Tell Me Where to Look: Guided Attention Inference Network

Kunpeng Li, Zian Wu, Kuan-Chuan Peng, Jan Ernst, Yun Fu

CVPR 2018

임희주

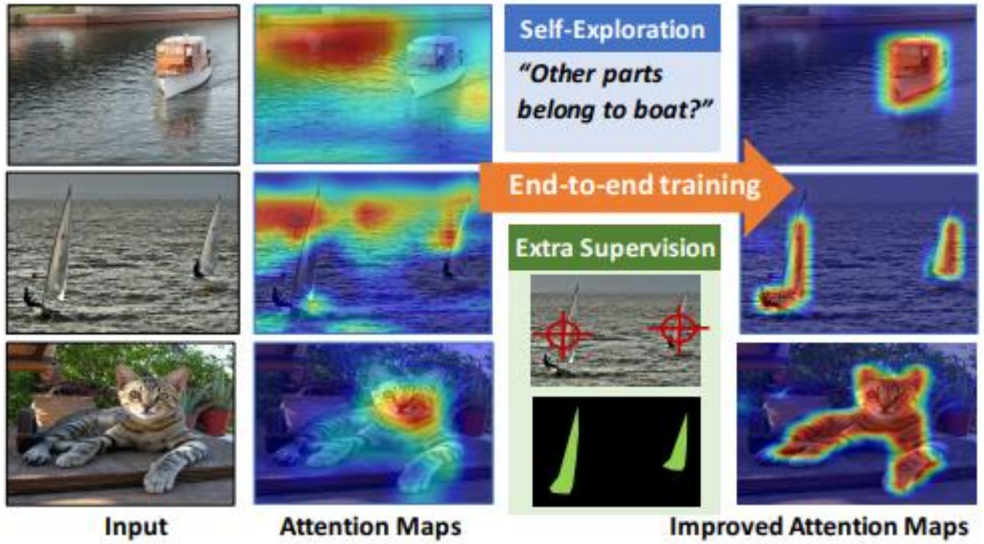
Index

Introduction

Introduction

Computer Vision 분야에서 신경망은 어떤 패턴에 집중(attention)하여 물체 인식

- 그러나 정확한 Attention 수행이 되지 않는 경우 존재
- Attention 성능 향상을 위해 **Guided Attention Inference Network** 제안



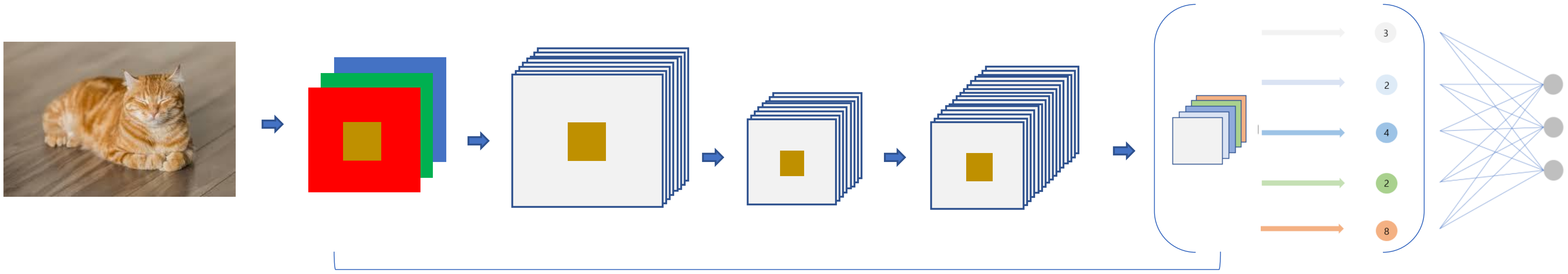
Ex) 첫번째 이미지에서 Attention Map 에 boat가 아닌 물에 attention 영역 존재

Related work

Related work

: CAM

CAM(Class Activation Map)
- CNN + GAP (Global Average Pooling)
- CNN을 통해 이미지 정보 요약 후 GAP 사용

