PUZZLE-CAM: IMPROVED LOCALIZATION VIA MATCHING PARTIAL AND FULL FEATURES (+RecurSeed and EdgePredictMix)

2023. 01. 12

Presenter : Taewoong Kang
twk@deepnoid.com

Overview

- Weakly Supervised Semantic Segmentation (WSSS)
- 이전은 CAM을 이용해 segmentation mask를 얻으려 refine 한 논문들이 많았다.
- 이 논문은 reconstructing regularization with a puzzle module을 포함한 CAM으로 Semantic Segmentation하는 것이 목표이다.

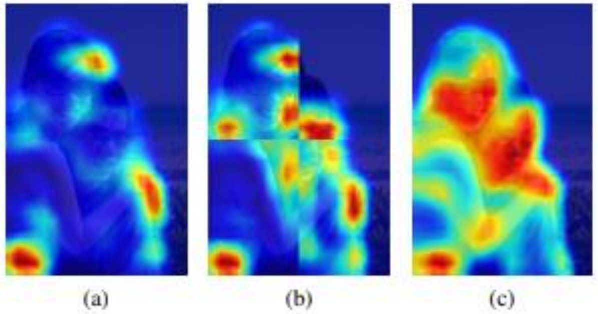


Fig. 1: CAMs generated from tiled and original images: (a) conventional CAMs from the original image, (b) generated CAMs from the tiled images, and (c) predicted CAMs by the proposed Puzzle-CAM.

- Comparison of CAM and Puzzle-CAM

- Puzzle-CAM architecture

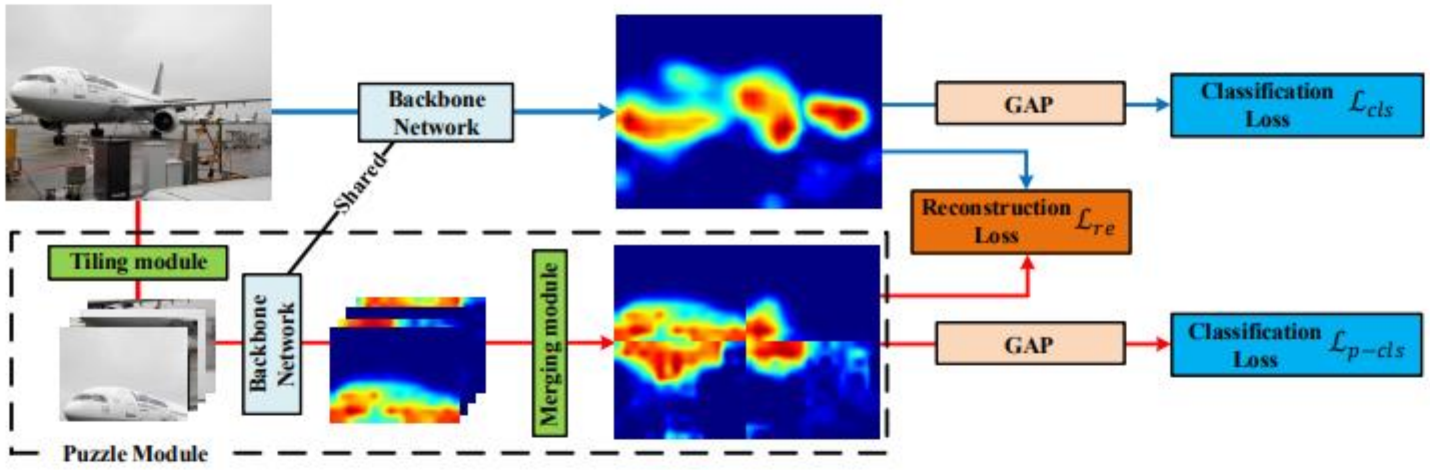


Fig. 2: The overall architecture of the proposed Puzzle-CAM showing the integration of reconstructing regularization and the puzzle module.

Class Activation Mapping (CAM)

Overview

- Weakly supervised **object localization**
- Identify the importance of the image regions by **projecting back the weights** of the output layer on the convolutional feature maps
- 기존 Network와 유사한 Classification 정확도(1~2% 하락)와 더불어 Localization까지 가능하다.
- (어디에 집중해서 Classification을 하는지와 연관)

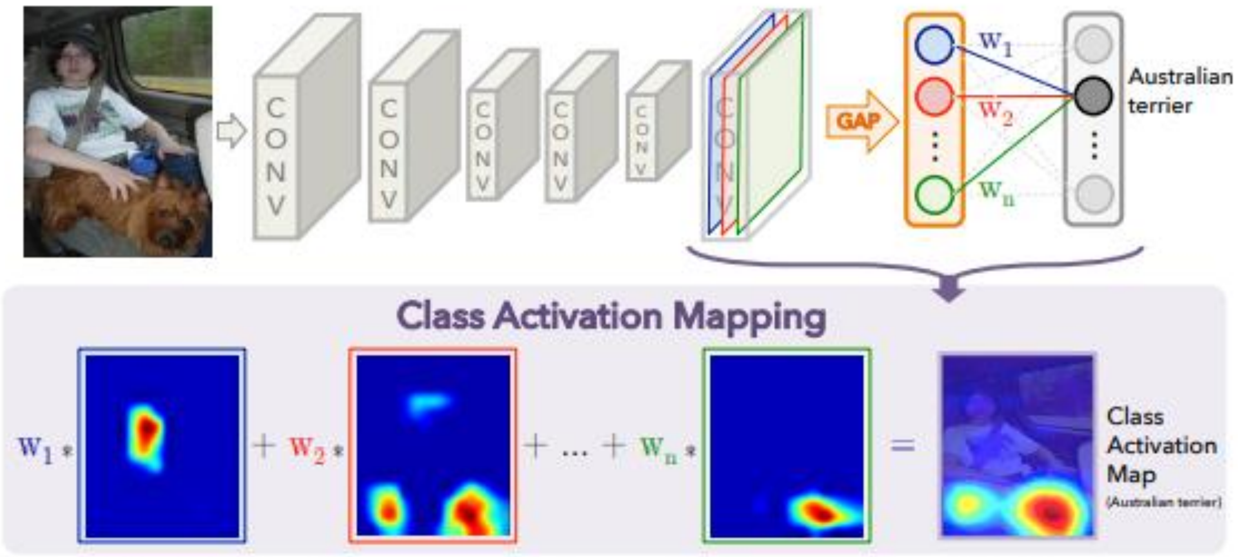


Figure 2. Class Activation Mapping: the predicted class score is mapped back to the previous convolutional layer to generate the class activation maps (CAMs). The CAM highlights the class-specific discriminative regions.

- Basic CAM Flow

Class Activation Mapping (CAM)

Architecture

- Fully Connected Layer 이전에 GAP Layer 사용 (just before the final out-put layer)

  = FC Layer -> GAP + FC softmax Layer

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

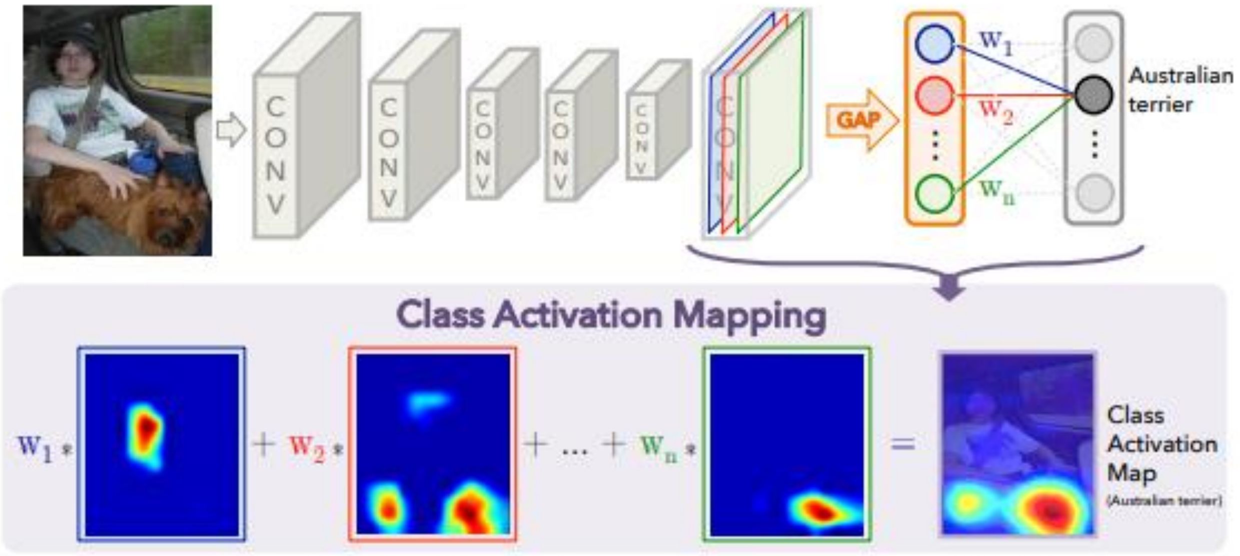


Figure 2. Class Activation Mapping: the predicted class score is mapped back to the previous convolutional layer to generate the class activation maps (CAMs). The CAM highlights the class-specific discriminative regions.

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

$$M_c(x,y) = \sum_k w_k^c f_k(x,y)$$

- Basic CAM Flow

Localization

- Use a simple thresholding technique to segment the heatmap
- CAM의 max value의 20%이상 만 사용.
- Segmentation map에서 largest connected component를 cover하는 Bbox 생성
    - But Top-5 test error 37.1
    - Annotated Bounding box로 train 되지 않았다는 것이 주목할 만한 점이다.

