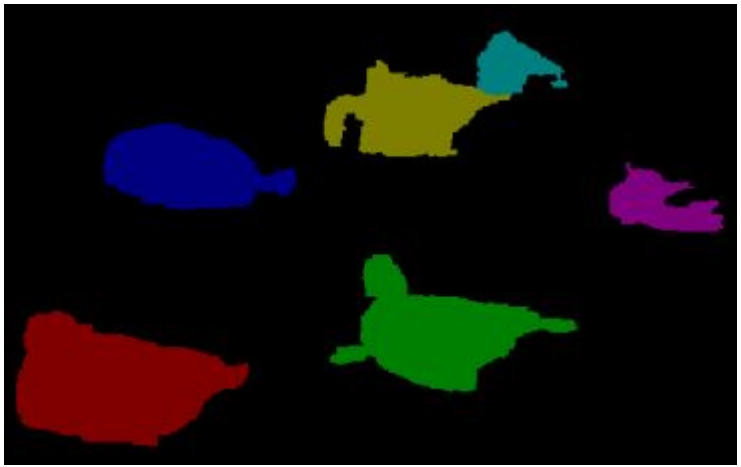
DeepCO^3: Deep Instance Co-segmentation by Co-peak Search and Co-saliency Detection

Presented by Uzair Inamdar, Aakaash Kapoor, and Matthew Kotila

Problem

Is it possible to find all ducks in this image?



Related Work

Object co-segmentation

What it does?

- Aims to segment common objects in images.

Cons

- Not instance aware
- Simple issues with objects converging into one.

Object co-localization

What it does?

- Instance aware results.

Cons

- Only one top instance.
- But only bounding boxes.
- Not pixel level image segmentation.

Instance-aware class specific

What it does?

- Instance aware and object segmentation.

Cons

- Only work with classes it has trained with.

Instance Aware Class agnostic

What it does?

- Instance aware and object segmentation and does not care about classes.

Cons

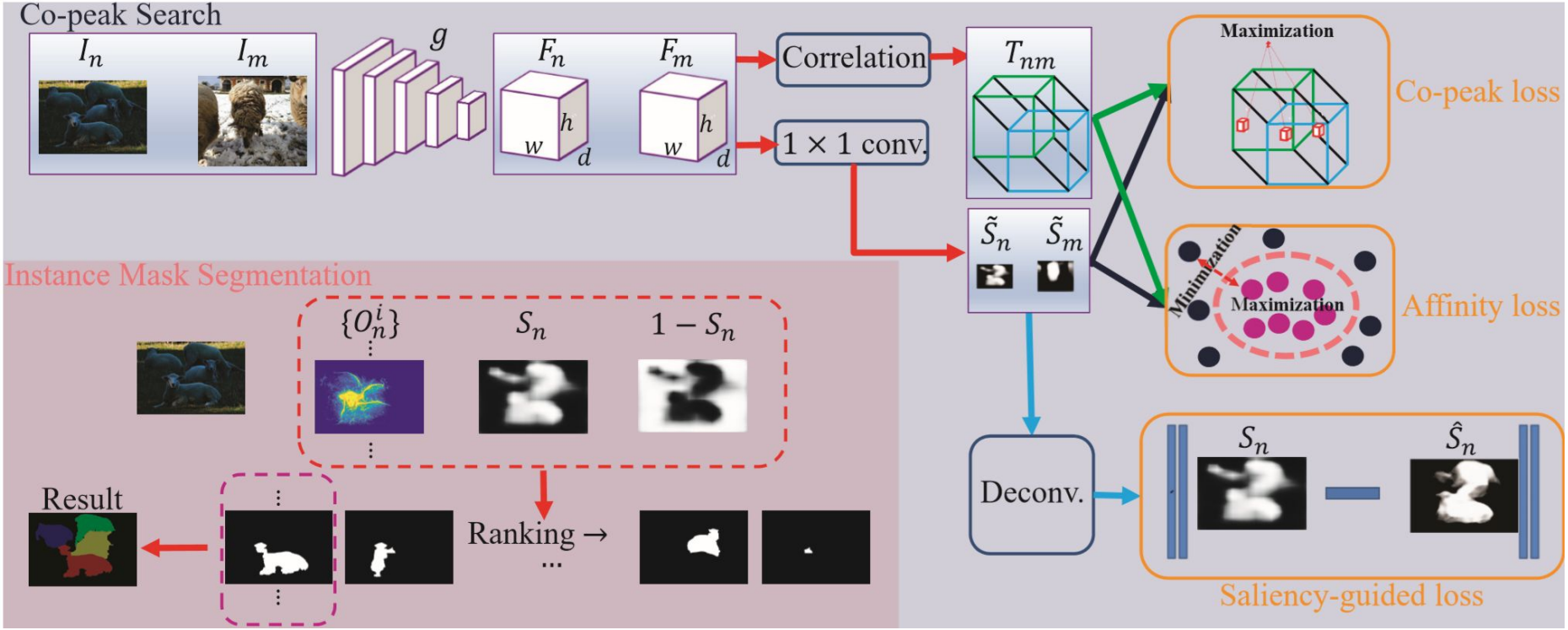
- Needs a lot of time and effort to train and is weakly supervised.