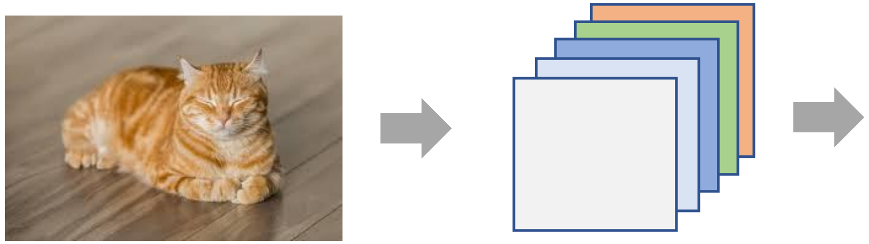


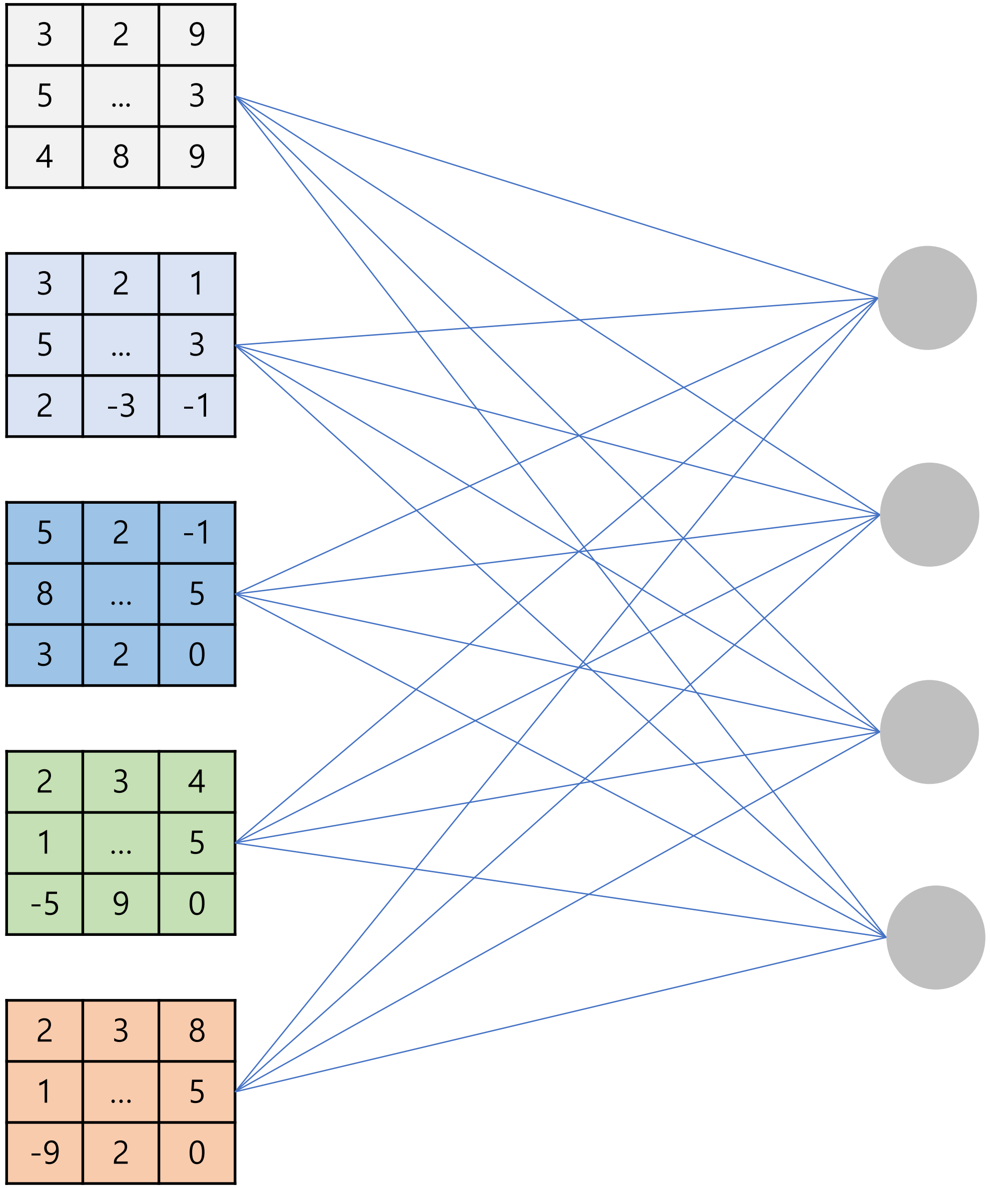
기존 CNN 구조 + Global Average Pooling

Related work

: CAM

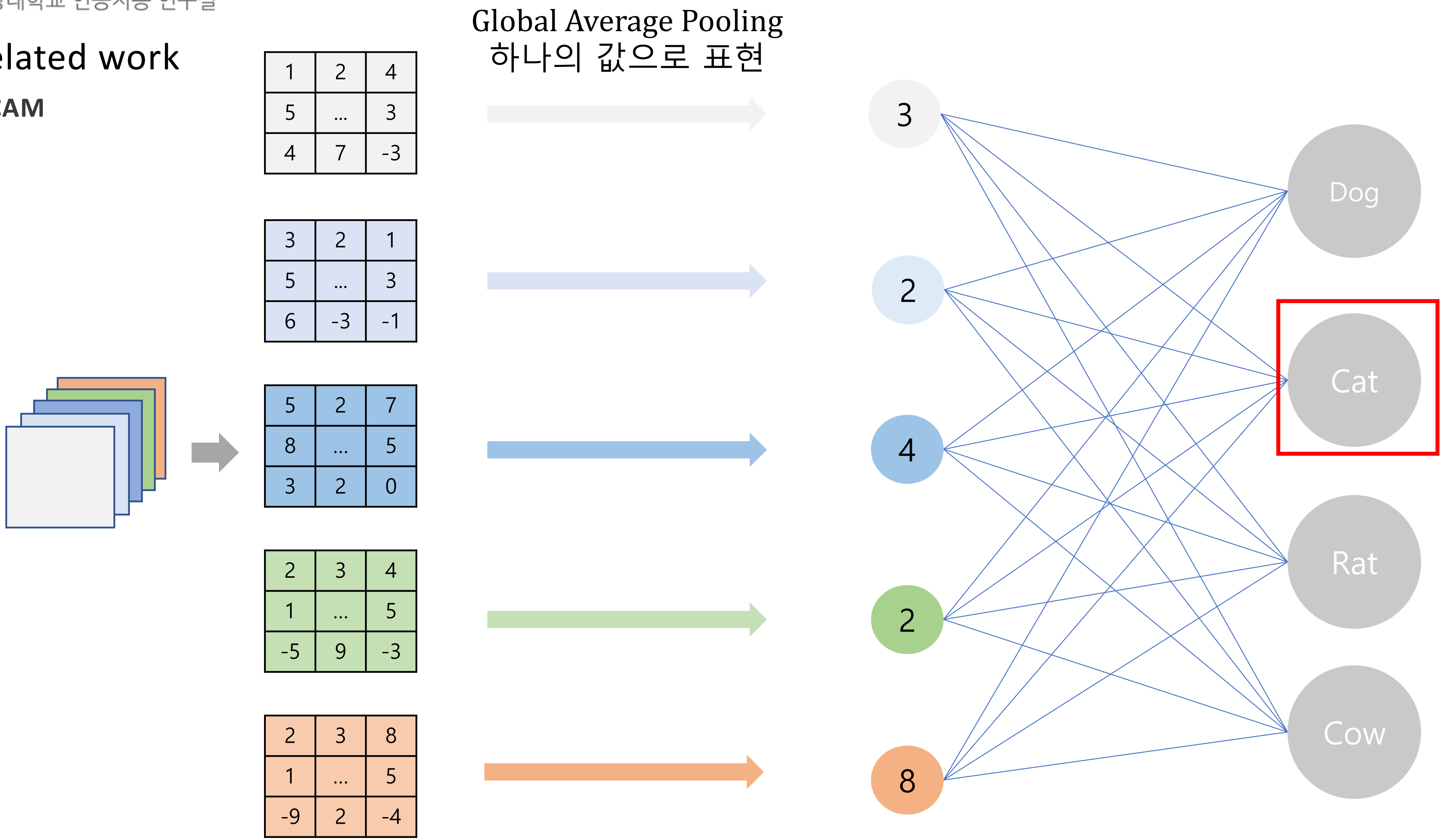
CNN Classification
- 이미지에서 feature map생성
- 이후 FC layer 를 거치고 Classification