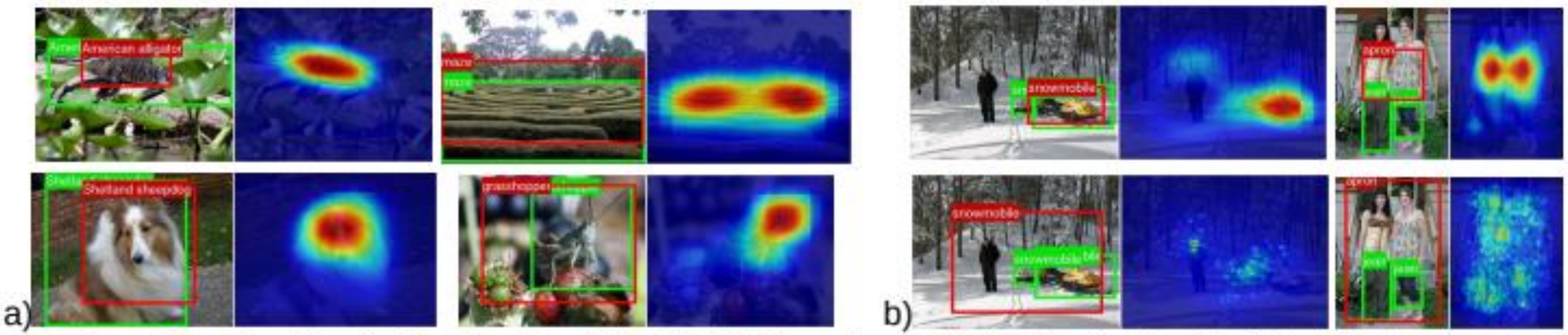


Figure 6. a) Examples of localization from GoogleNet-GAP. b) Comparison of the localization from GooleNet-GAP (upper two) and the backpropagation using AlexNet (lower two). The ground-truth boxes are in green and the predicted bounding boxes from the class activation map are in red.

DEEP NOID

Limit

- **Focused on small parts** of the semantic objects to efficiently classify them

  -〉prevent the segmentation models from learning pixel-level semantic knowledge.

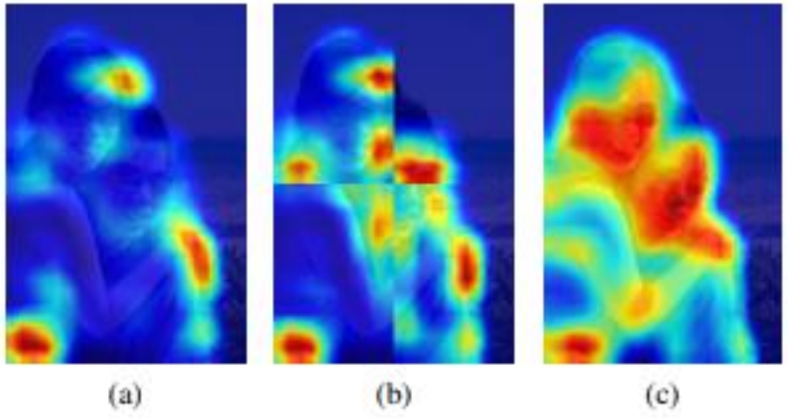


Fig. 1: CAMs generated from tiled and original images: (a) conventional CAMs from the original image, (b) generated CAMs from the tiled images, and (c) predicted CAMs by the proposed Puzzle-CAM.

Overview

- Image-level supervision (WSSS)
- Reconstructing regularization with a puzzle module

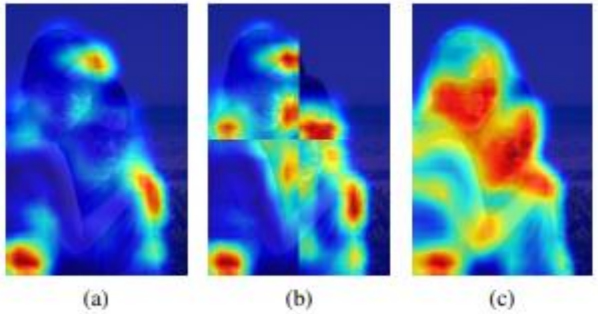


Fig. 1: CAMs generated from tiled and original images: (a) conventional CAMs from the original image, (b) generated CAMs from the tiled images, and (c) predicted CAMs by the proposed Puzzle-CAM.

- Comparison of CAM and Puzzle-CAM

- Puzzle-CAM architecture

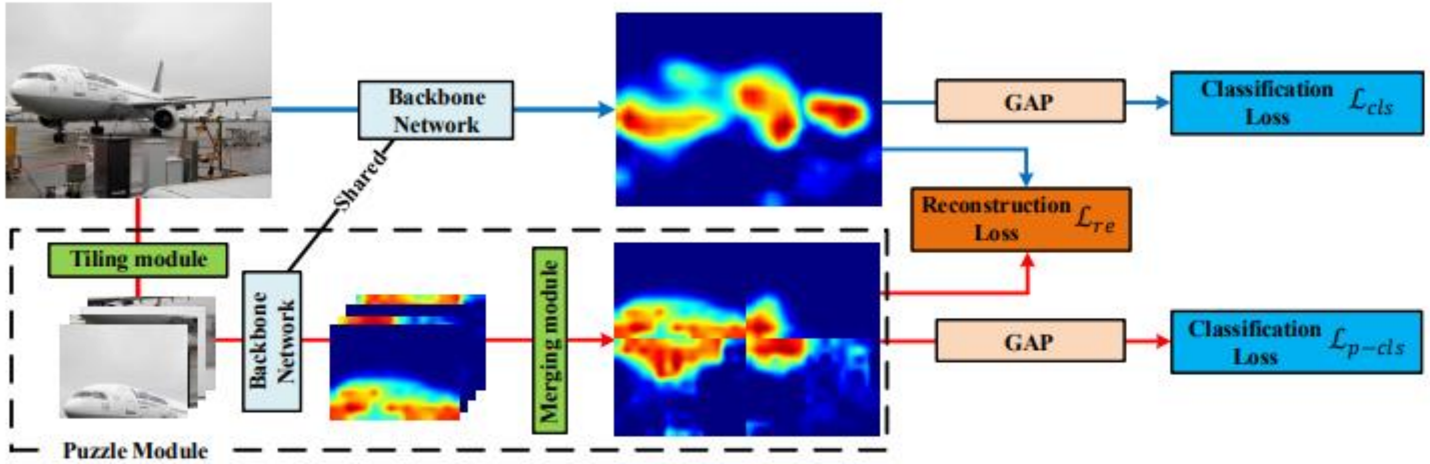


Fig. 2: The overall architecture of the proposed Puzzle-CAM showing the integration of reconstructing regularization and the puzzle module.

Overview

- 원래 이미지에 대한 CAM과, Puzzle Module을 이용한 CAM을 각각 구함
- Puzzle Module은 본 이미지를 4개로 나눈 후 각 puzzle에 대한 CAM을 구한 후 한 개로 합침. (두 개의 CAM은 같은 크기)
- 두 개의 CAM에 대한 Classification Loss, 두 CAM의 차이로 구한 Reconstruction Loss를 이용해 학습

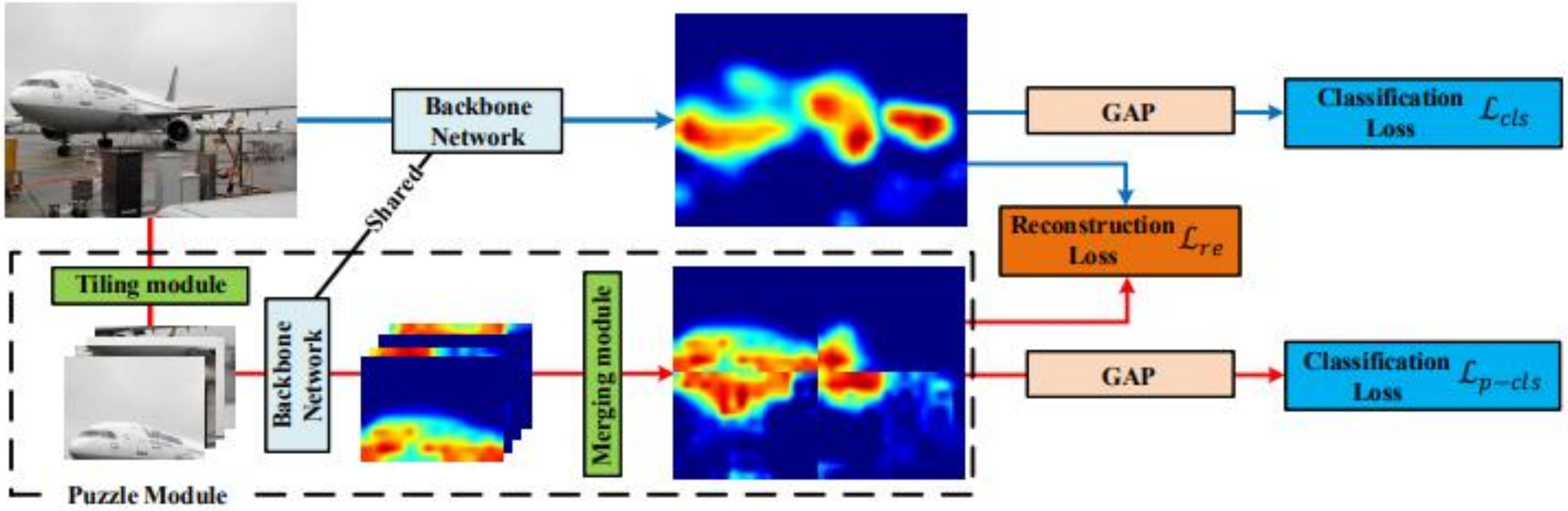
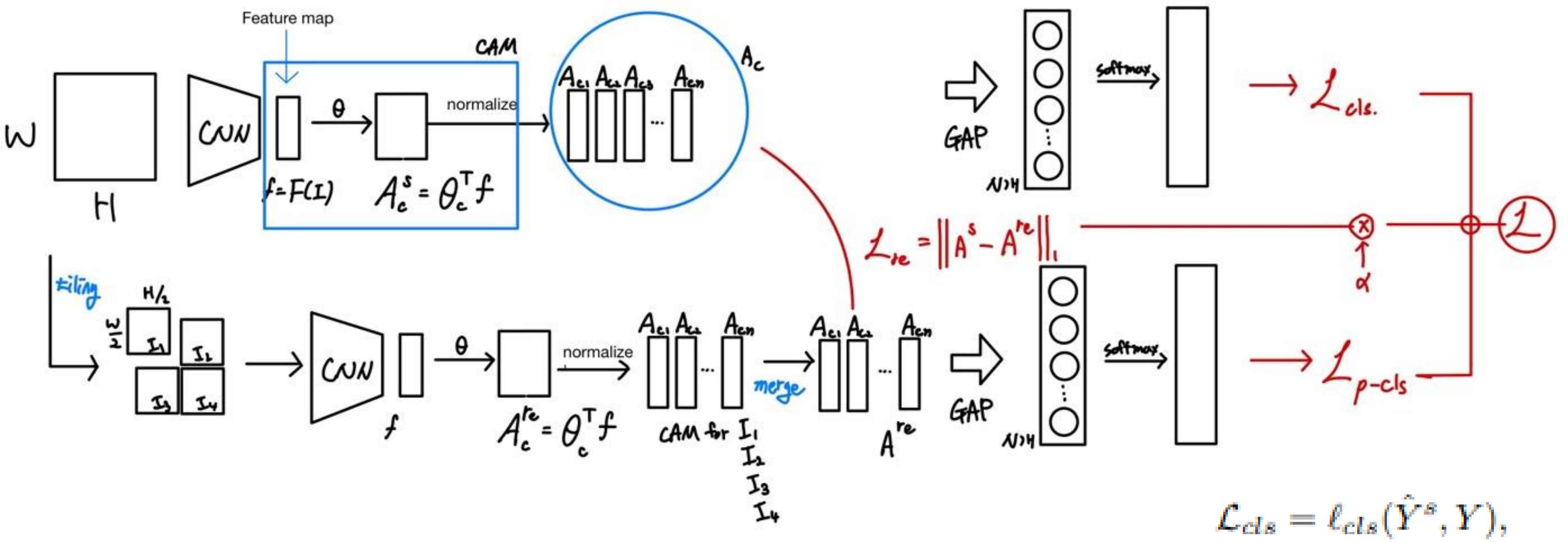


