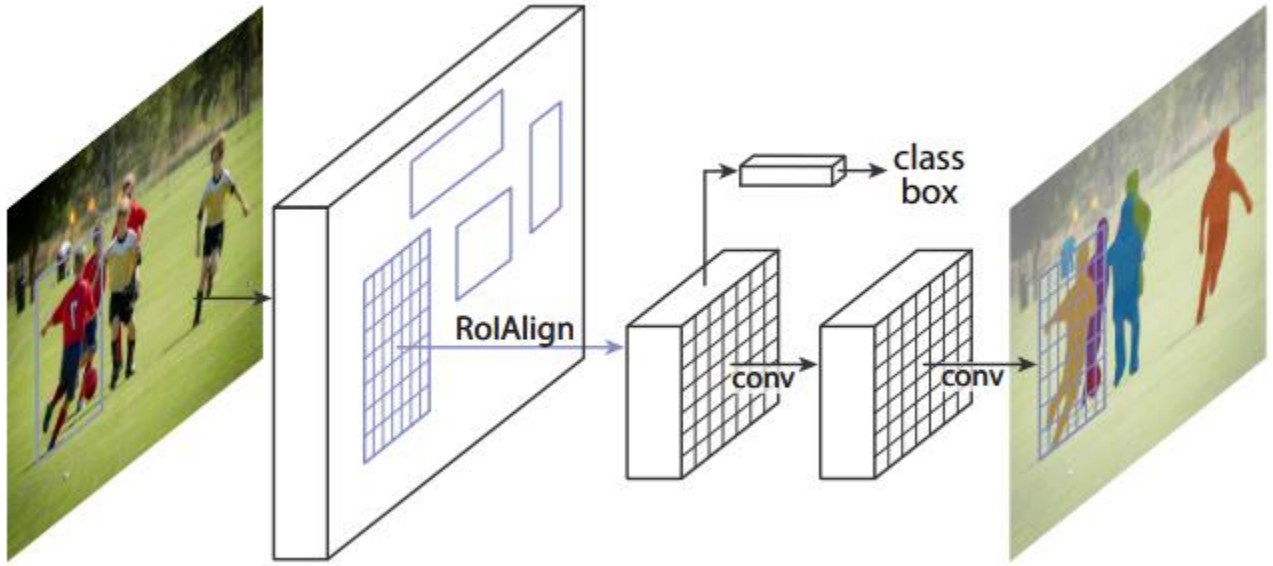
+ 3000 Class
Object Detection/Class DB

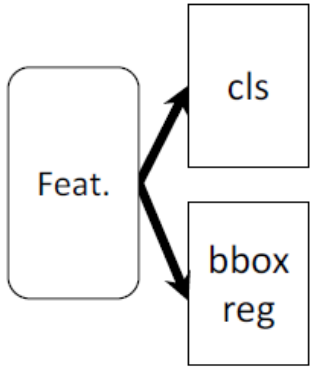
http://visualgenome.org/



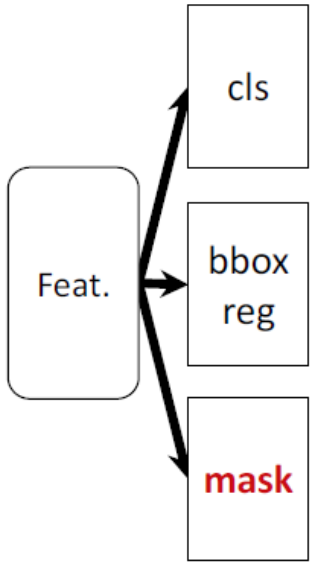
from Mask R-CNN

More about Mask R-CNN: 이번 주 일요일 Youtube PR12