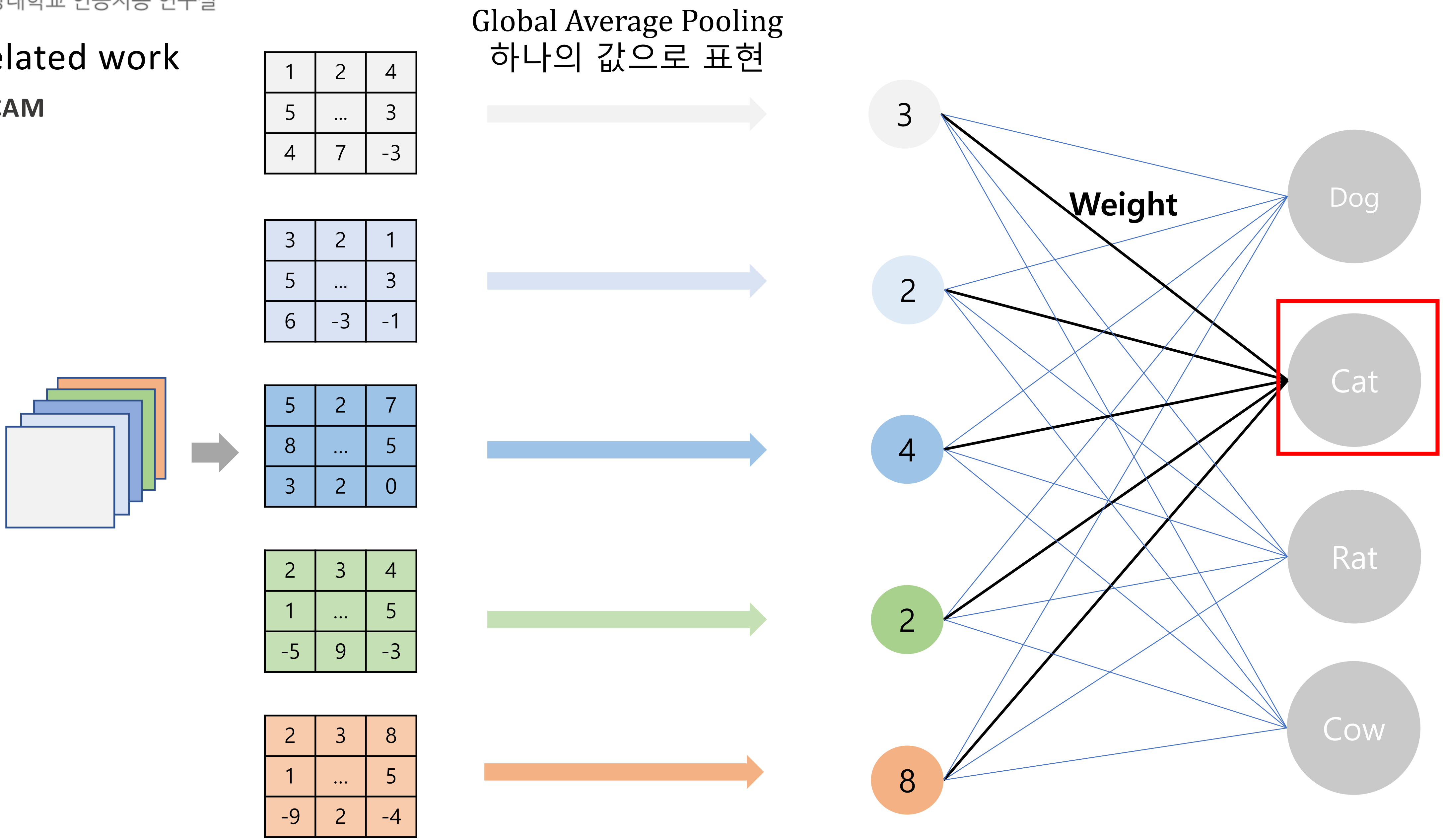
Related work

: CAM



Related work

: CAM

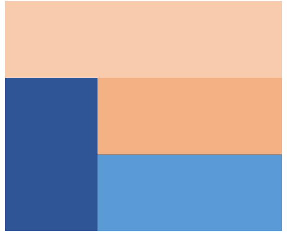


Related work

: CAM

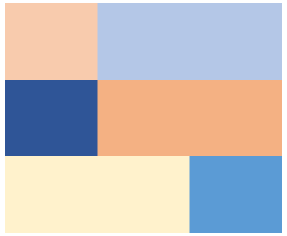
1	2	4
5	...	3
4	7	-3

x **Weight1** =



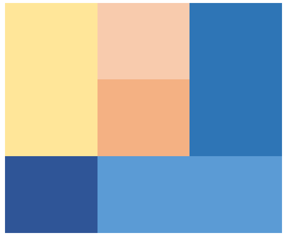
3	2	1
5	...	3
6	-3	-1

x **Weight2** =



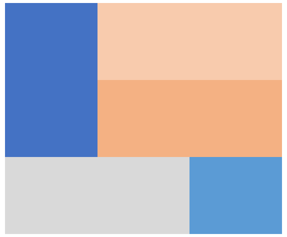
5	2	7
8	...	5
3	2	0

x **Weight3** =



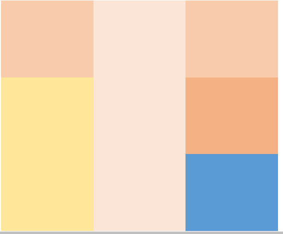
2	3	4
1	...	5
-5	9	-3

x **Weight4** =



2	3	8
1	...	5
-9	2	-4

x **Weight5** =



[CAM]
C class 에 대한 Score
(attention map)

$$S^c = \sum_k W_k^c \frac{1}{Z} \sum_i \sum_j F_{i,j}^k$$

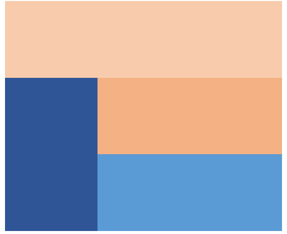
c = 예측 Class
W_k^c = c Class를 예측하는 k 번째 Feature Map 에 대한 weight
F^k = k번째 Feature Map
$F_{i,j}^k$ = Feature Map 내 i, j 위치 값
Z = 각 Feature Map의 합

Related work

: CAM

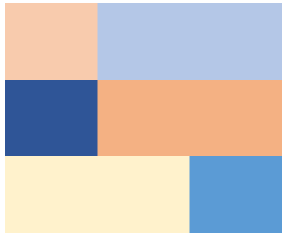
1	2	4
5	…	3
4	7	-3

x **Weight1** =



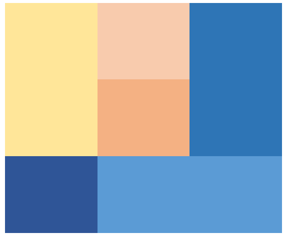
3	2	1
5	…	3
6	-3	-1

x **Weight2** =



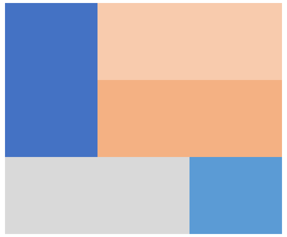
5	2	7
8	…	5
3	2	0

x **Weight3** =



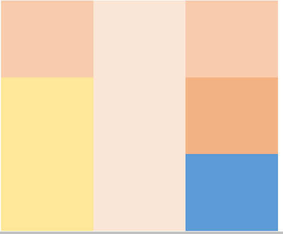
2	3	4
1	…	5
-5	9	-3

x **Weight4** =



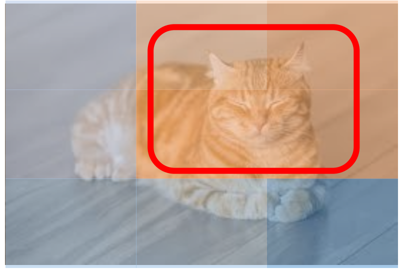
2	3	8
1	…	5
-9	2	-4

x **Weight5** =



[CAM]
C class 에 대한 Score
(attention map)