Contribution

- First, we introduce a new and interesting task called instance co-segmentation which is instance aware class agnostic.
- Second, a simple and effective method is developed for instance co-segmentation.
- Third, we collect four datasets for evaluating instance co-segmentation

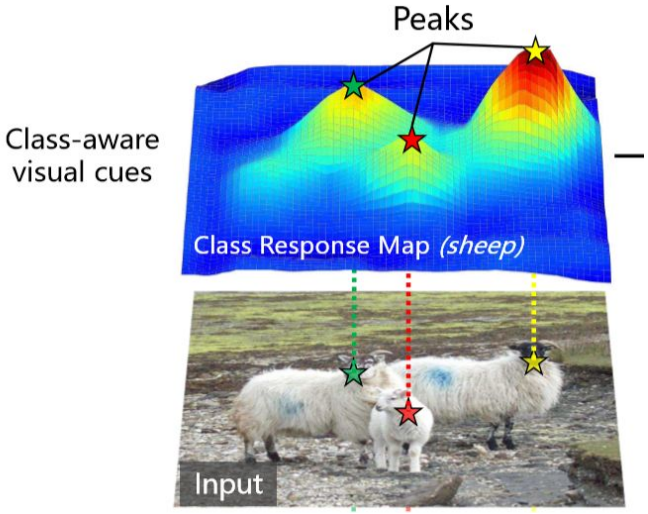
Approach

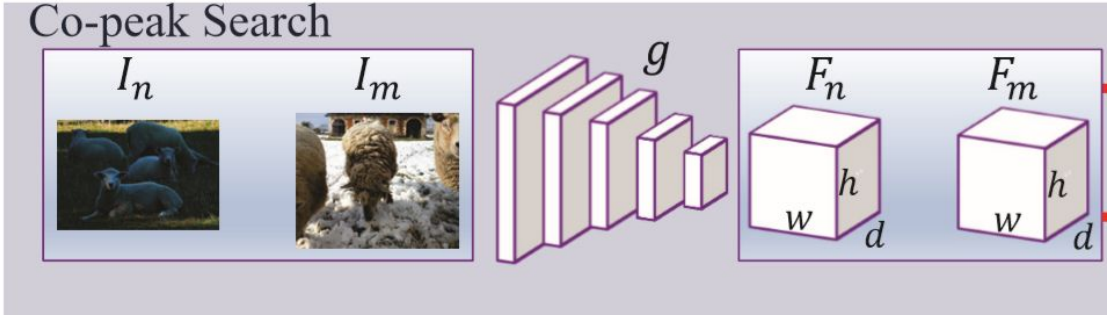
Overview


Co-peak Search
I_n
I_m
g
F_n
F_m
w
h
d
Correlation
1×1 conv.
T_{nm}
$\tilde{S}_n$
$\tilde{S}_m$
Maximization
Co-peak loss
Minimization
Maximization
Affinity loss
Instance Mask Segmentation
$\{O_n^i\}$
S_n
$1 - S_n$
Result
Ranking →
...
Deconv.
S_n
$\hat{S}_n$
Saliency-guided loss

