FickleNet: Weakly and Semi-supervised Semantic Image Segmentation using Stochastic Inference

Jungbeom Lee Eunji Kim Sungmin Lee Jangho Lee Sungroh Yoon†
Department of Electrical and Computer Engineering, Seoul National University, Seoul, South Korea
{jbeom.lee93, kce407, simonlee0810, ubuntu, sryoon}@snu.ac.kr

김성철

Contents

Introduction

- **Pixel-level annotation**

  → Fully supervised semantic segmentation

- **Image-level annotation으로 segmentation network 학습은 어려움!**

  → Weakly labeled data는 class의 존재 유무만 가리키기 때문!

  → Classification network에서 얻어진 localization map에 의존

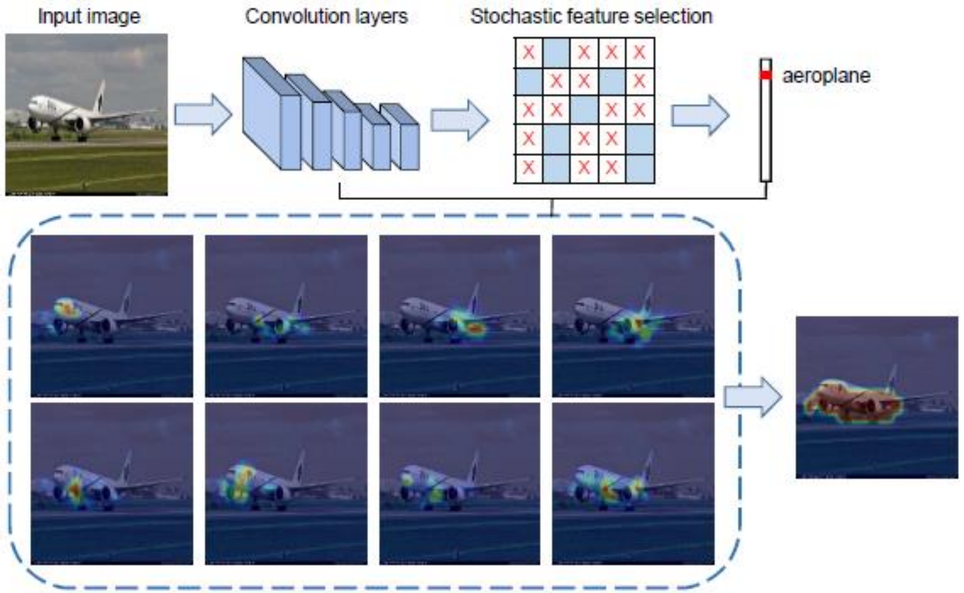
  → Localization map도 정확한 경계에 대한 표현은 없고, small discriminative part에 집중

Introduction

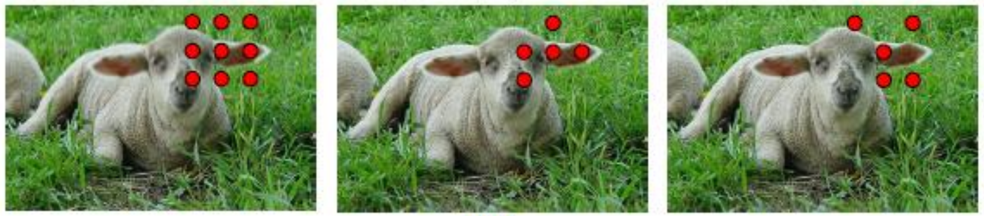
- **FickleNet**

→ CNN의 hidden unit의 **random combination**을 이용해 **여러 가지 localization map 생성**

→ 각 sliding window position에서 **random으로 hidden unit**을 선택



(a)



(b)