$$S^c = \sum_k W_k^c \frac{1}{Z} \sum_i \sum_j F_{i,j}^k$$



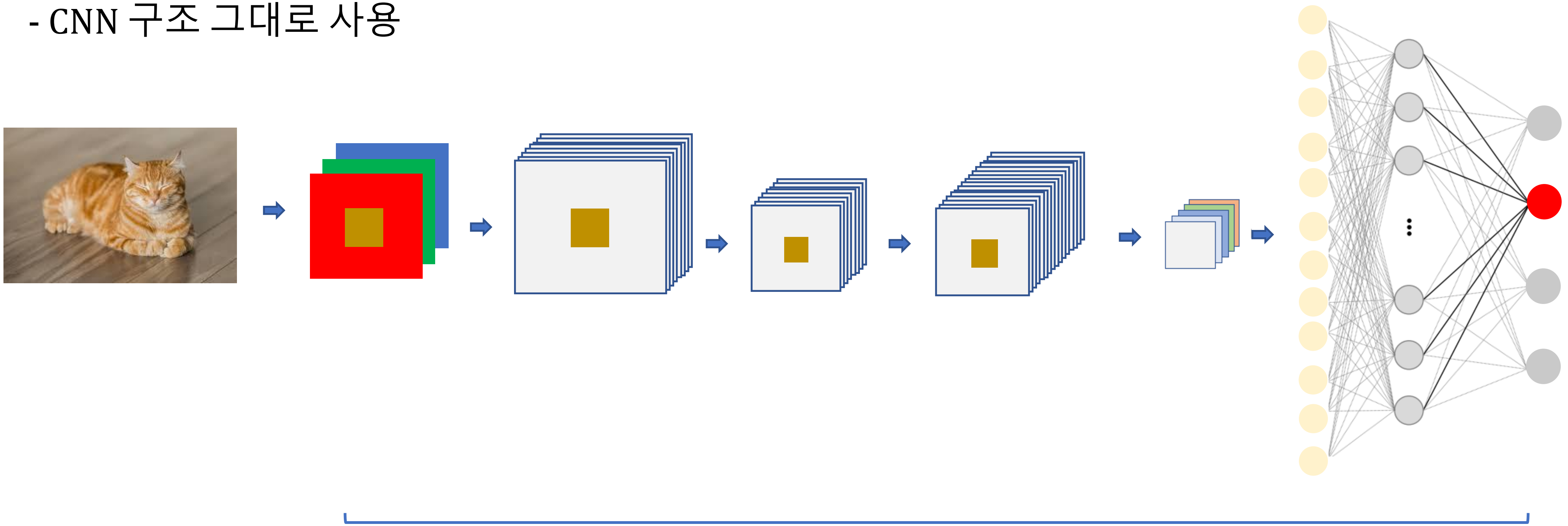
고양이의 얼굴을
예측 원인으로 판단

Related work

: Grad-CAM

Grad-CAM(Gradient-weighted CAM)
- CAM 을 generalize 한 버전
- CNN 구조 그대로 사용

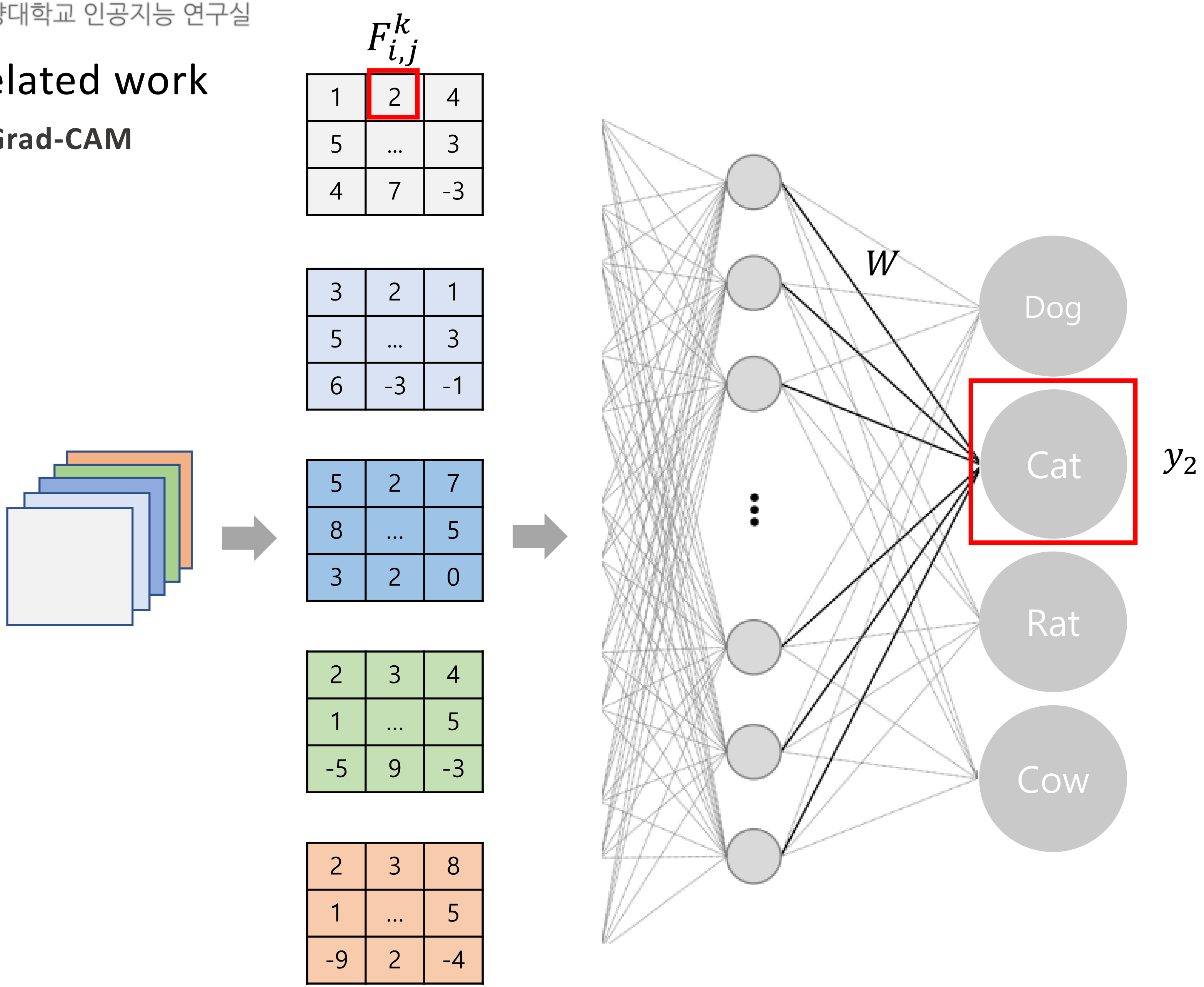
~~+ Global Average Pooling~~



기존 CNN 구조

Related work

: Grad-CAM



Global average pooling 대신
일반적인 CNN 구조 사용

$$S_{Grad\_CAM}^{c} = ReLU \sum_{k} f_{k}^{c} F^{k}$$

$$f_{k}^{c} = \frac{1}{Z} \sum_{i} \sum_{j} \frac{\partial y_{c}}{\partial F_{i,j}^{k}}$$