Fig. 2: The overall architecture of the proposed Puzzle-CAM showing the integration of reconstructing regularization and the puzzle module.

Loss function

- 원래 이미지에 대한 CAM과, Puzzle Module을 이용한 CAM을 각각 구함
- Puzzle Module은 본 이미지를 4개로 나눈 후 각 puzzle에 대한 CAM을 구한 후 한 개로 합침. (두 개의 CAM은 같은 크기)
- 두 개의 CAM에 대한 Classification Loss, 두 CAM의 차이로 구한 Reconstruction Loss를 이용해 학습

$$\hat{Y} = \sigma(G(A_c))$$



$$\mathcal{L} = \mathcal{L}_{cls} + \mathcal{L}_{p-cls} + \alpha\mathcal{L}_{re}.$$

$$\hat{Y}_t = \begin{cases} \hat{Y}, & \text{if } Y = 1 \\ 1 - \hat{Y}, & \text{otherwise} \end{cases}$$

$$\mathcal{L}_{cls} = \ell_{cls}(\hat{Y}^s, Y),$$
$$\mathcal{L}_{p-cls} = \ell_{cls}(\hat{Y}^{re}, Y).$$

$$\ell_{cls}(\hat{Y}, Y) = -log(Y_t).$$

Result

Method	Backbone	Supervision	val	test
AffinityNet [4]	Wide-ResNet-38	$\mathcal{I}$	61.7	63.7
DSRG [12]	ResNet-101	$\mathcal{I}+\mathcal{S}$	61.4	63.2
SeeNet [13]	ResNet-101	$\mathcal{I}+\mathcal{S}$	63.1	62.8
IRNet [4]	ResNet-50	$\mathcal{I}$	63.5	64.8
FickleNet [6]	ResNet-101	$\mathcal{I}+\mathcal{S}$	64.9	65.3
ICD [17]	ResNet-101	$\mathcal{I}$	64.1	64.3
SEAM [5]	Wide-ResNet-38	$\mathcal{I}$	64.5	65.7
Ours (Puzzle-CAM)	ResNeSt-101	$\mathcal{I}$	66.9	67.7
Ours (Puzzle-CAM)	ResNeSt-269	$\mathcal{I}$	**71.9**	**72.2**

여러 WSSS 기법 들과의 비교

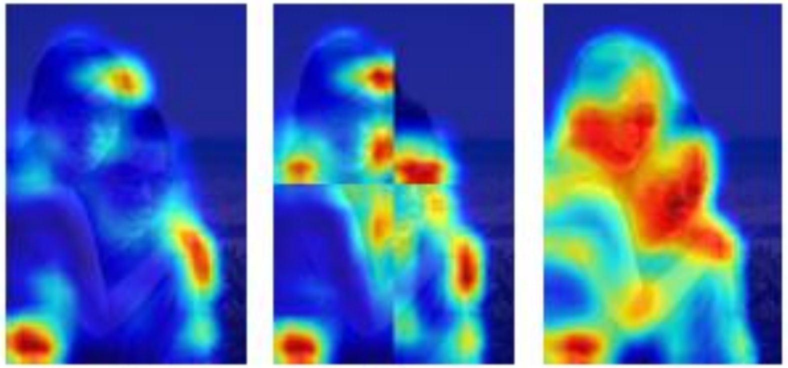
L_{cls}	L_{p-cls}	L_{re}	mIoU (%)
✓			47.82
✓	✓		47.70
✓		✓	49.21
✓	✓	✓	**51.53**

Loss 값 유무에 따른 성능의 변화

Fig. 4: Qualitative segmentation results on the PASCAL VOC 2012 *val* set. Top: original images. Middle: ground truth. Bottom: prediction of the segmentation model trained using the pseudo-labels from Puzzle-CAM.

Limit

- Represent partial regions for large-scale objects
- For small-scale objects, over activation causes them to deviate from the object edges



(a) (b) (c)

Fig. 1: CAMs generated from tiled and original images: (a) conventional CAMs from the original image, (b) generated CAMs from the tiled images, and (c) predicted CAMs by the proposed Puzzle-CAM.

DEEP NOID

Overview

- To solve two problem (in WSSS)

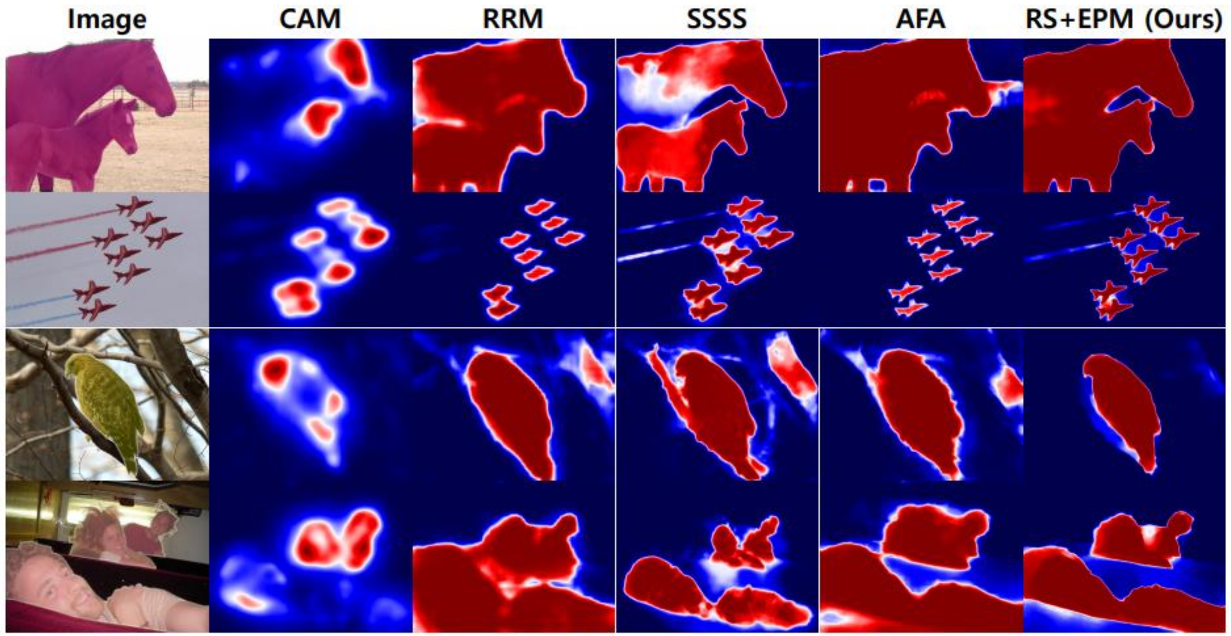
- FPN-WSSS-IL -〉 mismatching of FNs and FPs

(SCG considering high-order correlation reduces FNs, but increases the number of FPs)