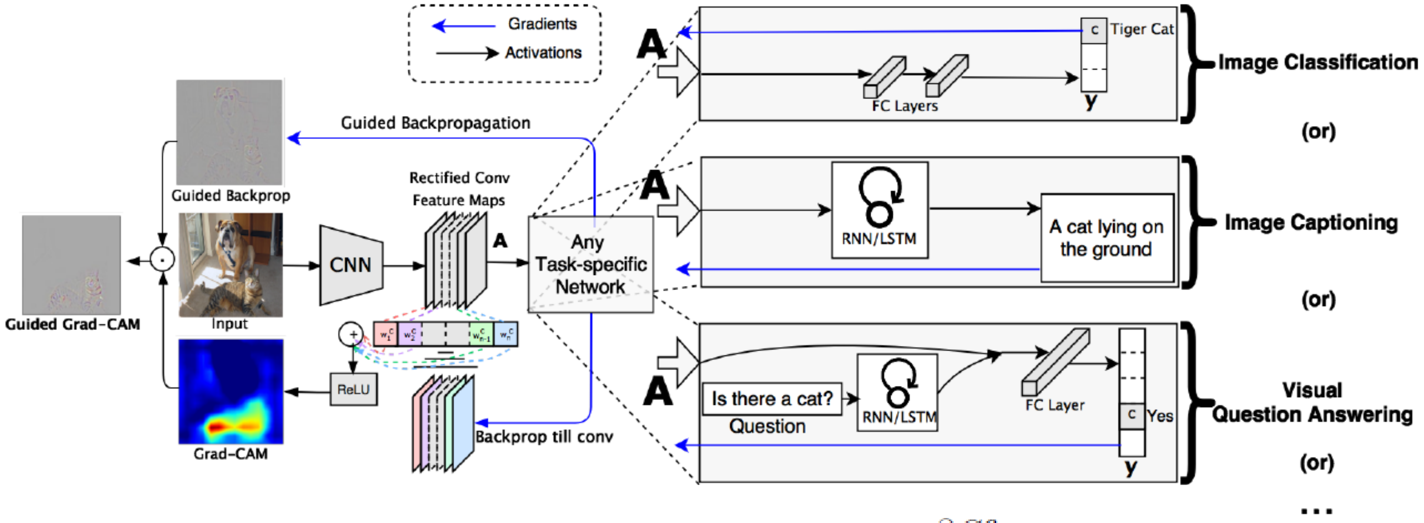
Figure 1. (a) FickleNet allows a single network to generate multiple localization maps from a single image. (b) Conceptual description of hidden unit selection. Selecting all hidden units (deterministic, *left*) produces smoothing effects as background and foreground are activated together. Randomly selected hidden units (stochastic, *center* and *right*) can provide more flexible combinations which can correspond more clearly to parts of objects or the background.

Related Work

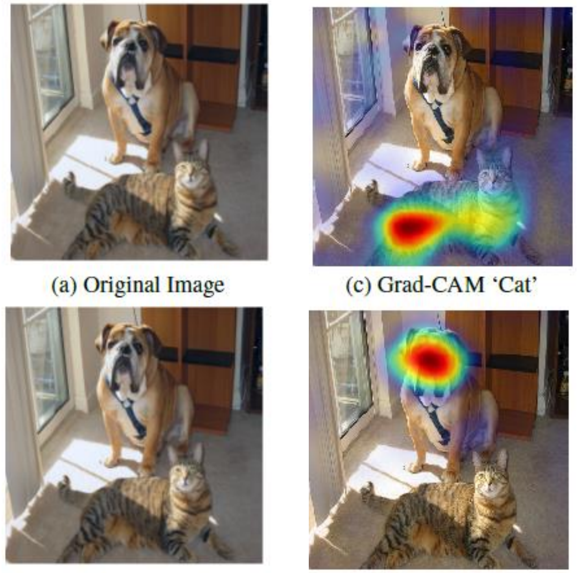
- **Grad-CAM**

  → Image-level annotation으로 pixel을 분류

  <u>→ Object의 small discriminative region에 집중</u>



$$\text{Grad-CAM}^c = \text{ReLU}\left(\sum_k x_k \times \frac{\partial S^c}{\partial x_k}\right), \qquad (1)$$



(a) Original Image (c) Grad-CAM 'Cat'

(g) Original Image (i) Grad-CAM 'Dog'