Co-peak Search

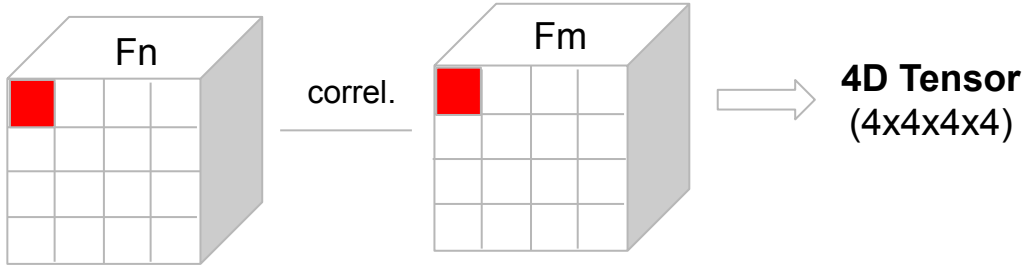
- Peaks
- Co-peaks Search





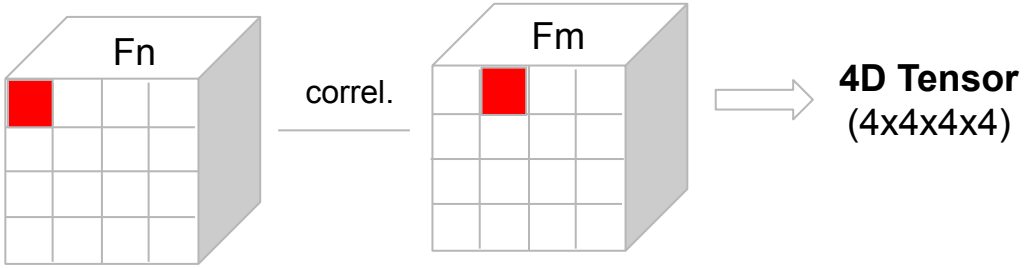
Co-peak Search

- The 2 streams/paths
- Correlation Tensor
- Saliency and co-saliency



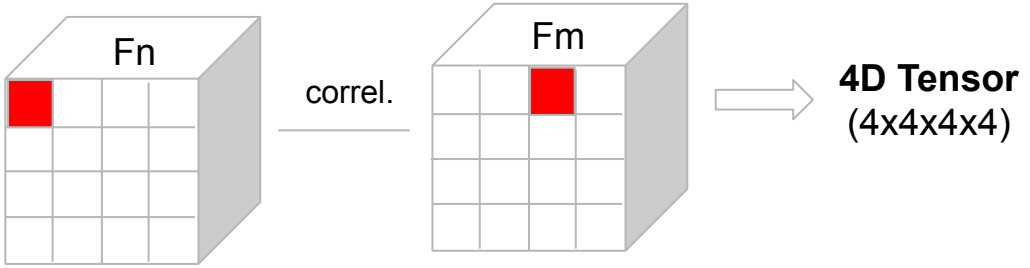
Co-peak Search

- The 2 streams/paths
- Correlation Tensor
- Saliency and co-saliency



Co-peak Search

- The 2 streams/paths
- Correlation Tensor
- Saliency and co-saliency



Co-peak Loss (l_t)

- Saliency boosted co-peak correlation, where p and q corresponding cells from the feature maps that are being correlated.

$$T^s_{nm}(\mathbf{p},\mathbf{q}) = \tilde{S}_n(\mathbf{p})\tilde{S}_m(\mathbf{q})T_{nm}(\mathbf{p},\mathbf{q})$$

- The co-peak loss function, where M_{nm} is the set of co-peaks.

$$\ell_t(I_n, I_m) = -log\left(\frac{1}{|\mathcal{M}_{nm}|}\sum_{(\mathbf{p},\mathbf{q})\in\mathcal{M}_{nm}} T^s_{nm}(\mathbf{p},\mathbf{q})\right)$$

Affinity Loss

- Aims to learn to find similar pixels in both images that also most distinct (salient).

$$\tilde{\ell}_a(I_n, I_m) = \sum_{\mathbf{p}\in\mathcal{P}} \sum_{\mathbf{q}\in\mathcal{P}} \tilde{S}_n(\mathbf{p})\tilde{S}_n(\mathbf{q})\left(1 - T_{nm}(\mathbf{p},\mathbf{q})\right) + \alpha(\tilde{S}_n(\mathbf{p}) - \tilde{S}_n(\mathbf{q}))^2 T_{nm}(\mathbf{p},\mathbf{q})$$