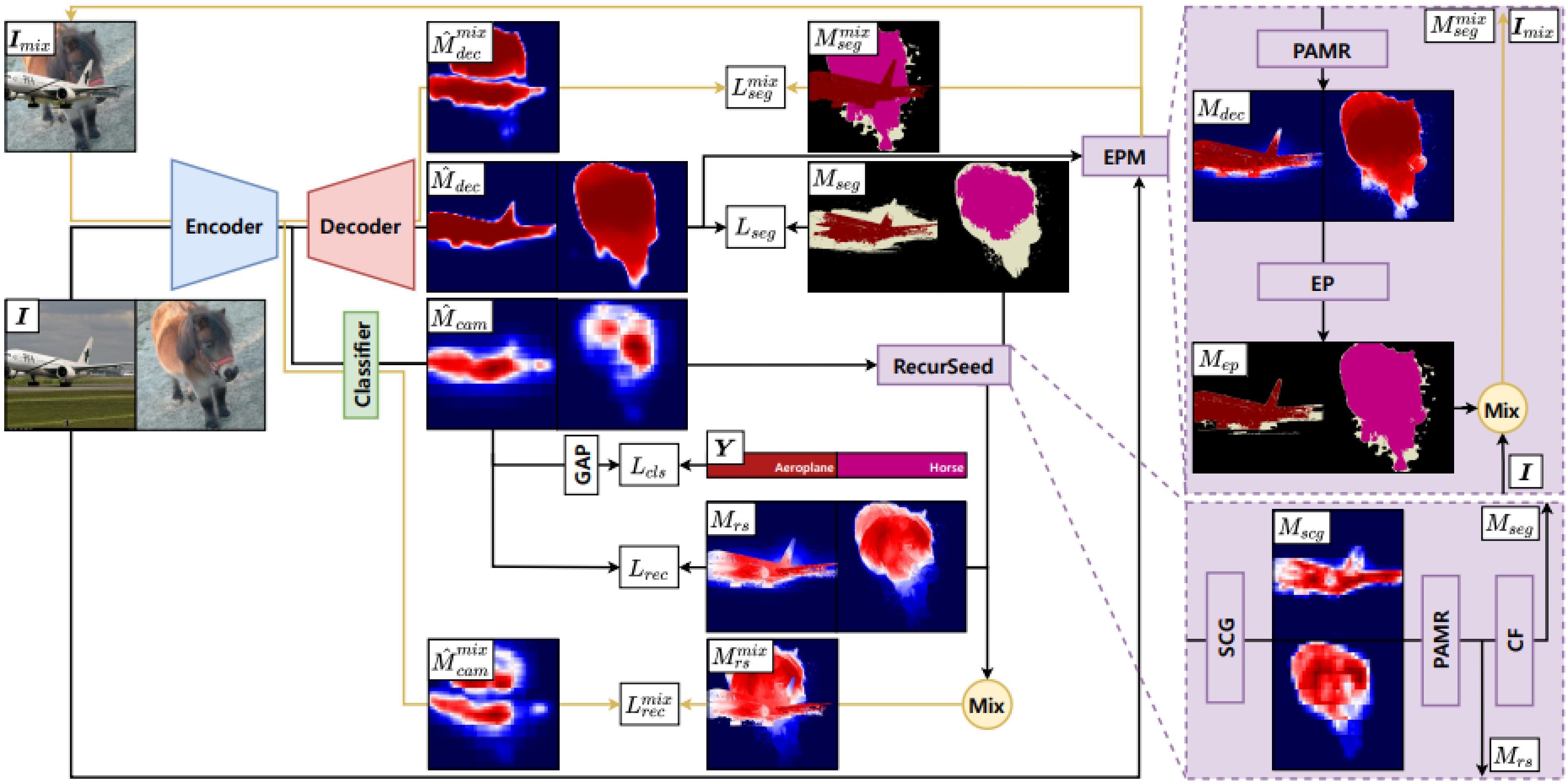
-〉 Using Recurseed (both SCG and PAMR)

- IBDA-WSSS-IL -〉 the simple synthesis(Cutout, CutMix, SaliencyGrafting, CDA and ClassMix) without any refinements using predicted masks inevitably accelerates the ambiguity of mixed result due to insufficient spread

-〉 using EPM



Overview



- SLF (Single-stage Learning Frame)
- Recursion
- Using RecurSeed and EdgePredictMix

Self Correlation map Generation (SCG)

Algorithm Flow

- Self-correlation 을 통해 CAM을 보정

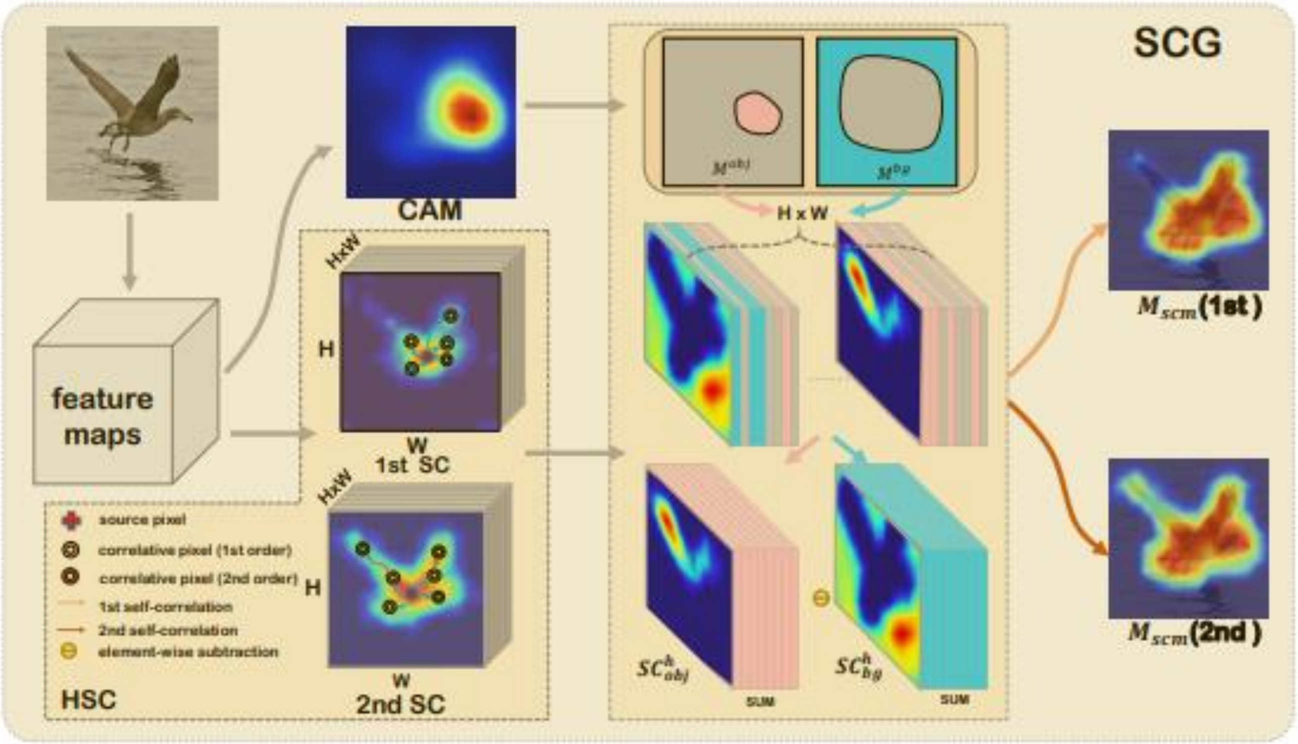
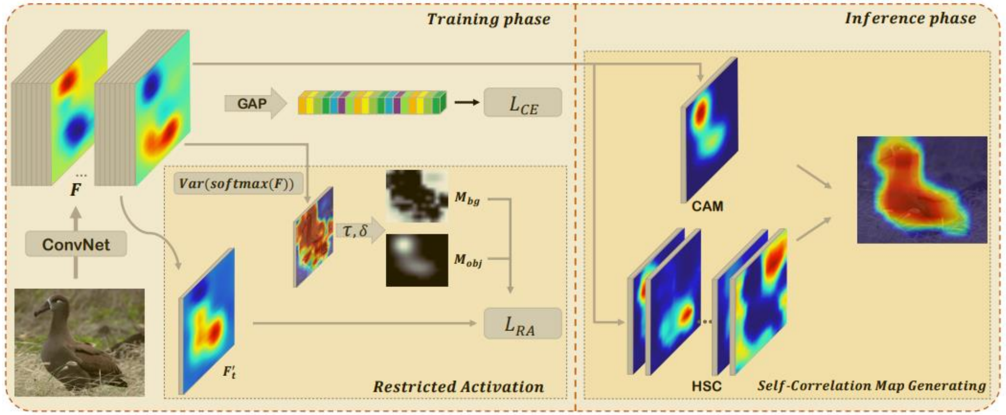


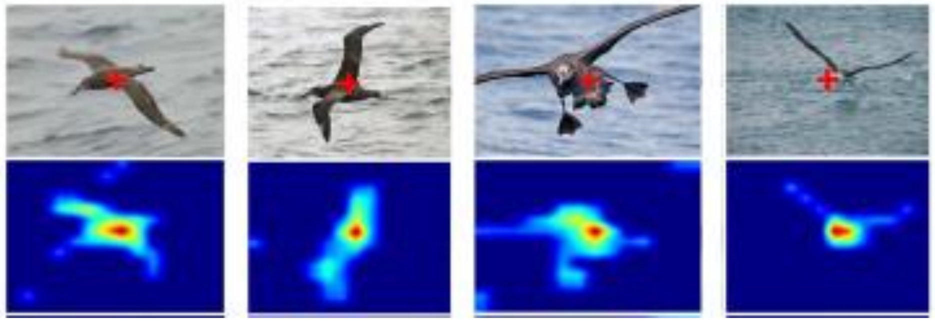
Figure 3. Pipeline of the proposed SCG module. Here we show examples of using first- and second-order SC to obtain final localization maps, respectively.