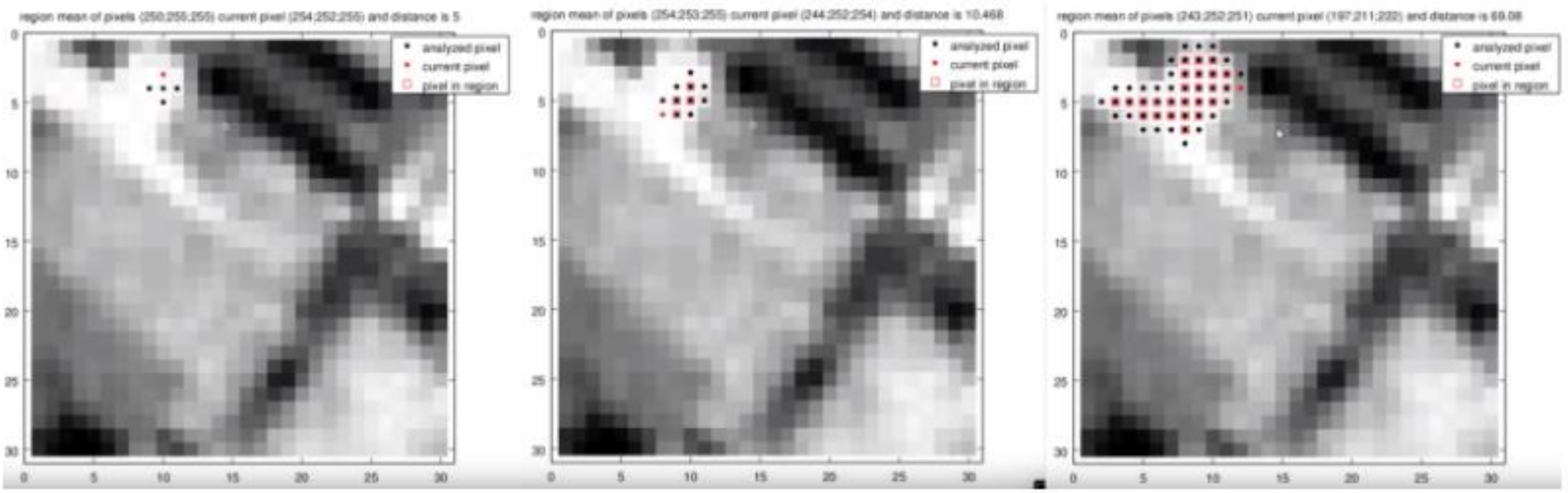
https://arxiv.org/abs/1610.02391

Related Work

- **SRG (Seeded Region Growing)**

  → Seeded cues와 이미지로 Region Growing

  → Pixel level을 기준으로 clustering



https://blog.lunit.io/2019/04/25/ficklenetweakly-and-semi-supervised-semantic-image-segmentation-using-stochastic-inference/

Related Work

- **DSRG (Deep Seeded Region Growing)**

→ Seeded cues와 CAM으로 Region Growing, Segmentation training

→ <u>**Seeding loss**</u> + Fully connected CRF와의 <u>**Boundary loss**</u>

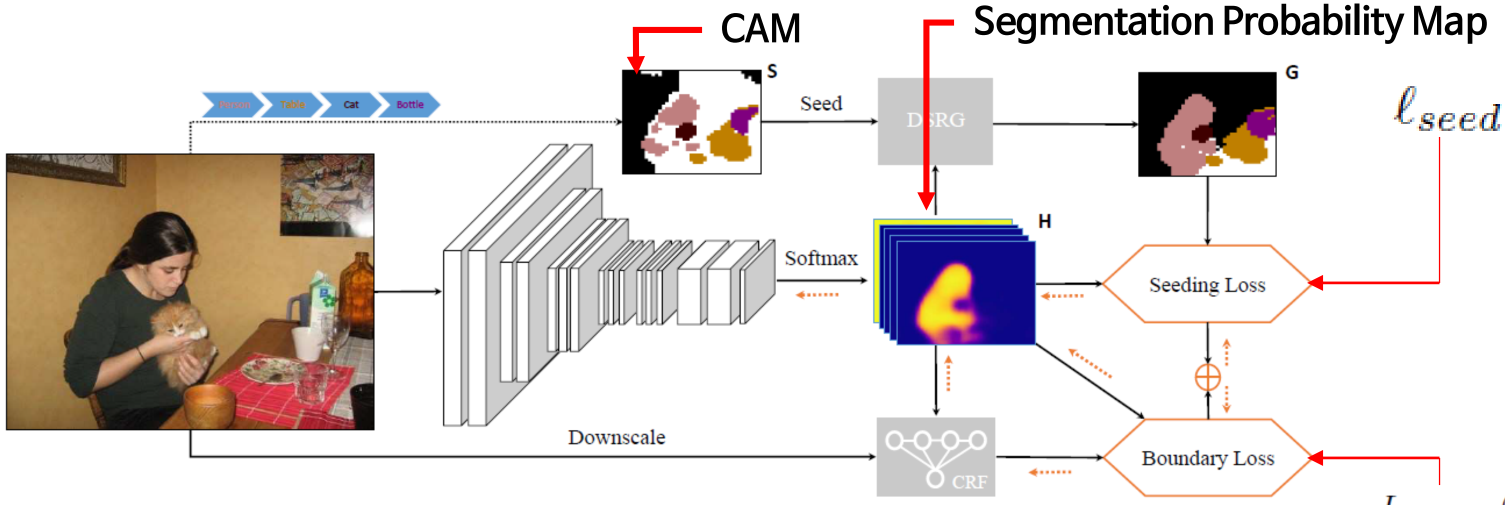


Figure 2. Overview of the proposed Deep Seeded Region Growing training approach. The Region Growing module takes the seed cues and segmentation map as input produces latent pixel-wise supervision which is more accurate and more complete than seed cues. Our method iterates between refining pixel-wise supervision and optimizing the parameters of a segmentation network.