F^{k} = k 번째 Feature map
y_{c} $= Wx + b$
$F_{i,j}^{k}$ = Feature Map 내 i, j 위치 값

Related work

: Grad-CAM

F_k^c

1	2	4
5	...	3
4	7	-3

x $\frac{1}{Z}\sum_i \sum_j \frac{\partial y_c}{\partial F_{i,j}^k}$ =

3	2	1
5	...	3
6	-3	-1

x $\frac{1}{Z}\sum_i \sum_j \frac{\partial y_c}{\partial F_{i,j}^k}$ =

5	2	7
8	...	5
3	2	0

x $\frac{1}{Z}\sum_i \sum_j \frac{\partial y_c}{\partial F_{i,j}^k}$ =

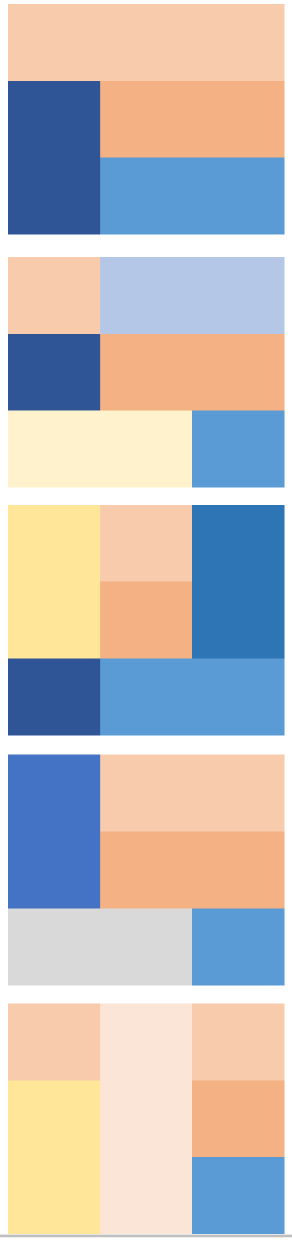
2	3	4
1	...	5
-5	9	-3

x $\frac{1}{Z}\sum_i \sum_j \frac{\partial y_c}{\partial F_{i,j}^k}$ =

2	3	8
1	...	5
-9	2	-4

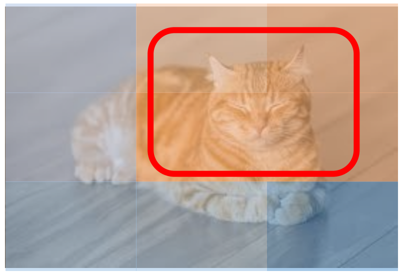
x $\frac{1}{Z}\sum_i \sum_j \frac{\partial y_c}{\partial F_{i,j}^k}$ =

Feature heat map



$$S_{Grad\_CAM}^c = ReLU \sum_k f_k^c F^k$$

$$f_k^c = \frac{1}{Z}\sum_i \sum_j \frac{\partial y_c}{\partial F_{i,j}^k}$$



고양이의 얼굴을
예측 원인으로 판단

Methodology

Methodology

: GAIN

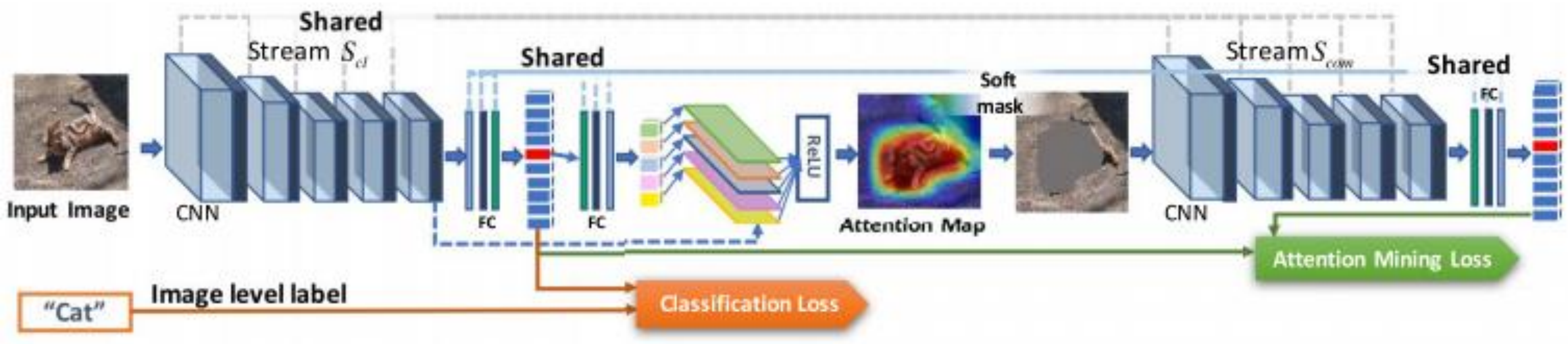
Guided Attention Inference Network

Classification 한 결과를 가지고 스스로 attention 영역을 재 학습하는 구조 (self-guidance)

Methodology

: GAIN

Guided Attention Inference Network
- 두개의 stream S_{cl}, S_{am}

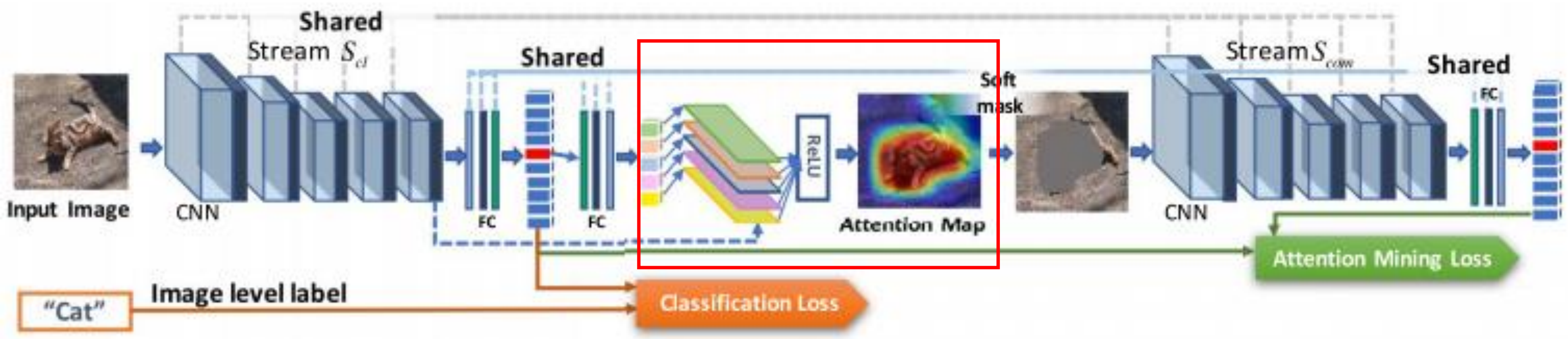


S_{cl} : Class 를 인식하는 region을 찾아내는 stream
S_{am} : Classification 에 영향을 주는 모든 영역에 attention이 되도록 하는 stream