Self Correlation map Generation (SCG)

Algorithm Flow

First-order Self-correlation

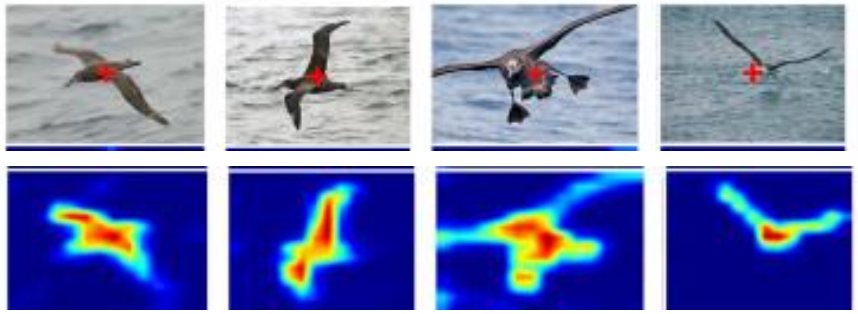


$$S(f_i, f_j) = \frac{{f_i}^{\mathrm{T}} f_j}{||(f_i)|| \cdot ||(f_j)||},$$

$$SC^1(f) = \left[SC^1(f)_{i,j} \right],$$

$$where \;\; SC^1(f)_{i,j} = ReLU(S(f_i, f_j)).$$

Second-order Self-correlation



$$S^2(f_i, f_j) = \frac{1}{(HW)} \sum_{k \in \Omega} S(f_i, f_k) \cdot S(f_k, f_j),$$

$$\hat{S}^2(f_i, f_j) = \frac{S^2(f_i, f_j) - min_{k\in\Omega} S^2(f_i, f_k)}{max_{k\in\Omega} S^2(f_i, f_k) - min_{k\in\Omega} S^2(f_i, f_k)},$$

$$SC^2(f) = \left[\hat{S}^2(f_i, f_j))|_{i,j} \right].$$

Self Correlation map Generation (SCG)

Algorithm Flow

f_i, f_j : $CAM\ features\ of\ specific\ index$

$f_i, f_j \in R^{C\ X\ 1}$

First-order Self-correlation

$$SC_l^1[i,j,:,:] = \mathrm{ReLU}\left(\frac{{f_i^l}^\top f_j^l}{\|f_i^l\|\,\|f_j^l\|}\right),$$

Second-order Self-correlation

$$SC_l^2[i,:,:,:] = \mathrm{mnmx}_j\left(\mathrm{avg}_k\left[\frac{{f_i^l}^\top f_k^l}{\|f_i^l\|\,\|f_k^l\|} \odot \frac{{f_k^l}^\top f_j^l}{\|f_k^l\|\,\|f_j^l\|}\right]\right),$$

merge

$$HSC = \frac{1}{L}\sum_{l=1}^{L}\max(SC_l^1, SC_l^2)$$

Adapt to CAM

$$\begin{aligned} M_{scg} &= SCG(M_{cam}) \\ &= ReLU(K_{scg}(\{M_{cam}\}_{>\delta_h}) - K_{scg}(\{M_{cam}\}_{<\delta_l}), \end{aligned}$$

$$K_{scg}(\{M\}_{\lessgtr\delta}) = \frac{1}{|\{M\}_{\lessgtr\delta}|}\sum_{(i,j)\in\{M\}_{\lessgtr\delta}} HSC[i,j,:,:].$$

Algorithm 1 Localization algorithm of SCG.

Input: Coarse localization map $M_{cam} \in \mathbb{R}^{H\times W}$; feature map $f \in \mathbb{R}^{H\times W\times C}$; threshold δ_h and δ_l;

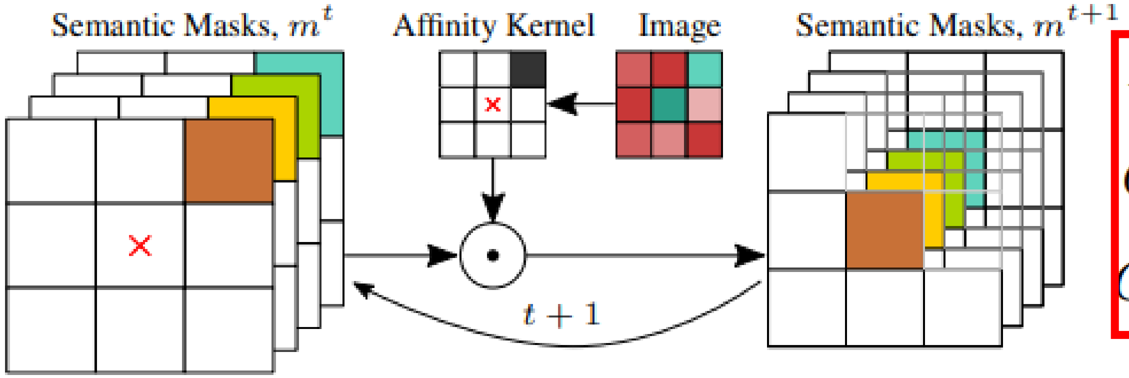
Output: Final localization map M_{scg};

1: Obtain high-order self-correlation $HSC \in \mathbb{R}^{HW\times HW}$
2: Reshape $HSC \in \mathbb{R}^{H\times W\times H\times W} \leftarrow reshape(HSC)$
3: Discover the coarse object region $M_{cam}^{obj} \leftarrow M_{cam} > \delta_h$
4: Extract object HSC $HSC_{obj} \leftarrow G(HSC, M_{cam}^{obj})$
5: Obtain the object map $M_{scg}^{obj} \leftarrow sum(HSC_{obj})$
6: Discover background region $M_{cam}^{bg} \leftarrow M_{cam} < \delta_l$
7: Extract background HSC $HSC_{bg} \leftarrow G(HSC, M_{cam}^{bg})$
8: Obtain the background map $M_{scg}^{bg} \leftarrow sum(HSC_{bg})$
9: Obtain localization map $M_{scg} \leftarrow (M_{scg}^{obj} - M_{scg}^{bg})_{(>0)}$
return M_{scg};

Pixel-adaptive mask refinement (PAMR)

Algorithm Flow

- Local Consistency : nearby regions sharing the same appearance should be assigned to the same class
- Reduced the computational complexity by narrowing the affinity kernel computation to regions of contiguous pixels
- Pixel-wise mask prediction $m_{:,:,:} \in (0,1)^{(C+1)\times h \times w}$ (+1 for the background class)

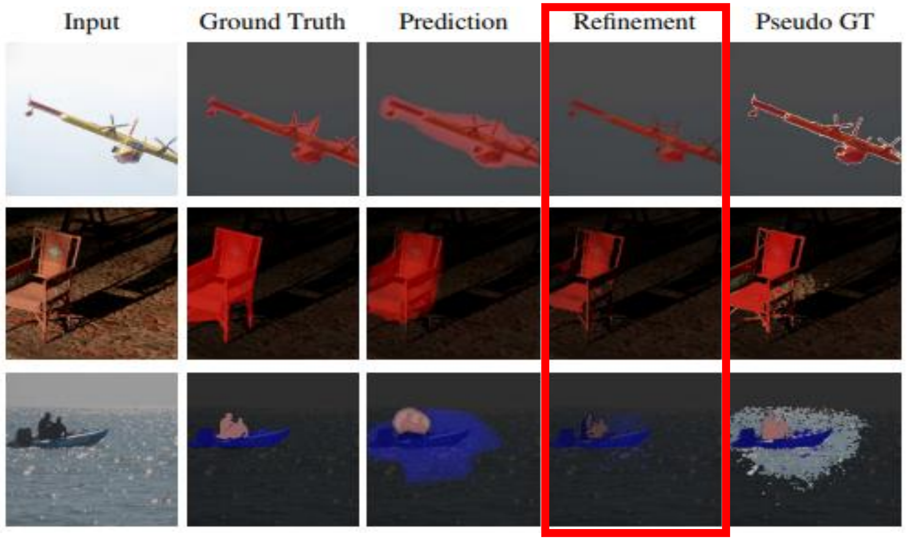


$$M_{pamr} = PAMR(M_{cam}; \mathcal{W}) = G_T(M_{cam})$$

$$G_0(M_{cam}) = M_{cam}$$

$$G_t(M_{cam})[i,j] = \sum_{(k,n)\in\mathcal{N}(i,j)} \alpha_{i,j,k,n}$$

$$m^t_{:,i,j} = \sum_{(l,n)\in\mathcal{N}(i,j)} \alpha_{i,j,l,n} \cdot m^{t-1}_{:,l,n},$$



$$\alpha_{i,j,k,n} = \frac{\exp[\bar{k}(i,j,k,n)]}{\sum_{(q,r)\in\mathcal{N}(i,j)} \exp[\bar{k}(i,j,q,r)]}$$

$$k(i,j,k,n) = -|I_{i,j} - I_{k,n}|/\sigma^2_{i,j}$$

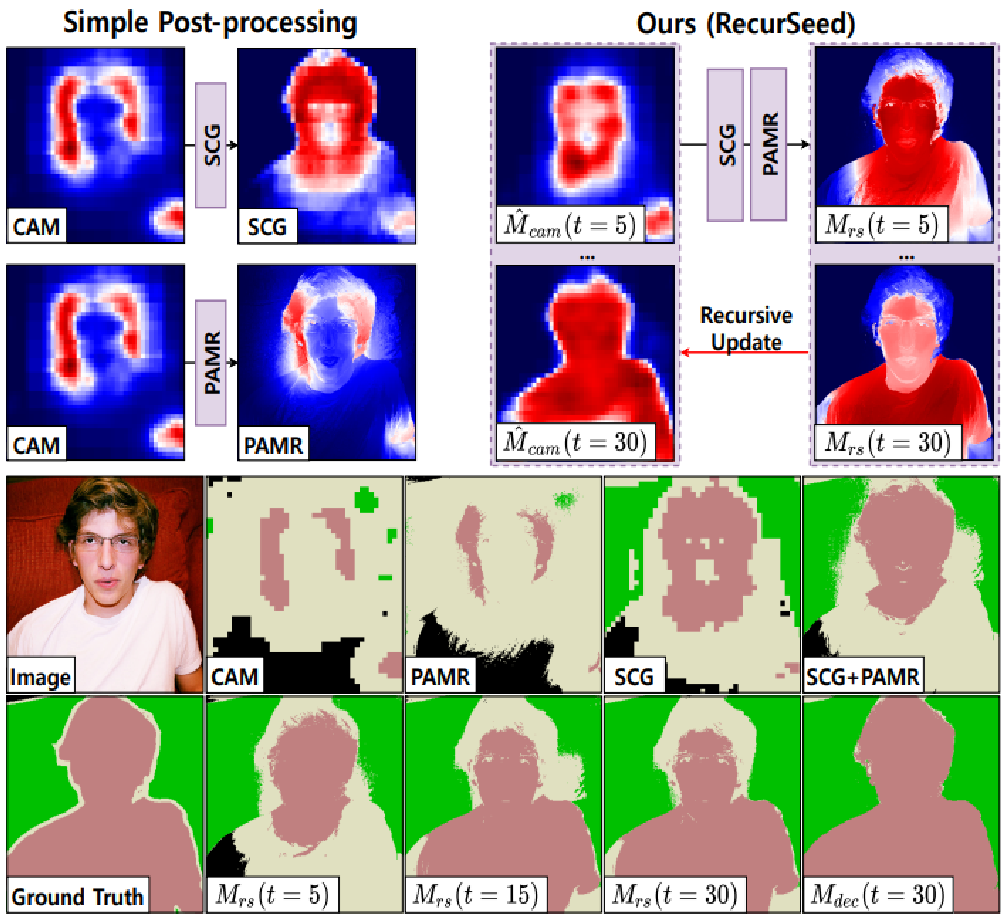
$\bar{k}(\cdot)$ is the average affinity value $k(\cdot)$ across the RGB channels