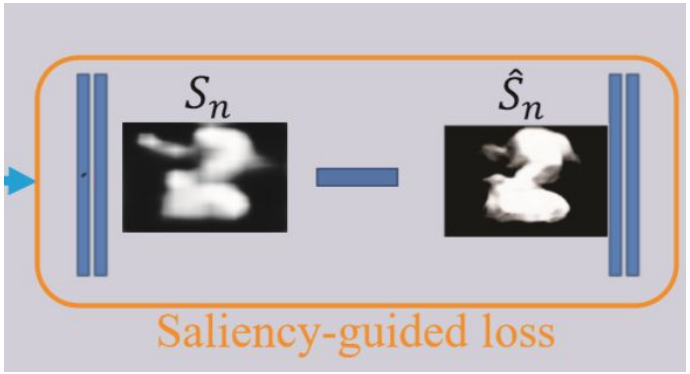
- Generalized Loss:

$$\ell_a(I_n, I_m) = \tilde{\ell}_a(I_n, I_m) + \tilde{\ell}_a(I_n, I_n) + \tilde{\ell}_a(I_m, I_m).$$

Saliency Loss

- Tries to learn to find salient pixels that represent the object pixels and stand out from the background pixels.
- Off the shelf methods.

$$\ell_s(I_n) = \sum_{\mathbf{p} \in I_n} \rho_n(\mathbf{p}) \| S_n(\mathbf{p}) - \hat{S}_n(\mathbf{p}) \|_2^2,$$

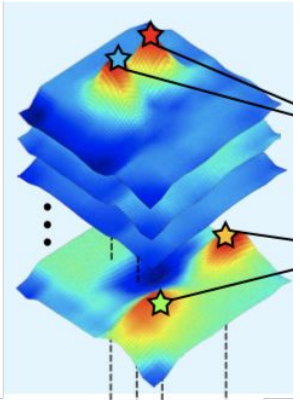


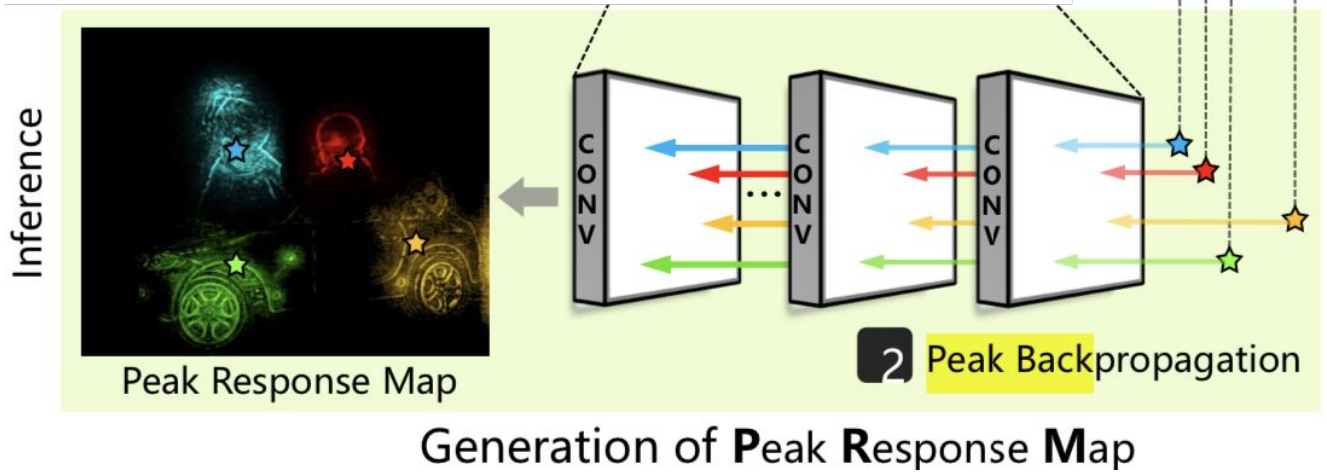
Final Objective Function

$$\mathcal{L}(\mathbf{w}) = \lambda_t \sum_{n=1}^{N} \sum_{m \neq n} \ell_t(I_n, I_m; \mathbf{w}) + \lambda_a \sum_{n=1}^{N} \sum_{m \neq n} \ell_a(I_n, I_m; \mathbf{w}) + \sum_{n=1}^{N} \ell_s(I_n; \mathbf{w}),$$