Methodology

: GAIN

Guided Attention Inference Network
- Grad-CAM 을 통해 얻은 Feature map A 와 original input image 사용



$$w^c_{l,k} = \mathrm{GAP}\left(\frac{\partial y^c}{\partial f_{l,k}}\right)$$
$$A^c = \mathrm{ReLU}\left(\mathrm{conv}\left(f_l, w^c\right)\right)$$

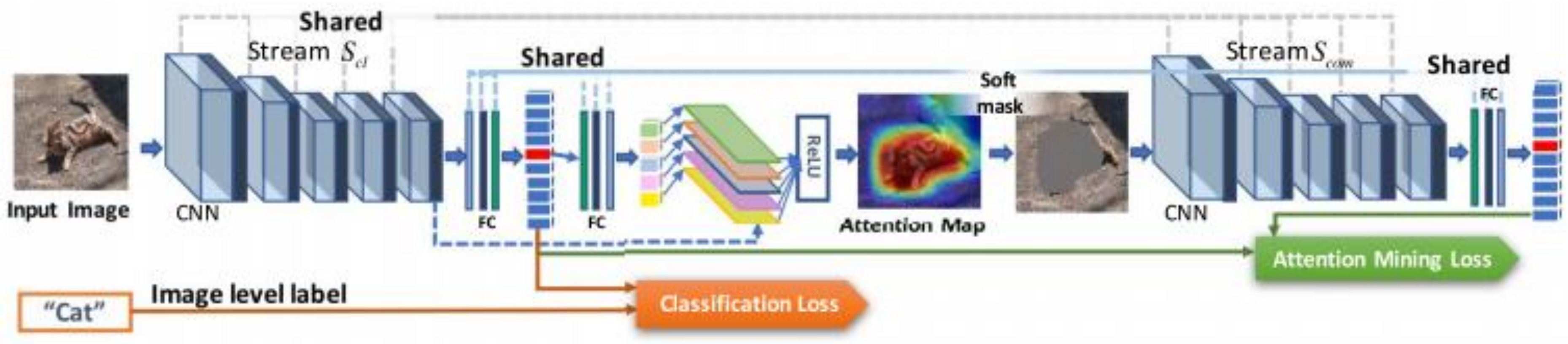
By Grad-CAM

Feautre Map 과 각 요소별 gradient 의 Global Average Pooling 한 값을
Conv 하여 Attention Map 획득
ReLU 의 의미 : 같은 방향으로의 변화만을 취급
Conv(pos, pos) or Conv(neg, neg)

Methodology

: GAIN

Guided Attention Inference Network
- Modified sigmoid func. T



$$T(A^c) = \frac{1}{1+\exp(-\omega(A^c - \boldsymbol{\sigma}))}$$

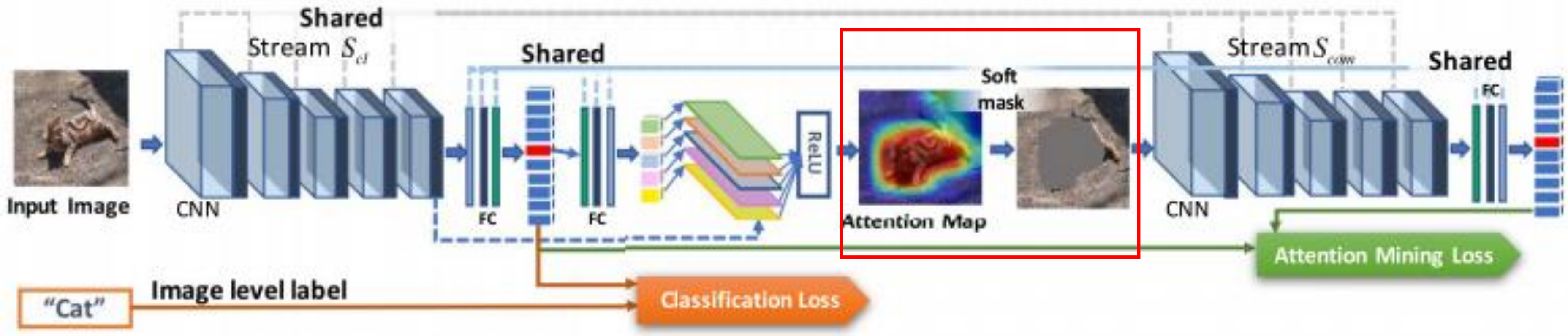
T의 역할 : sigmoid 를 더 가파르게 만들기 위함
(1일 수록 더 1에 가깝게, 0일수록 더 0에 가깝게)

$I^{*c} = S_{am}$ 에 사용하기위해 soft-masked 된 Image
T = modified sigmoid function
ω = scale parameter
$\boldsymbol{\sigma}$ = 모든 요소 값이 같은 threshold matrix
$\odot$ = element-wise multiplication

Methodology

: GAIN

Guided Attention Inference Network
- Masked 된 Residual Image I^{*c}



$$T\left(A^{c}\right)=\frac{1}{1+\exp\left(-\omega\left(A^{c}-\boldsymbol{\sigma}\right)\right)}$$

$$I^{*c}=I-\left(T\left(A^{c}\right)\odot I\right)$$

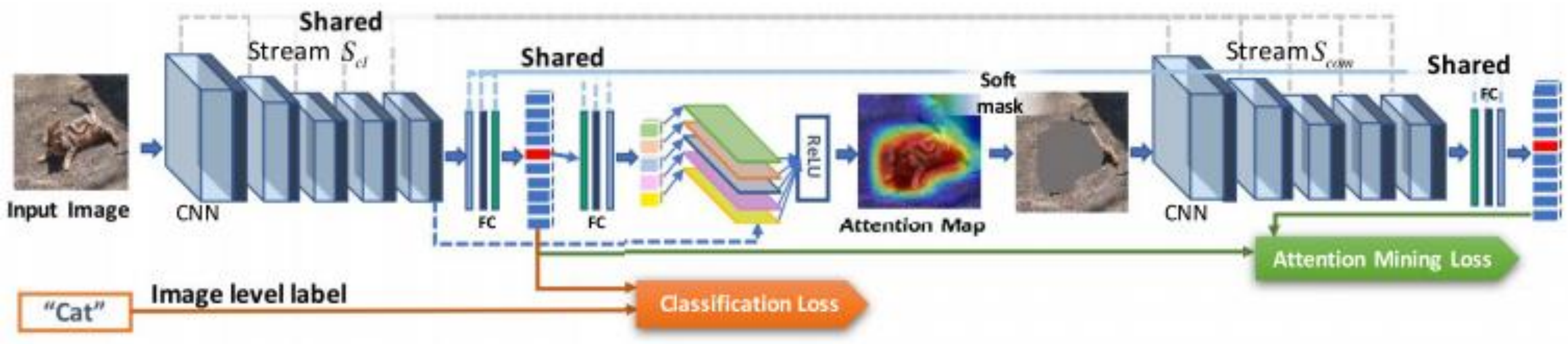
I^{*c} = S_{am} 에 사용하기위해 soft-masked 된 Image
T = modified sigmoid function
ω = scale parameter
$\boldsymbol{\sigma}$ = 모든 요소 값이 같은 threshold matrix
$\odot$ = element-wise multiplication