$\sigma_{i,j}$ denotes the standard deviation of the image intensity computed locally for the affinity kernel

Algorithm Flow

- The performance of SCG depends highly on feature maps owing to its ability to update the CAM from SCG and PAMR recursively
- The proposed RS **gradually updates the initial CAM** to remedy the shortcoming of the SCG and PAMR

$$M_{rs}(t) = PAMR(SCG(M_{cam}(t)); \mathcal{W})$$
$$M_{rs}(t) \approx M_{cam}(t+1) \text{ for every epoch } t \in \{0 : T\}$$



Loss function

To achieve the objective of $M_{rs}(t) \approx M_{cam}(t+1)$

$$
\begin{aligned}
\theta_t = \theta_{t-1} & - \eta \frac{\partial}{\partial\theta} \mathbb{E}_{\boldsymbol{I}} \left[\mathcal{L}_{cls}(\hat{Y}_{cls}(\tau), \boldsymbol{Y}; \theta) \right] \Big|_{\theta=\theta_\tau, \tau=t-1} \\
& - \eta \frac{\partial}{\partial\theta} \mathbb{E}_{\boldsymbol{I}} \left[\mathcal{L}_{seg}(\hat{M}_{dec}(\tau), M_{seg}(\tau); \theta) \right] \Big|_{\theta=\theta_\tau, \tau=t-1} \\
& - \eta \frac{\partial}{\partial\theta} \mathbb{E}_{\boldsymbol{I}} \left[\mathcal{L}_{rec}(\hat{M}_{cam}(\tau), M_{rs}(\tau); \theta) \right] \Big|_{\theta=\theta_\tau, \tau=t-1}
\end{aligned}
$$

where $\boldsymbol{I} = \{I^1, I^2, ..., I^B\}$

$$\boxed{\hat{Y}_{cls}(t) = \sigma(GAP(\hat{M}_{cam}(t)))}$$

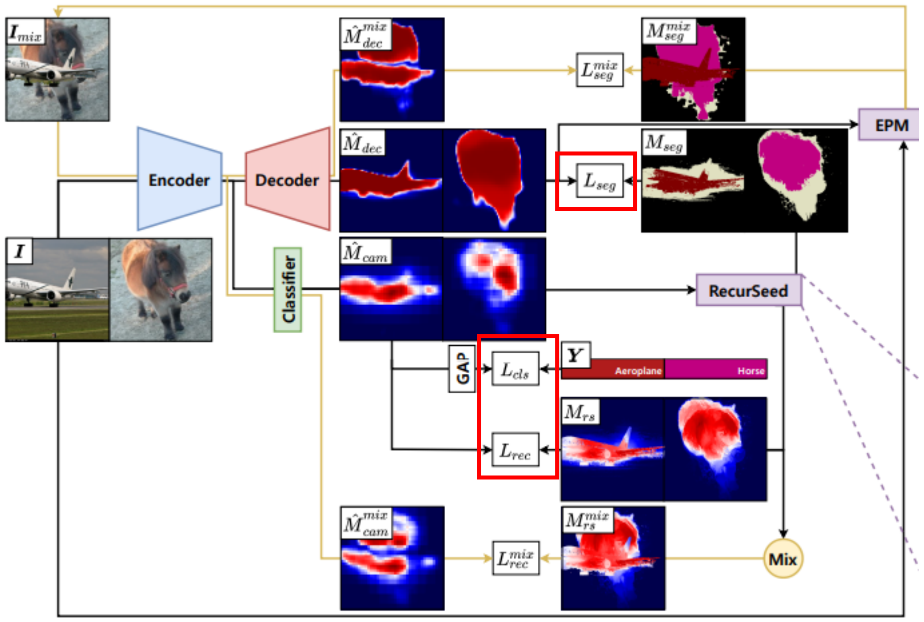
Classification (multi-label soft margin loss)

$$\boxed{\begin{aligned} \hat{M}_{dec}(t) &= D_{\theta_t^{dec}}(f_{enc}(t)) \\ f_{enc}(t) &= E_{\theta_t^{enc}}(\boldsymbol{I}), \end{aligned}}$$

Decoder (cross-entropy loss)

$$\boxed{\hat{M}_{cam}(t) = A_{\theta_t^{cls}}(f_{enc}(t))}$$

Encoder (L1 loss)

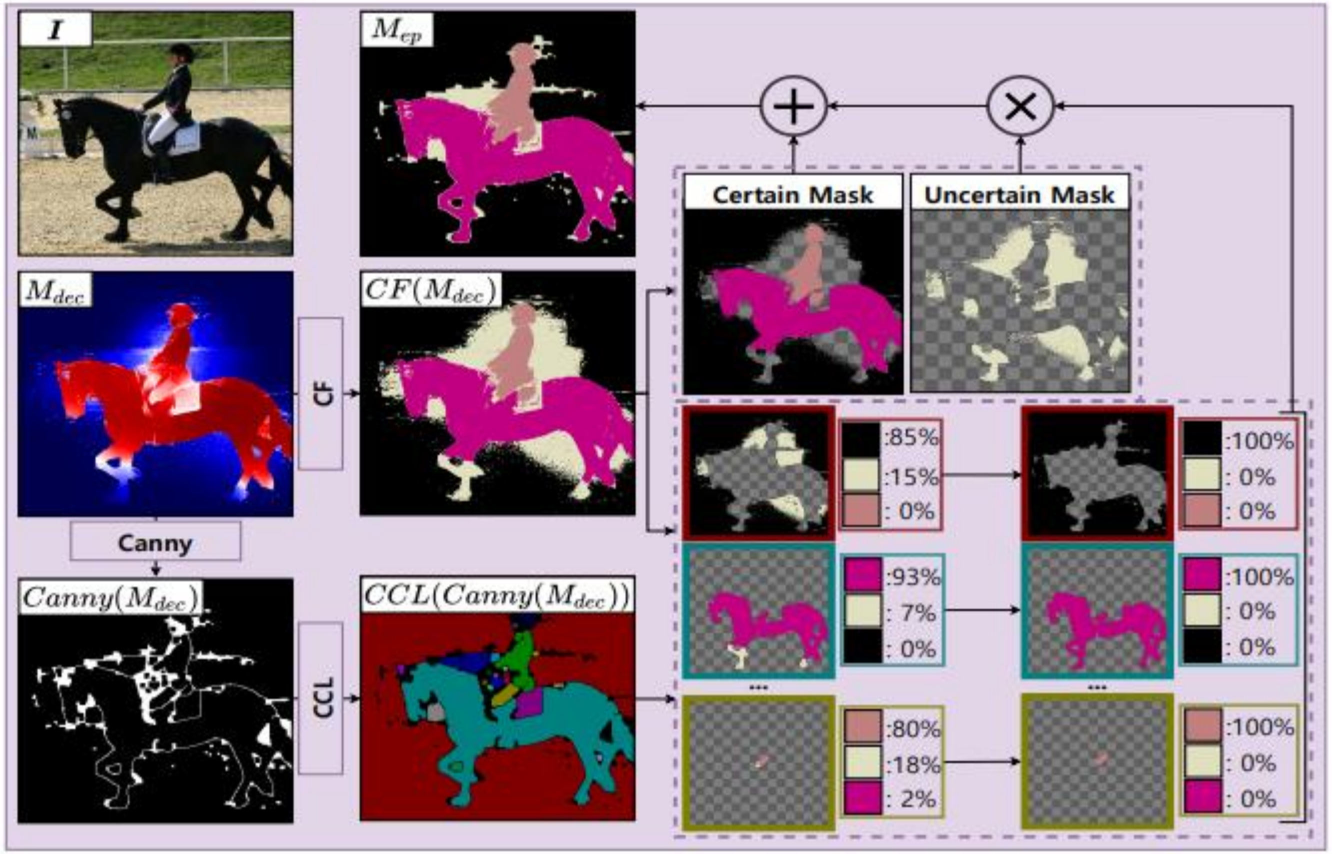


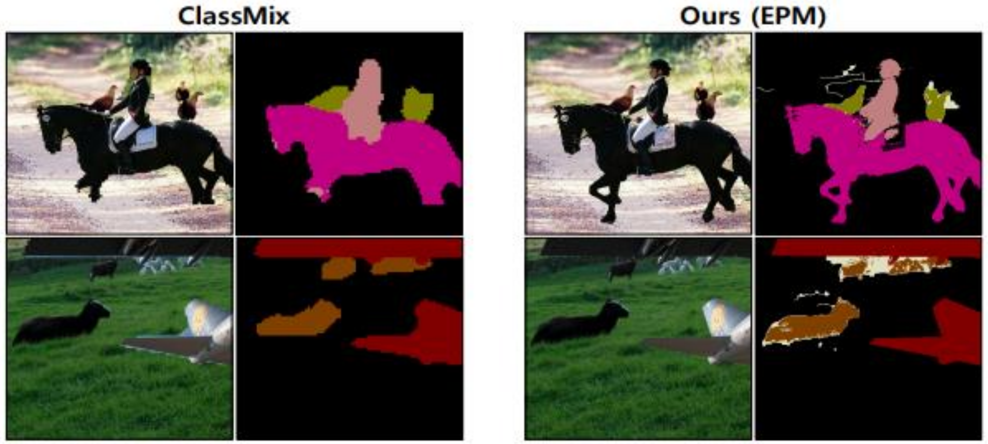
$$
M_{seg}(t) = CF(M_{rs}(t)) =
\begin{cases}
\operatorname{argmax}_{c \in \mathcal{C}} \left(M^c_{rs}(t)[:, i, j] \right) & \text{if } \max_{c \in \mathcal{C}} \left(M^c_{rs}(t)[:, i, j] \right) > \delta_{fg}, \\
0 & \text{if } \max_{c \in \mathcal{C}} \left(M^c_{rs}(t)[:, i, j] \right) < \delta_{bg}, \\
255 & \text{otherwise.}
\end{cases}
$$