$$\ell_{seed} = -\frac{1}{\sum_{c\in\mathcal{C}}|S_c|}\sum_{c\in\mathcal{C}}\sum_{u\in S_c}\log H_{u,c} - \frac{1}{\sum_{c\in\bar{\mathcal{C}}}|S_c|}\sum_{c\in\bar{\mathcal{C}}}\sum_{u\in S_c}\log H_{u,c},$$

$$L_{\text{constrain}}(X, f(X)) = \frac{1}{n}\sum_{u=1}^{n}\sum_{c\in\mathcal{C}} Q_{u,c}(X, f(X))\log\frac{Q_{u,c}(X, f(X))}{f_{u,c}(X)}.$$

$\boldsymbol{H_{u,c}}$

https://openaccess.thecvf.com/content_cvpr_2018/papers/Huang_Weakly-Supervised_Semantic_Segmentation_CVPR_2018_paper.pdf

Related Work

- DSRG (Deep Seeded Region Growing)

https://openaccess.thecvf.com/content_cvpr_2018/papers/Huang_Weakly-Supervised_Semantic_Segmentation_CVPR_2018_paper.pdf

Proposed Method

1. Hidden unit의 stochastic selection 사용, multi-class classification 학습
2. Training image의 localization map 생성
3. Localization map을 segmentation network 학습에 pseudo-label로 사용

Algorithm 1: Training and Inference Procedure

Input: Image I, ground-truth label c, dropout rate p
Output: Classification score S and localization maps M

x = Forward(I) until conv5 layer;
Stochastic hidden unit selection: Sec. 3.1
  x^{expand} = Expand(x); Sec. 3.1.1
  x_p^{expand} = Center-fixed spatial dropout(x^{expand}, p); Sec. 3.1.2
  S = Classifier(x_p^{expand}); Sec. 3.1.3
Training Classifier:
  Update network by L=SigmoidCrossEntropy(S, c)

①

Inference CAMs: Sec. 3.2
  For different random selections i ($1 \le i \le N$):
    $M^c[i]$ = Grad-CAM(x, S^c); Sec. 3.2.1
  M^c = Aggregate($M^c[i]$); Sec. 3.2.2

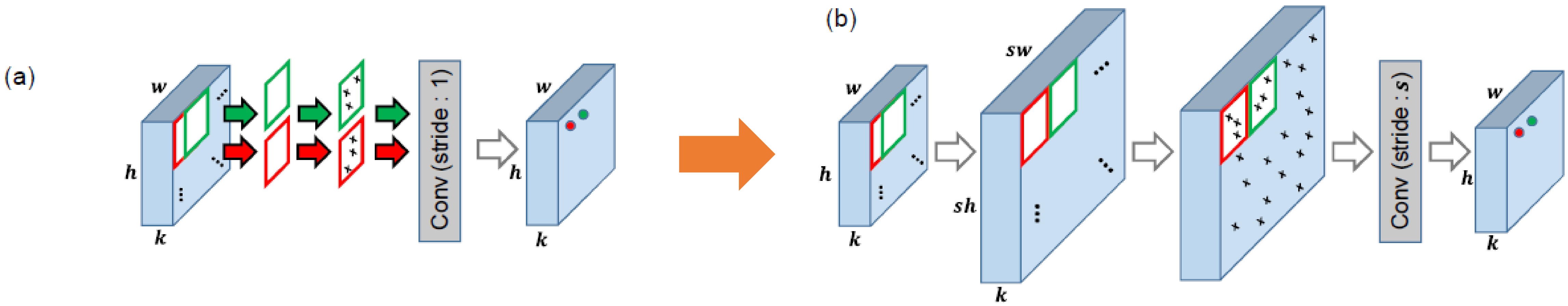
②

Proposed Method

1. Stochastic Hidden Unit Selection

① Feature Map Expansion

- GPU 활용을 극대화하기 위해 localization map을 stride = convolutional kernel size가 되도록 확장
- 한 번에 random selection을 시행하여 빠른 연산 가능



Proposed Method

1. Stochastic Hidden Unit Selection

② Center-preserving Spatial Dropout

- 각 sliding window position의 kernel 중심은 drop하지 않음

→ 중심과 다른 위치와의 관계 발견 가능!