Instance Mask Segmentation

1. Peak back-propogation -> Heatmap
2. Multi-scale combinatorial grouping -> Instance Proposals
3. Heatmap + Co-Saliency Map -> Rank Proposals
4. 1 Top Instance per Peak





Instance Mask Segmentation - Rank Function

$$R(P) = \beta(O_n^i * S_n) * P + (O_n^i * S_n) * \hat{P} - \gamma(1 - S_n) * P,$$

How close whole instance proposal is to heatmap+saliency

How close just contour of instance proposal is to heatmap+saliency

Penalize non-salient instance proposals

Implementation Details

- Used Matlab (MatConvNet)
- Initial training: VGG16 + ImageNet
- Additional training: 3 custom loss functions
- ADAM optimizer

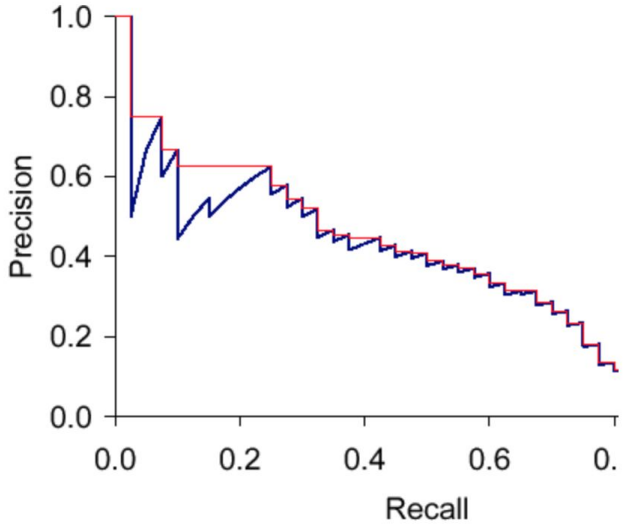
Experimental Setup

Dataset

- MS COCO dataset -> COCO-VOC and COCO-NONVOC
- PASCAL VOC dataset
- SOC dataset
- Removing the images where objects of more than one category are present.
- Second, discard the categories that contain less than 10 images.

Evaluation : Mean Average Precision

- For different values of threshold you get different precision and recall.
- Goal is to maximize the area under the precision recall curve.
- **Instance co-segmentation**.



The area under this Precision-Recall curve gives you the “Average Precision”.

Evaluation: CorLoc

- The correct localization metric is defined as the percentage of images correctly localized according to the PASCAL criterion.
- Pascal criterion is

$$\frac{area(b_p \cap b_{gt})}{area(b_p \cup b_{gt})} > 0.5$$

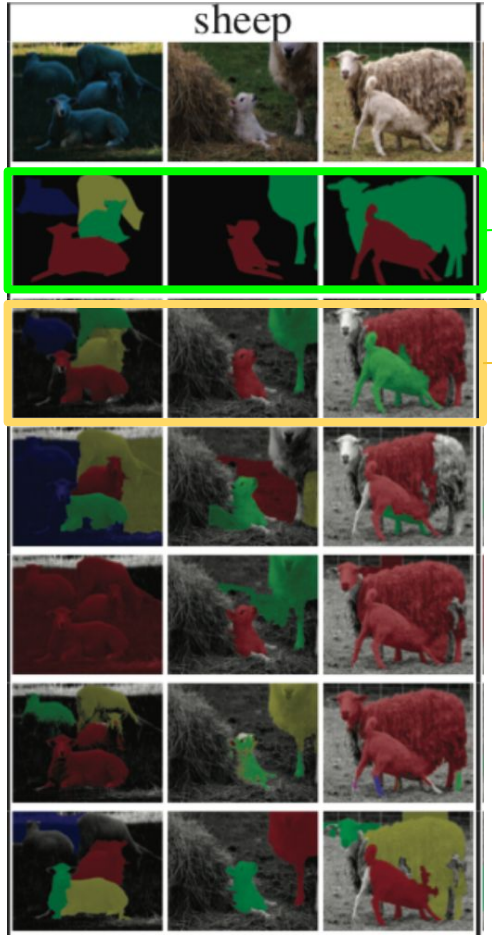
Where where bp is the predicted box and bgt is the ground-truth box.

- **For Object Co-localization**

What to compare with?