Fast/er R-CNN

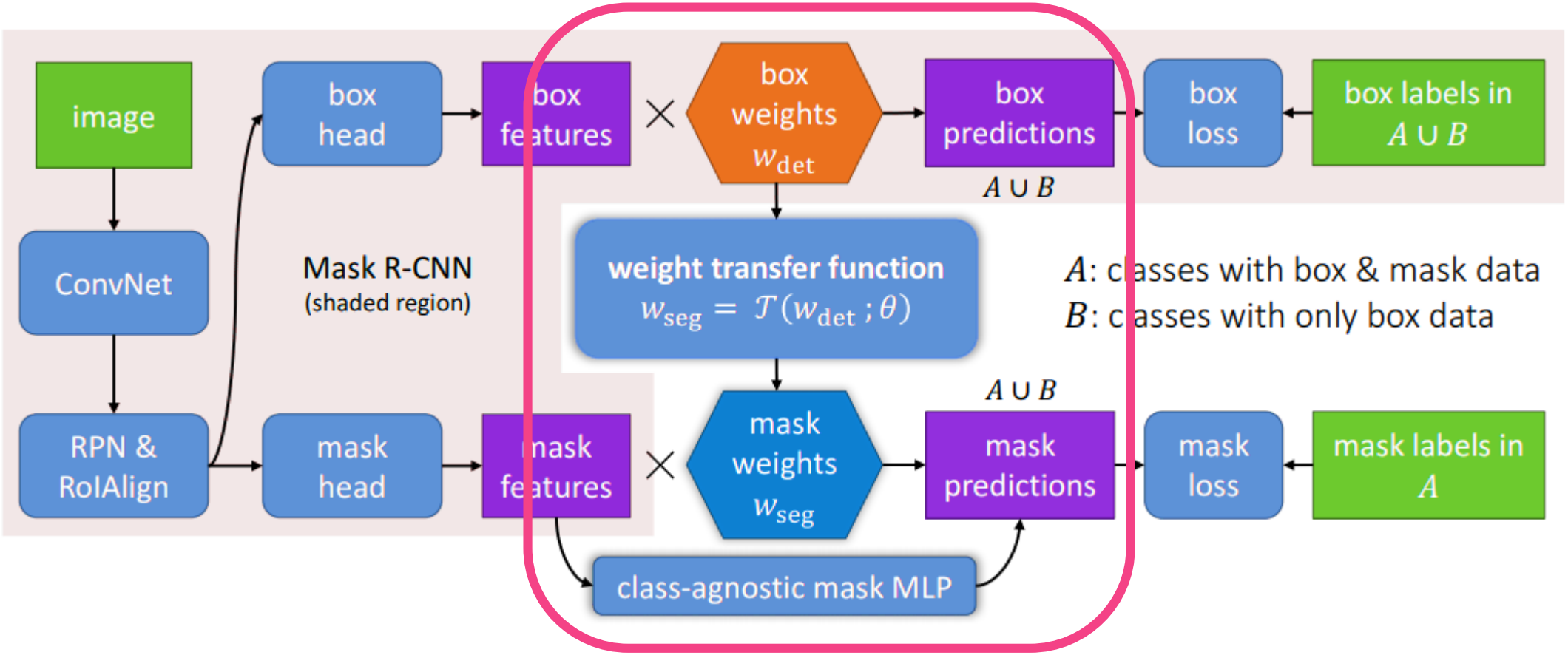


Mask R-CNN

Mask R-CNN: Faster R-CNN + FCN (in the Bounding Box)

Re-training Mask R-CNN

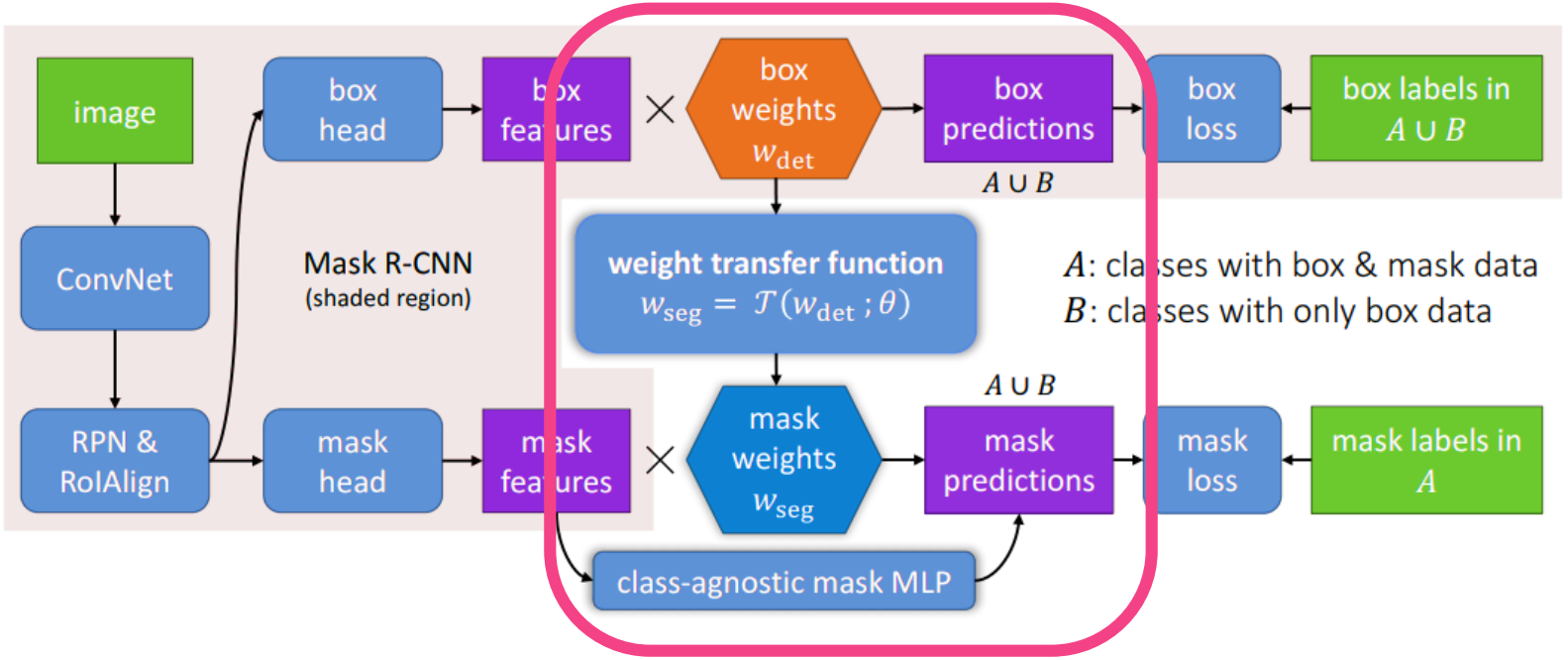
Segmentation DB를 학습할 때 Transfer Function을 학습