DEEP NOID

Overview

- Mixes two images and pseudo masks refined by EP, which disentangles foreground and background regions by using edge information in the per-pixel class probability domain.
- Leading to sample diversification and significantly improving performance for WSSS





EdgePredictMix(EPM)

Overview

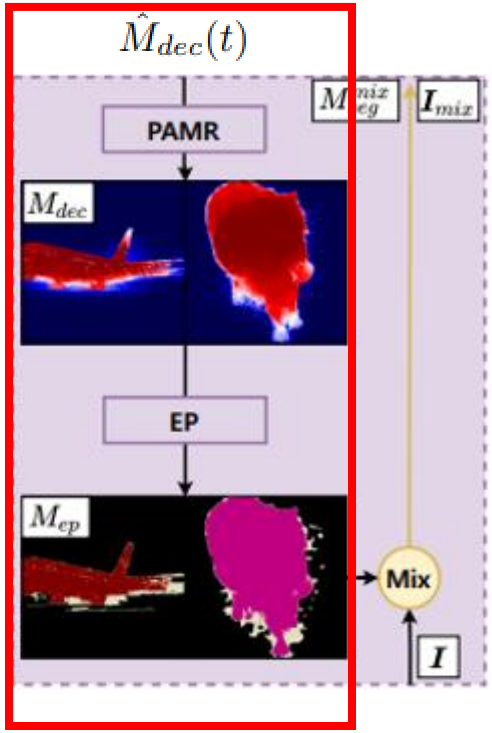
- 1. perform the mask refinement by using absolute and relative per-pixel probability values. (EP)
- 인접한 per-pixel probability values의 상대적 차이를 이용한다.
- Deriving edge from the mask, obtaining super pixels from the edge
- Singling out the most dominant class within each super pixel -> boundary aware mask
- Extract edge -> canny,   Extract super pixel -> Connected-component labeling (CCL)

$$M^i_{dec}(t) = PAMR\Big(\overbrace{\boxed{D_{\theta_t^{dec}}(E_{\theta_t^{enc}}(I_i)); \mathcal{W}}}^{\hat{M}_{dec}(t)}\Big)$$

$$\boxed{M^i_{ep}(t) = EP(M^i_{dec}(t))}$$

$$\boxed{\begin{aligned} &\mathcal{R}^i_c(t) = \{(k,n) \mid M^i_{ep}(t)[c,k,n] > \delta_{fg}\} \\ &\mathcal{M}^i_{fg} = \mathbb{1}\left[\cup_{c\in\mathcal{C}}\, \mathcal{R}^i_c(t)\right], \end{aligned}}$$

Extract the union of all EP-refined foregrounds for image



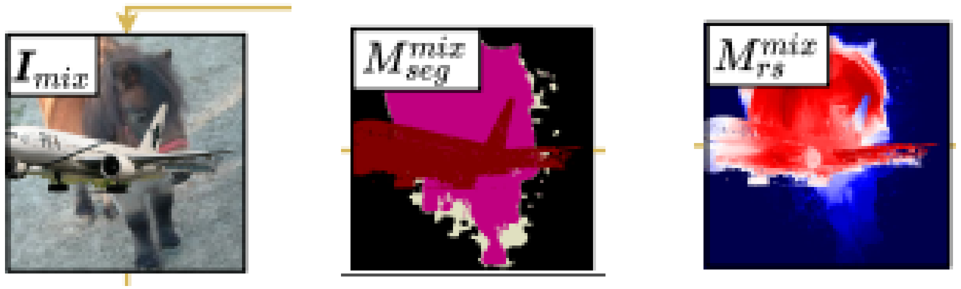
EdgePredictMix(EPM)

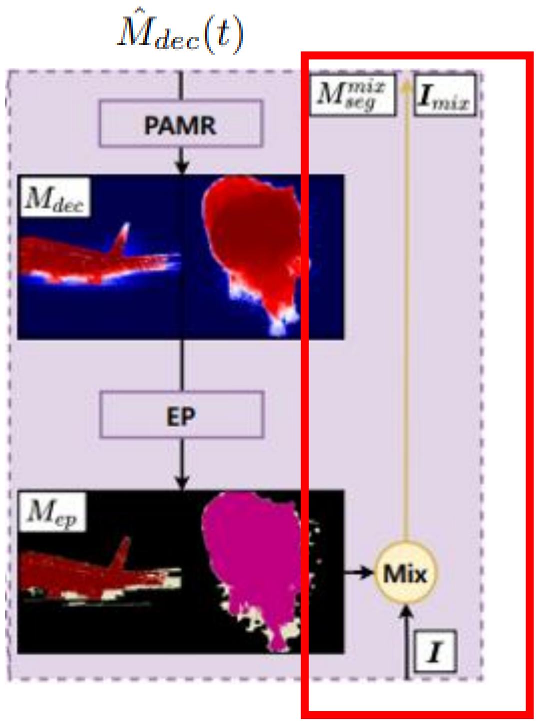
Overview

- 2. blend two EP-refined masks (and their corresponding original images)

-for two arbitrary indices i, j

$$I_{i\to j} = I_i \odot \mathcal{M}^i_{fg} + I_j \odot (1 - \mathcal{M}^i_{fg})$$
$$M^{seg}_{i\to j}(t) = M^i_{ep}(t) \odot \mathcal{M}^i_{fg} + M^j_{ep}(t) \odot (1 - \mathcal{M}^i_{fg})$$
$$M^{rs}_{i\to j}(t) = M^i_{rs}(t) \odot \mathcal{M}^i_{fg} + M^j_{rs}(t) \odot (1 - \mathcal{M}^i_{fg})$$





Loss function

- make the network train the mixed images and labels

$$\theta_t = \theta_{t-1} - (*) - \eta\frac{\partial}{\partial\theta}\mathbb{E}_{\boldsymbol{I}}\left[\mathcal{L}_{seg}^{mix}\left(\hat{M}_{dec}^{mix}(\tau), M_{seg}^{mix}(\tau);\theta\right)\right]\Big|_{\theta=\theta_\tau,\tau=t-1} - \eta\frac{\partial}{\partial\theta}\mathbb{E}_{\boldsymbol{I}}\left[\mathcal{L}_{rec}^{mix}\left(\hat{M}_{cam}^{mix}(\tau), M_{rs}^{mix}(\tau);\theta\right)\right]\Big|_{\theta=\theta_\tau,\tau=t-1},$$

$$\boxed{\begin{aligned}&\eta\frac{\partial}{\partial\theta}\mathbb{E}_{\boldsymbol{I}}\left[\mathcal{L}_{cls}\left(\hat{Y}_{cls}(\tau), \boldsymbol{Y};\theta\right)\right]\Big|_{\theta=\theta_\tau,\tau=t-1}\\ &\eta\frac{\partial}{\partial\theta}\mathbb{E}_{\boldsymbol{I}}\left[\mathcal{L}_{seg}\left(\hat{M}_{dec}(\tau), M_{seg}(\tau);\theta\right)\right]\Big|_{\theta=\theta_\tau,\tau=t-1}\\ &\eta\frac{\partial}{\partial\theta}\mathbb{E}_{\boldsymbol{I}}\left[\mathcal{L}_{rec}\left(\hat{M}_{cam}(\tau), M_{rs}(\tau);\theta\right)\right]\Big|_{\theta=\theta_\tau,\tau=t-1},\end{aligned}}$$