- Training과 inference 모두 dropout 적용

③ Classification

- Global Average Pooling + Sigmoid
- Cross-entropy loss

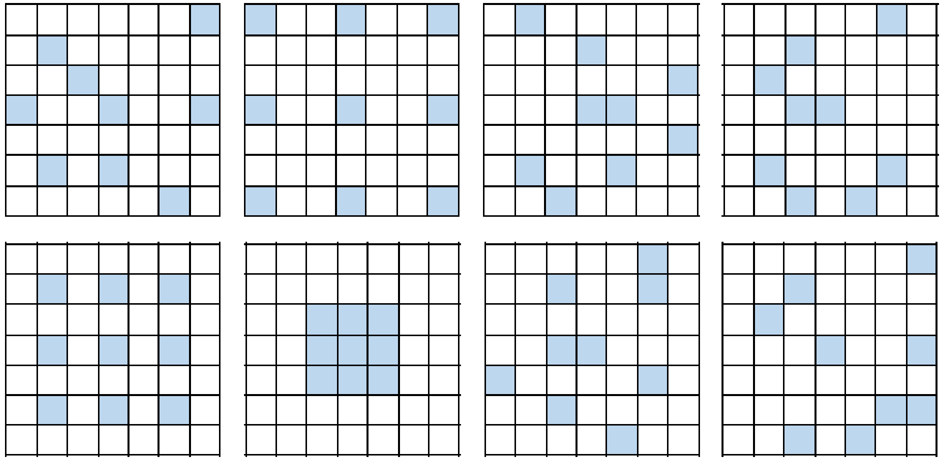


Figure 3. Examples of the selection of 9 hidden units (marked as blue) from a 7 × 7 kernel. Channels are not shown for simplicity. The shapes of those selected hidden units sometimes contain the shape of kernel of convolution with different dilation rates.

Proposed Method

2. Inference Localization Map

① Grad-CAM

② Aggregate Localization Map

- 하나의 이미지로 부터 많은 localization map 생성

  → **다양한 조합으로 classification score 계산**

- Activation score가 일정 threshold 보다 높으면 그 pixel에는 해당 class가 위치

  → **어떠한 class도 위치하지 않은 pixel은 training에서 제외**

  → **한 pixel에 다수의 class가 위치하면, 특정 class 별로 score의 평균을 계산하여 가장 높은 score의 class로 결정**

Proposed Method

3. Training the Segmentation Network

– 생성된 localization map은 semantic segmentation network 학습에 이용

Weakly-supervised:

$$L = L_{seed} + L_{boundary}$$

Predicted segmentation map

Normalization

$$L_{seed} = -\frac{1}{\sum_{c \in C} \boxed{|S_c|}} \sum_{c \in \boxed{C}} \sum_{u \in S_c} \log \boxed{H_{u,c}} - \overbrace{\frac{1}{\sum_{c \in \bar{C}} |S_c|} \sum_{c \in \boxed{\bar{C}}} \sum_{u \in S_c} \log H_{u,c}}^{\text{Normalization}}$$

Seed cues

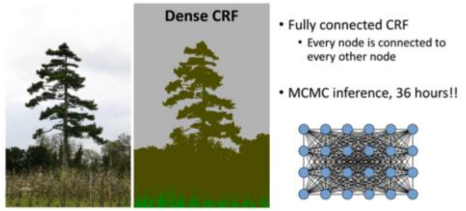
Foreground class

Background class

$$L_{boundary} = -\frac{1}{n} \sum_{u=1}^{n} \sum_{c \in C} \boxed{Q_{u,c}}\left(X, \underbrace{\boxed{f(X)}}_{}\right) \log \frac{Q_{u,c}(X, f(X))}{f_{u,c}(X)}$$

$= H_{u,c}$

Fully-connected
Conditional Random Field



https://m.blog.naver.com/laonple/221017461464

Semi-supervised:

$$L = L_{seed} + L_{boundary} + \alpha L_{full}$$

$$L_{full} = -\frac{1}{\sum_{c \in C} \boxed{|F_c|}} \sum_{c \in C} \sum_{u \in F_c} \log H_{u,c}$$

Ground truth

Experiments

1. Experimental Setup

- Dataset : Pascal VOC 2012
- Network details
  - Classification : pre-trained VGG-16
  - Segmentation : Deeplab-CRF-LargeFOV
- Experimental details
  - Batch size : 10
  - Image size : 321 x 321
  - Learning rate & Optimizer : 0.001 and halved every 10 epochs & Adam optimizer
  - # of localization map : 200
  - threshold & α for semi-supervised learning : 0.35 & 2

Experiments

2. Comparison to the State of the Art

- Weakly supervised segmentation

Table 1. Comparison of weakly supervised semantic segmentation methods on VOC 2012 validation and test image sets. The methods listed here use DeepLab-VGG16 for segmentation.