이미지에서 attention 영역을 제거
I = Original image Feature Map
$T(A^c)$ = Attention 영역이면 1, 아니면 0 return

Methodology

: GAIN

Guided Attention Inference Network
- GAIN loss function



$$L_{am} = \frac{1}{n}\sum_{c} (I^{*c})$$

$$L_{self} = L_{cl} + \alpha L_{am}$$

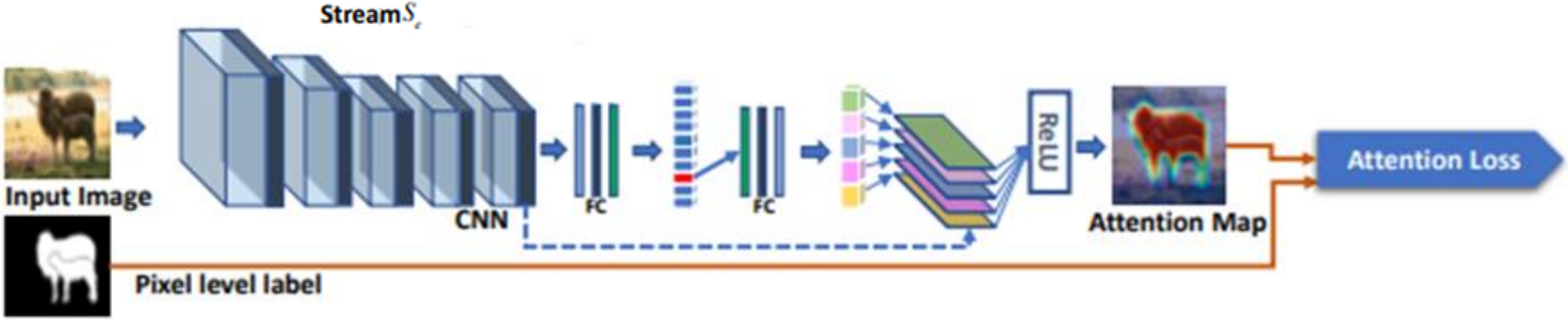
L_{cl} : Classification Loss
L_{am} : Attention Mining Loss
L_{self} : Self-guidance Loss
α : weighting parameter (fixed 1)

Methodology

: GAINext

Guided Attention Inference Network – External Version

- External stream S_e 추가
- S_{cl}, S_{am}, S_e 가 모든 파라미터를 공유

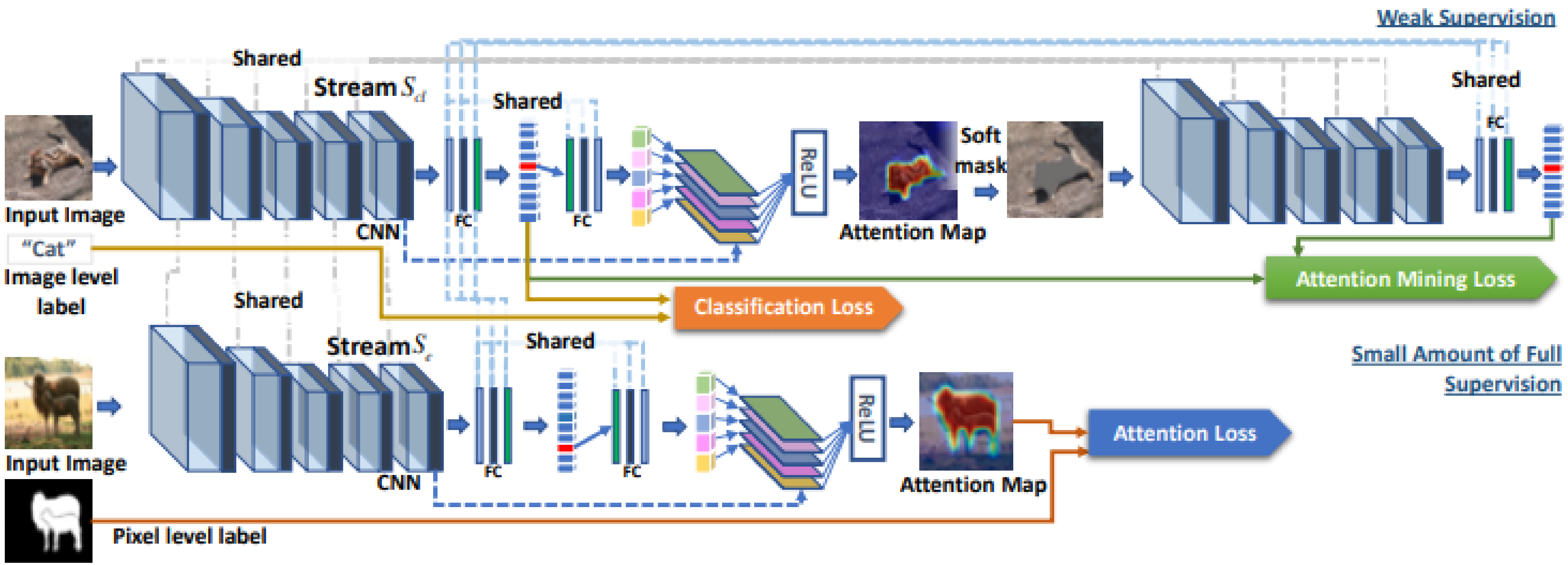


$$L_e = \frac{1}{n}\sum_c (A^c - H^c)^2$$

H_c = pixel level label

Methodology

: GAINext Framework



$$L_{ext} = L_{cl} + \alpha L_{am} + \omega L_e$$

L_{cl} : Classification Loss
L_{am} : Attention Mining Loss
L_{self} : Self-guidance Loss
α : weighting parameter (fixed to 1)
ω : weighting parameter (fixed to 10)