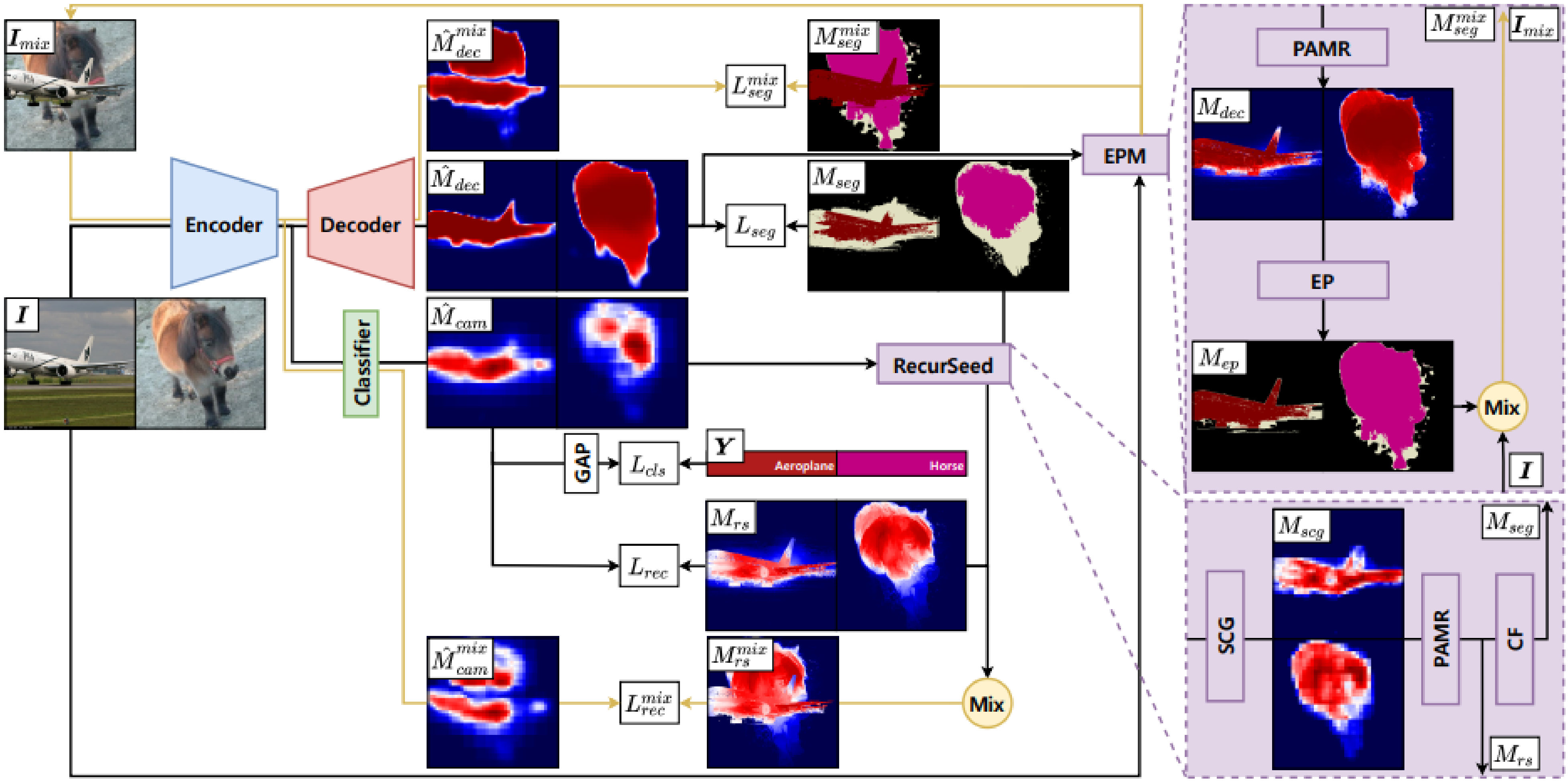
= RecurSeed Loss function

= Cross Entropy Loss

$$\begin{aligned}\boldsymbol{I}_{mix} &= \left\{I_{i\to j}\,\middle|\, j\sim \text{unif}(\{1:B\}/i) \text{ for } i\in\{1:B\}\right\}\\ M_{seg}^{mix}(t) &= \left\{M_{i\to j}^{seg}(t)\,\middle|\, I_{i\to j}\in \boldsymbol{I}_{mix}\right\}\\ \hat{M}_{dec}^{mix}(t) &= D_{\theta_t^{dec}}\left(E_{\theta_t^{cam}}(\boldsymbol{I}_{mix})\right)\\ M_{rs}^{mix}(t) &= \left\{M_{i\to j}^{rs}(t)\,\middle|\, I_{i\to j}\in \boldsymbol{I}_{mix}\right\}\\ \hat{M}_{cam}^{mix}(t) &= A_{\theta_t^{cls}}\left(E_{\theta_t^{cam}}(\boldsymbol{I}_{mix})\right)\end{aligned}$$

DEEP NOID

Overview



Result

Method	Backbone	Sup.	VOC val	VOC test	COCO val
Single stage:					
EM ICCV'15 [17]	VGG16	$\mathcal{I}$	38.2	39.6	-
RRM AAAI'20 [19]	WR38	$\mathcal{I}$	62.6	62.9	-
SSSS CVPR'20 [20]	WR38	$\mathcal{I}$	62.7	64.3	-
AFA CVPR'22 [43]	MiT-B1	$\mathcal{I}$	66.0	66.3	38.9
Ours (single-stage, RS)	R50	$\mathcal{I}$	66.5	67.9	40.0
Ours (single-stage, RS+EPM)	R50	$\mathcal{I}$	**69.5**	**70.6**	**42.2**
Multiple stages:					
DSRG CVPR'18 [22]	R101	$\mathcal{I}$+$\mathcal{S}$	61.4	63.2	26.0*
FickleNet CVPR'19 [6]	R101	$\mathcal{I}$+$\mathcal{S}$	64.9	65.3	-
NSRM CVPR'21 [51]	R101	$\mathcal{I}$+$\mathcal{S}$	68.3	68.5	-
AuxSegNet ICCV'21 [44]	WR38	$\mathcal{I}$+$\mathcal{S}$	69.0	68.6	33.9
EDAM CVPR'21 [33]	R101	$\mathcal{I}$+$\mathcal{S}$	70.9	70.6	-
DRS AAAI'21 [52]	R101	$\mathcal{I}$+$\mathcal{S}$	71.2	71.4	-
CLIMS CVPR'22 [34]	R50	$\mathcal{I}$+$\mathcal{D}$	69.3	68.7	-
W-OoD CVPR'22 [45]	R101	$\mathcal{I}$+$\mathcal{D}$	70.7	70.1	-
EPS CVPR'21 [53]	R101	$\mathcal{I}$+$\mathcal{S}$	70.9	70.8	35.7*
L2G CVPR'22 [14]	R101	$\mathcal{I}$+$\mathcal{S}$	72.1	71.7	44.2
RCA CVPR'22 [12]	R101	$\mathcal{I}$+$\mathcal{S}$	72.2	72.8	36.8*
PPC CVPR'22 [35]	R101	$\mathcal{I}$+$\mathcal{S}$	72.6	**73.6**	-
PSA CVPR'18 [5]	WR38	$\mathcal{I}$	61.7	63.7	-
IRNet CVPR'19 [31]	R50	$\mathcal{I}$	63.5	64.8	-
SEAM CVPR'20 [29]	WR38	$\mathcal{I}$	64.5	65.7	31.9
AdvCAM CVPR'21 [7]	R101	$\mathcal{I}$	68.1	68.0	-
CSE ICCV'21 [54]	WR38	$\mathcal{I}$	68.4	68.2	36.4
CPN ICCV'21 [30]	WR38	$\mathcal{I}$	67.8	68.5	-
RIB NIPS'21 [32]	R101	$\mathcal{I}$	68.3	68.6	43.8
ReCAM CVPR'22 [55]	R101	$\mathcal{I}$	68.5	68.4	-
ADEHE CVPR'22 [41]	R101	$\mathcal{I}$	68.6	68.9	-
AMR AAAI'22 [56]	R101	$\mathcal{I}$	68.8	69.1	-
URN AAAI'22 [57]	R101	$\mathcal{I}$	69.5	69.7	40.7
SIPE CVPR'22 [13]	R101	$\mathcal{I}$	68.8	69.7	40.6
AMN CVPR'22 [36]	R101	$\mathcal{I}$	69.5	69.6	44.7
MCTformer CVPR'22 [39]	WR38	$\mathcal{I}$	71.9	71.6	42.0
SANCE CVPR'22 [42]	R101	$\mathcal{I}$	70.9	72.2	44.7†
Ours (multi-stage, RS)	R101	$\mathcal{I}$	72.8	72.8	45.8
Ours (multi-stage, RS+EPM)	R101	$\mathcal{I}$	**74.4**	**73.6**	**46.4**

RS	SCG	PAMR	mIoU	FP	FN
	✓		58.0	0.268	0.165
	✓	✓	59.3	0.225 ($\downarrow$ 0.043)	0.194 ($\uparrow$ 0.029)
✓	✓		65.2	0.216	0.143
✓	✓	✓	65.9	0.210 ($\downarrow$ 0.006)	0.141 ($\downarrow$ 0.002)
✓	✓		*67.4	*0.196	*0.141
✓	✓	✓	*70.7	*0.171 ($\downarrow$ 0.025)	*0.134 ($\downarrow$ 0.007)

$*$ denotes the decoder map result.

Method	Backbone	F_1	mIoU
RecurSeed	R50	94.7	70.7
RecurSeed + *CutMix [25]	R50	95.6	68.5
RecurSeed + *SaliencyGrafting [26]	R50	**96.8**	68.6
RecurSeed + *CDA [27]	R50	96.0	69.0
RecurSeed + *ClassMix [28]	R50	94.6	71.2
RecurSeed + EdgePredictMix	R50	95.2	**75.2**

Result

Method	Backbone	CAM (%)	CAM +RW (%)	CAM+RW +dCRF (%)
AffinityNet [4]	ResNet-50	47.82	58.10	59.70
Puzzle-CAM	ResNet-50	51.53	64.16	64.70
Puzzle-CAM	ResNeSt-50	57.59	69.48	69.91
Puzzle-CAM	ResNeSt-101	61.85	71.92	72.46
Puzzle-CAM	ResNeSt-269	**62.45**	**74.14**	**74.67**

Puzzle-CAM with MLF

Method	Backbone	Seed	RW
SEAM [29]	WR38	55.4	63.6
IRNet [31]	R50	48.8	66.3
CSE [54]	WR38	56.0	66.9
CDA [27]	R50	50.8	67.7
CPN [30]	WR38	57.4	67.8
CONTA [58]	R50	48.8	67.9
AMR [56]	R50	56.8	69.7
AdvCAM [7]	R50	55.6	69.9
PPC [35]	WR38	61.5	70.1
RIB [32]	R50	56.5	70.6
Ours (RS)	R50	70.7	74.8
Ours (RS+EPM)	R50	**75.2**	**76.7**

RS & EPM with MLF

Conclusion

- WSSS에서 PASCAL VOC 2012, MS COCO 2014 benchmark SOTA 달성
- RS 와 EPM은 encoder 와 decoder을 포함한 SLF에 범용적으로 적용할 수 있으며, 더 높은 성능을 위해 Backbone을 Upgrade 할 수 있다.

D E E P N O I D

Reference

- **Learning Deep Features for Discriminative Localization**

Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, Antonio Torralba Computer Science and Artificial Intelligence Laboratory, MIT

- **PUZZLE-CAM: IMPROVED LOCALIZATION VIA MATCHING PARTIAL AND FULL FEATURES**

Sanghyun Jo GYNetworks In-Jae Yu∗ KAIST School of Computing

- **RecurSeed and EdgePredictMix: Single-stage Learning is Sufficient for Weakly-Supervised Semantic Segmentation**

Sanghyun Jo1∗, In-Jae Yu2∗, Kyungsu Kim3,4†

- **Unveiling the Potential of Structure Preserving for Weakly Supervised Object Localization**

Xingjia Pan1,3∗ Yingguo Gao1∗ Zhiwen Lin1∗ Fan Tang2† Weiming Dong3,4,5 Haolei Yuan1 Feiyue Huang1 Changsheng Xu3,4,5

- **Single-Stage Semantic Segmentation from Image Labels**

Nikita Araslanov Stefan Roth

Q & A

감사합니다

DEEP NOID