Methods	Training	*val*	*test*
Supervision: Image-level and additional annotations			
MIL-seg CVPR '15 [23]	700K	42.0	40.6
STC TPAMI '17 [32]	50K	49.8	51.2
TransferNet CVPR '16 [9]	70K	52.1	51.2
CrawlSeg CVPR '17 [10]	970K	58.1	58.7
AISI ECCV '18 [11]	11K	61.3	62.1
Supervision: Image-level annotations only			
SEC ECCV '16 [16]	10K	50.7	51.1
CBTS-cues CVPR '17 [24]	10K	52.8	53.7
TPL ICCV '17 [14]	10K	53.1	53.8
AE_PSL CVPR '17 [31]	10K	55.0	55.7
DCSP BMVC '17 [2]	10K	58.6	59.2
MEFF CVPR '18 [8]	10K	-	55.6
GAIN CVPR '18 [19]	10K	55.3	56.8
MCOF CVPR '18 [30]	10K	56.2	57.6
AffinityNet CVPR '18 [1]	10K	58.4	60.5
DSRG CVPR '18 [12]	10K	59.0	60.4
MDC CVPR '18 [33]	10K	60.4	60.8
FickleNet (Ours)	10K	**61.2**	**61.9**

Table 2. Comparison of weakly supervised semantic segmentation methods on VOC 2012 validation and test image sets. The methods listed here use ResNet-based DeepLab for segmentation.

Methods	Backbone	*val*	*test*
MCOF [30]	ResNet 101	60.3	61.2
DCSP [2]	ResNet 101	60.8	61.9
DSRG [12]	ResNet 101	61.4	63.2
AffinityNet [1]	ResNet 38	61.7	63.7
FickleNet (ours)	ResNet 101	**64.9**	**65.3**

Experiments

2. Comparison to the State of the Art

- Semi-supervised segmentation

Table 3. Comparison of semi-supervised semantic segmentation methods on VOC 2012 validation sets. We also give the performances of DeepLab using 1.4K and 10.6K strongly annotated data.

Methods	Training Set	mIoU
DeepLab [3]	1.4K strong	62.5
WSSL [21]	1.4K strong + 9K weak	64.6
GAIN [19]	1.4K strong + 9K weak	60.5
MDC [33]	1.4K strong + 9K weak	65.7
DSRG [12] (baseline)	1.4K strong + 9K weak	64.3
FickleNet (ours)	1.4K strong + 9K weak	**65.8**
DeepLab [3]	10.6K strong	67.6