Re-training Mask R-CNN

Segmentation DB를 학습할 때 Transfer Function을 학습

- Segmentation DB를 통해서
- BBox에서 Segmentation으로 가는 "관계 " 를 학습
- Label Classification은 다른 Head로부터 학습



Mask X R-CNN Results

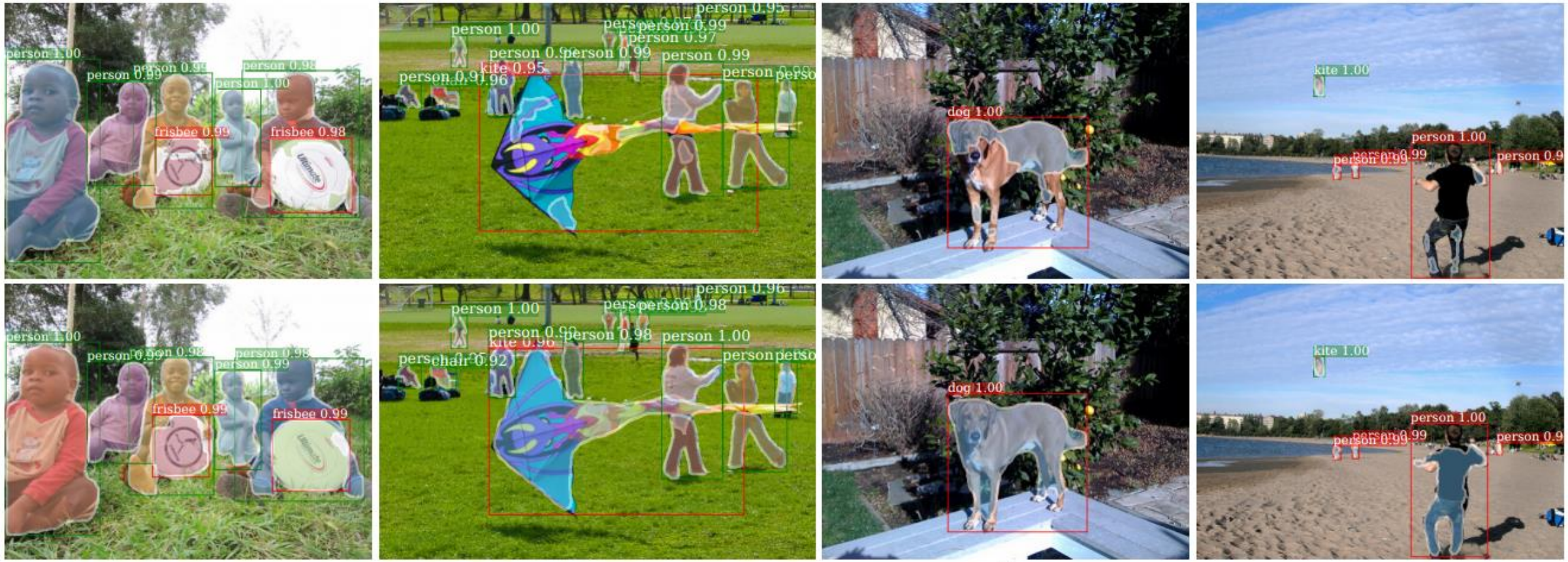
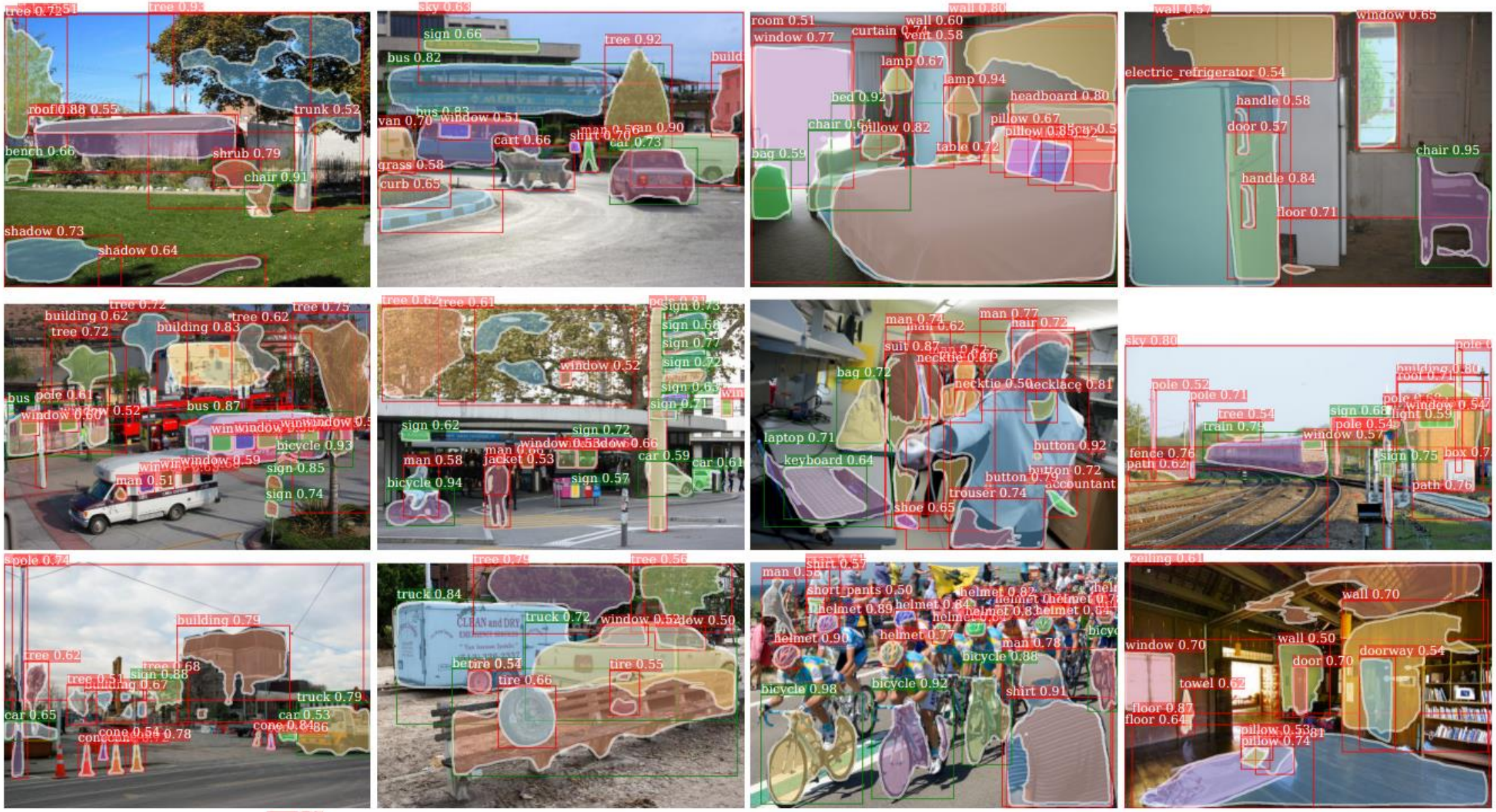


Figure 4. **Mask predictions from the class-agnostic baseline (top row) *vs*. our Mask^X R-CNN approach (bottom row).** Green boxes are classes in set A while the red boxes are classes in set B. The left 2 columns are $A = \{\text{voc}\}$ and the right 2 columns are $A = \{\text{non-voc}\}$.

Mask X R-CNN Results



Conclusion 1: CAM

완벽한 Network
완벽한 Algorithm
완벽한 Dataset은 없다

결국 개선을 위해서는 Handcrafted된 Debugging이 필요
Deep Learning의 “해석”이 중요하고, 가능하다.

Conclusion 2: Weakly Supervised Learning

결국 알고리즘만큼
문제를 정의하는것이 중요하다

DB가 적을 경우 / 없을 경우 / (잘못되었을 경우)
완벽한 학습을 위해서는?
제약조건은 무엇인가?

감사합니다