Experiment

Experiment

Results

Methods	Training Set	*val.* (mIoU)	*test* (mIoU)
Supervision: Purely Image-level Labels			
CCNN [19]	10K weak	35.3	35.6
MIL-sppxl [20]	700K weak	35.8	36.6
EM-Adapt [18]	10K weak	38.2	39.6
DCSM [25]	10K weak	44.1	45.1
BFBP [23]	10K weak	46.6	48.0
STC [32]	50K weak	49.8	51.2
AF-SS [21]	10K weak	52.6	52.7
CBTS-cues [22]	10K weak	52.8	53.7
TPL [11]	10K weak	53.1	53.8
AE-PSL [31]	10K weak	55.0	55.7
SEC [12] (baseline)	10K weak	50.7	51.7
GAIN (ours)	10K weak	**55.3**	**56.8**
Supervision: Image-level Labels (* Implicitly use pixel-level supervision)			
MIL-seg* [20]	700K weak + 1464 pixel	40.6	42.0
TransferNet* [9]	27K weak + 17K pixel	51.2	52.1
AF-MCG* [21]	10K weak + 1464 pixel	54.3	55.5
GAIN$_{ext}$* (ours)	10K weak + 200 pixel	**58.3**	**59.6**
GAIN$_{ext}$* (ours)	10K weak + 1464 pixel	**60.5**	**62.1**

Semantic segmentation experiments

PASCAL VOC 2012 segmentation set

mIoU : mean Intersection over Union

Experiment

Results

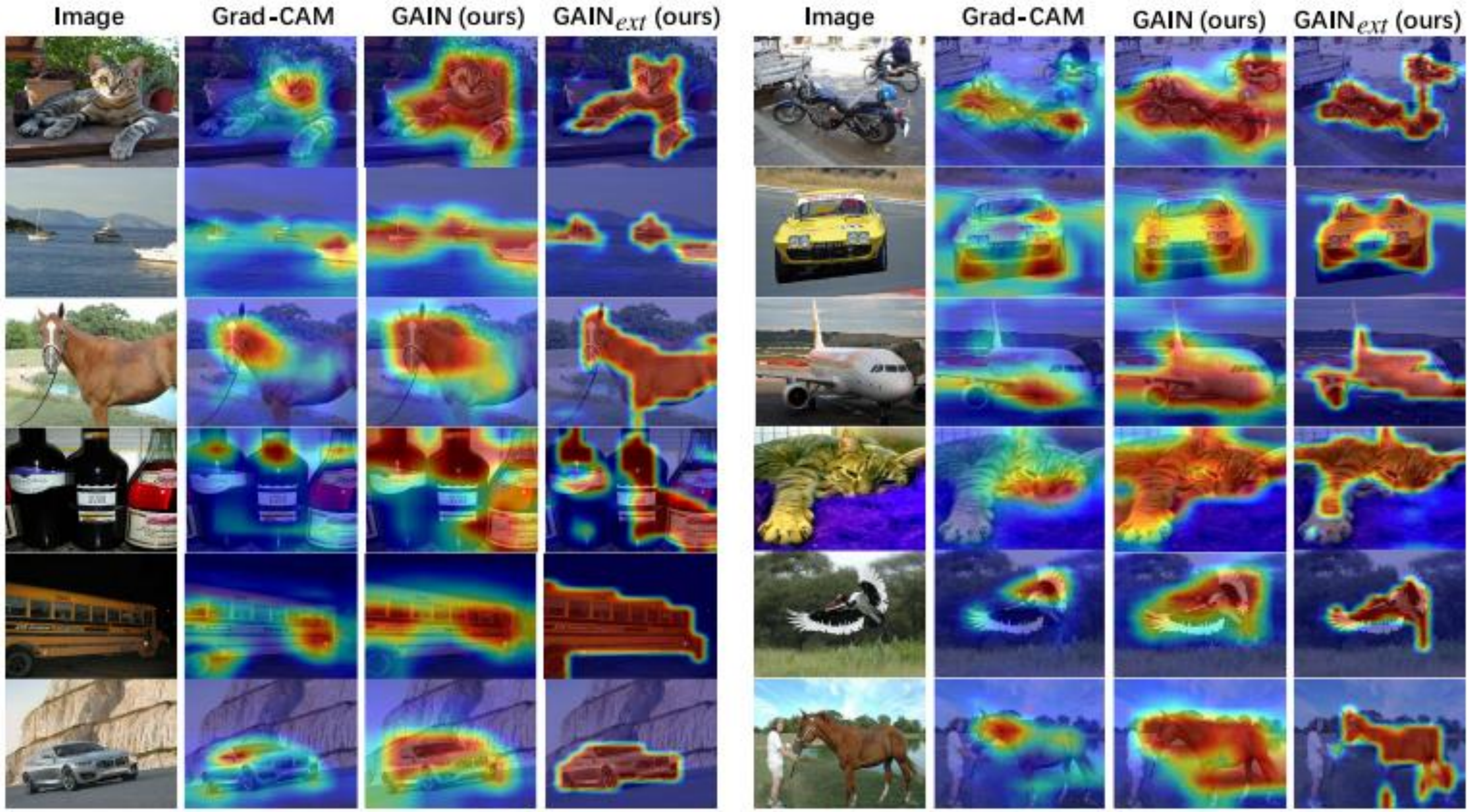


Figure 5. Qualitative results of attention maps generated by Grad-CAM [24], our GAIN and GAIN$_{ext}$ using 200 randomly selected (2%) extra supervision.

Experiment

Results

Tested on author's biased boat

Figure 6. Qualitative results generated by Grad-CAM [24], our GAIN and GAIN$_{ext}$ on our *biased boat* dataset. All the methods are trained on Pascal VOC 2012 dataset. -# denotes the number of pixel-level labels of *boat* used in the training which were randomly chosen from VOC 2012. Attention map corresponding to *boat* shown only when the prediction is positive (i.e. test image contains *boat*).

Experiment

Results

Tested on author's biased boat

Test set	Grad-CAM	GAIN	GAIN_{ext} (# of PL) 9	23	78
VOC val.	83%	90%	93%	93%	94%
Boat without water	42%	48%	64%	74%	84%
Water without boat	30%	62%	68%	76%	84%
Overall	36%	55%	66%	75%	84%

Conclusion & Discussion

Conclusion

Self-guidance, supervision 구조의 신경망으로 attention map을 더 잘 만드는 framework 제안
-> 발표당시 segmentation SOTA 성능

Contribution

- Attention map에 적용되는 지도 학습 방법 제안
- 신경망이 이미지 전체적으로 attention 을 가질 수 있도록 하는 self-guidance in training 제안
- 하나의 Framework 에서 Full supervision 이 원활하도록 supervision 과 self-guidance 를 잘 통합

Thanks

Appendix

Appendix

references

- **Paper**
  - https://openaccess.thecvf.com/content_cvpr_2018/papers/Li_Tell_Me_Where_CVPR_2018_paper.pdf

- **Etc.**
  - https://www.youtube.com/watch?v=fFyv1wCN4DU
  - https://github.com/HYU-AILAB/ai-seminar/blob/master/season_13/07.%20A%20Graph%20Convolutional%20Neural%20Network%20for%20Emotion%20Recognition%20in%20Conversation/200831_DialogueGCN_Yuri.pdf
  - https://github.com/chullhwan-song/Reading-Paper/issues/11