Experiments

2. Comparison to the State of the Art

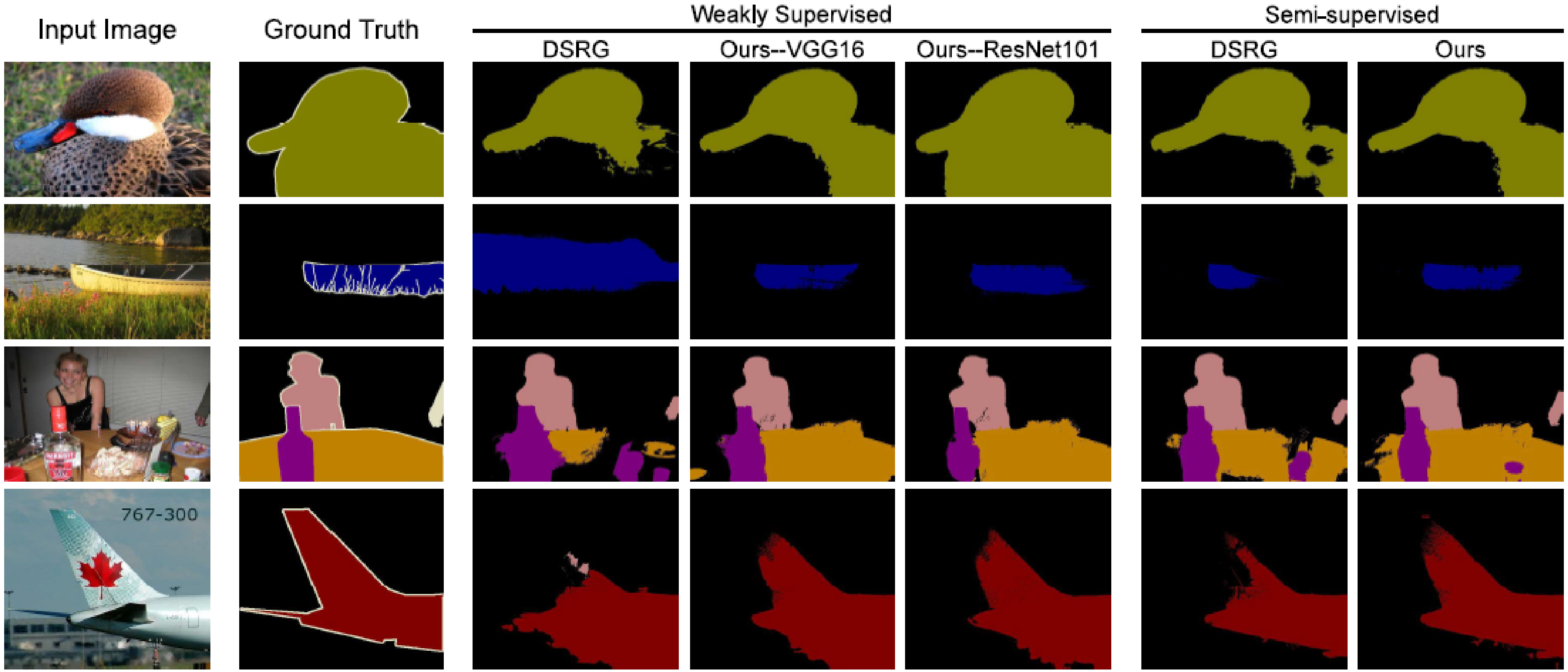


Figure 4. Examples of predicted segmentation masks for Pascal VOC 2012 validation images in weakly and semi-supervised manner.

Experiments

3. Ablation studies

① Effects of the Map Expansion Technique

→ Training, CAM extraction 각각 15.4, 14.2배 감소

→ GPU usage는 12% 증가 (확장된 localization map의 크기 때문에)

Table 4. Run time and GPU memory usage for training and CAM extraction without and with map expansion.

Methods	Training	CAM Extract	GPU Usage
Naive	20 sec/iter	2.98 sec/img	8.4 GB
Expansion	1.3 sec/iter	0.21 sec/img	10.1 GB

Experiments

3. Ablation studies

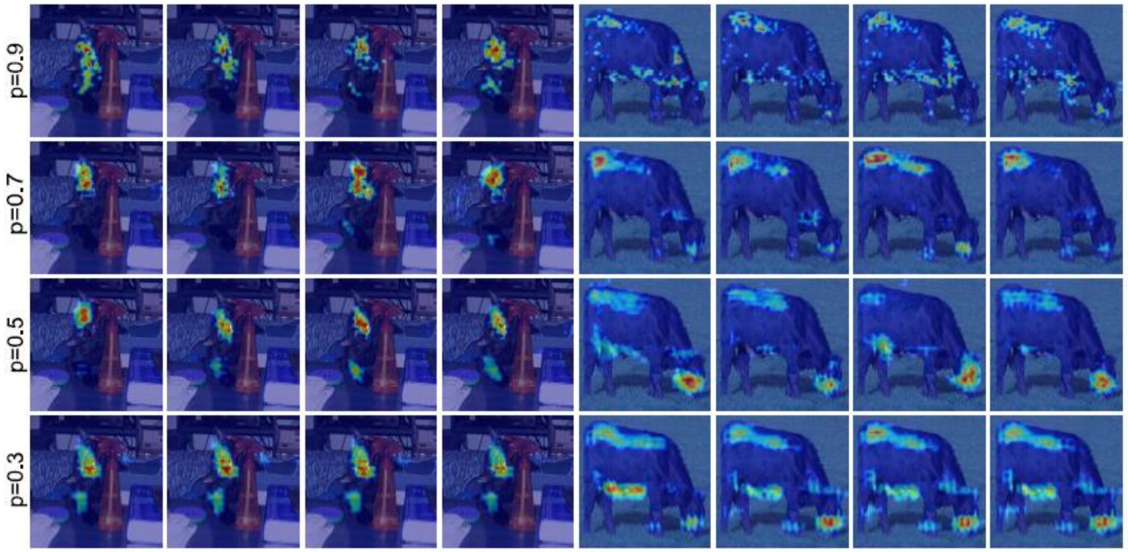
② Analysis of Iterative Inference

→ N이 커질수록 mIoU 증가 추세

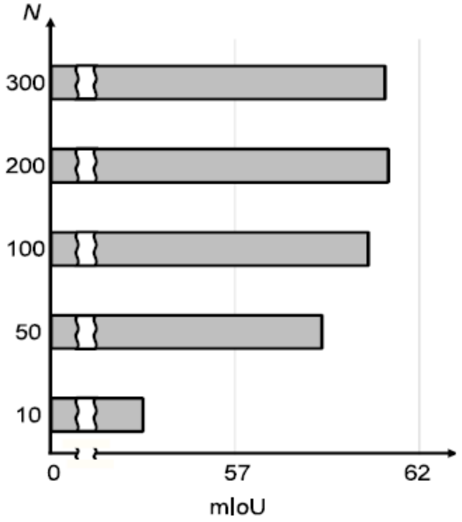
→ N이 커질수록 metric의 std가 감소

Table 6. Standard deviation of mIoU, recall, precision (direct measures).

N	10	100	200	300
std (mIoU, 10^{-3})	21	14	6.8	4.8
std (recall, 10^{-5})	22.4	14.8	6.72	3.41
std (prec, 10^{-5})	27.7	12.3	8.77	9.99



(a)



(b)

Experiments

3. Ablation studies

③ Analysis of Dropout

Effects of dropout rate

- Dropout rate = 0.9일 때 DSRG보다 넓은 영역을 커버함
- 높은 rate는 object의 discriminative part를 drop
  → non-discriminative part를 사용하게 만듦
- 낮은 rate는 object의 discriminative part가 drop되지 않을 수 있음
  → 이 부분만으로 classification을 진행할 수 있기 때문에 non-discriminative part 사용 X

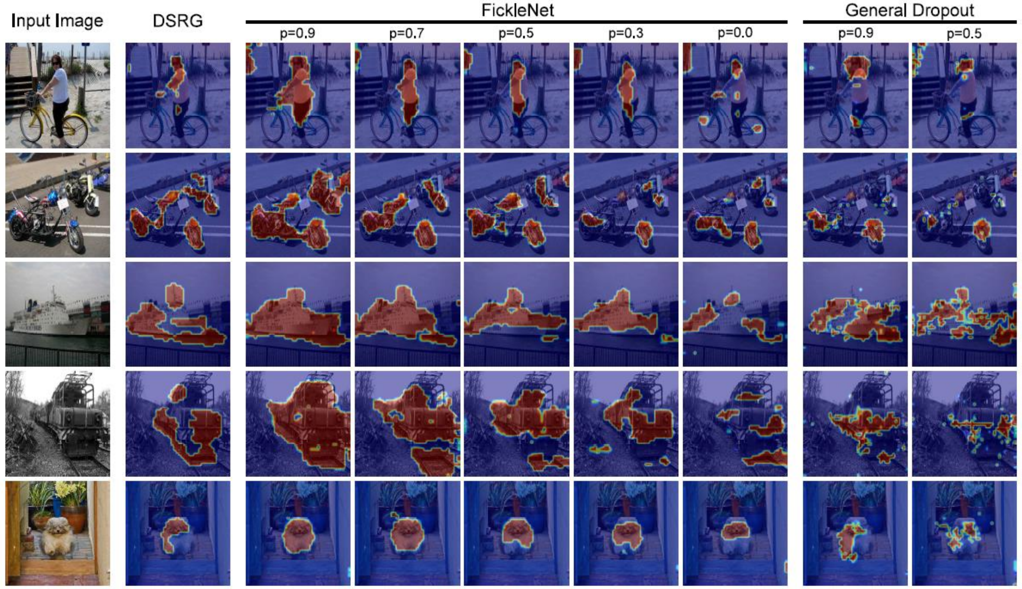


Figure 6. Localization maps from DSRG and FickleNet, with various dropout rates (p = 0 denotes a deterministic network), and from the general dropout method. Localization maps of DSRG (the 2nd column) were visualized using the publicly available DSRG localization cue.

Experiments

3. Ablation studies

③ Analysis of Dropout

Comparison to general dropout

- General dropout으로 만들어진 feature map은 noisy해보임

  → inference 시 dropout 시행 X (+ center preserving X)

Effectiveness of each steps

Table 5. Comparison of mIoU scores using different dropout rates (p) on PASCAL VOC 2012 validation images.

Methods	Dropout Rate (p)	mIoU
Deterministic	0.0	56.3
General Dropout	0.5	45.6
	0.9	49.1
FickleNet	0.3	58.8
	0.5	59.4
	0.7	60.0
	0.9	**61.2**

Table 7. Effectiveness of each step. $\mathcal{G}$− general dropout, $\mathcal{S}$− stochastic selection, $\mathcal{D}$− deterministic approach.

Training	$\mathcal{G}$	$\mathcal{G}$	$\mathcal{G}$	$\mathcal{S}$	$\mathcal{S}$	$\mathcal{D}$
Inference	$\mathcal{G}$	$\mathcal{S}$	$\mathcal{D}$	$\mathcal{S}$	$\mathcal{D}$	$\mathcal{D}$
mIoU	49.1	55.5	57.1	**61.2**	59.6	59.0

Conclusions

1. Stochastic selection으로 많은 localization map을 얻은 뒤, 하나로 통합

   → 기존보다 더 넓은 activation map을 구할 수 있었음

2. Localization map을 kernel size에 맞게 확장

   → GPU를 효율적으로 사용하여 학습 및 CAM 추출에 시간 단축

3. Weakly supervised & semi-supervised segmentation 모두 활용 가능

감 사 합 니 다