- Object co-localization
  - **CLRW, UODL, DDT, DDT and DFF**
- Class-agnostic saliency segmentation
  - **NLDF and C2S-Net**
- Weakly supervised instance segmentation.
  - **PRM**

Experimental Results



Ground Truth

Our Method

Object Cosegmentation

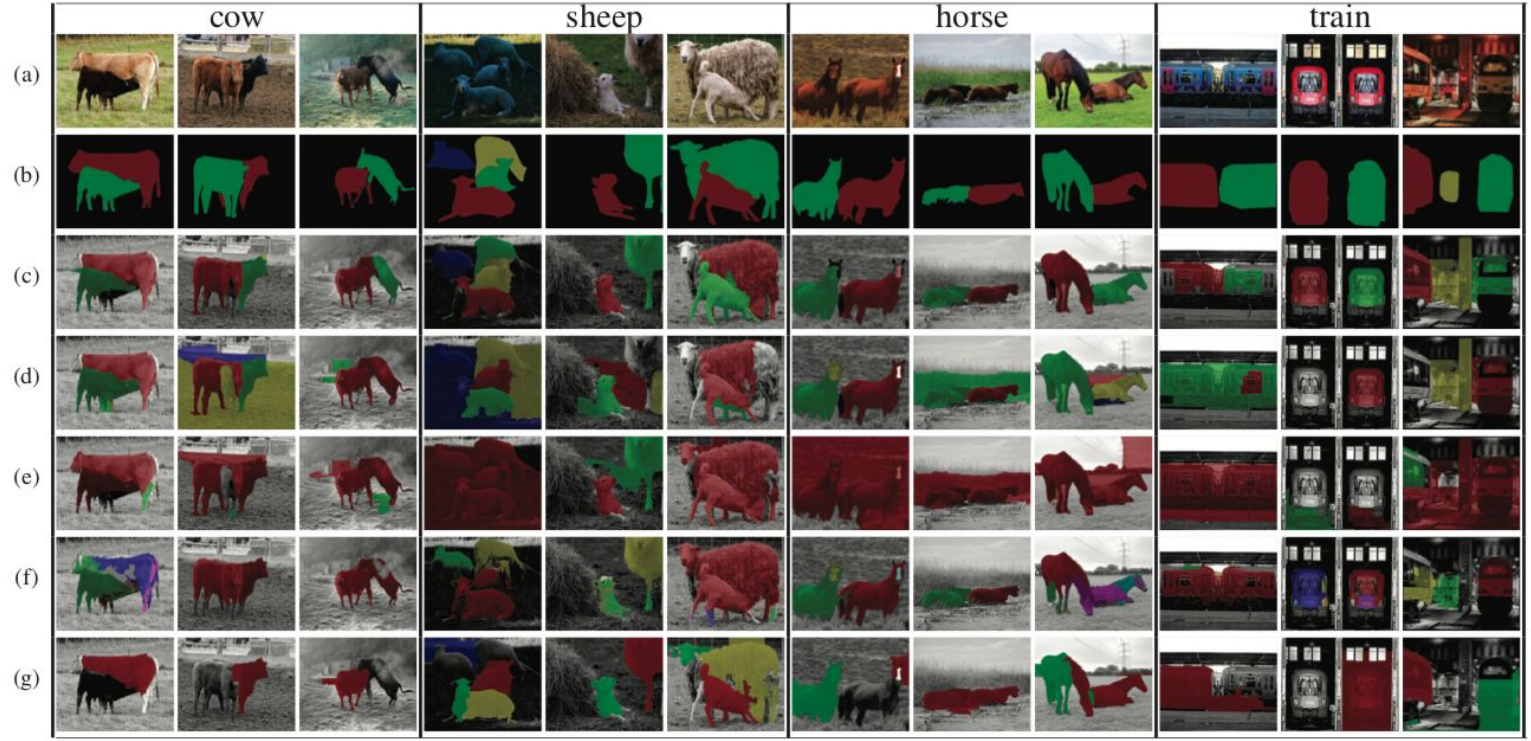


Figure 3. Results of instance co-segmentation on four object categories, *i.e.cow*, *sheep*, *horse*, and *train*, of the COCO-VOC dataset. (a) Input images. (b) Ground truth. (c) ∼ (g) Results with instance-specific coloring generated by different methods including (c) our method, (d) CLRW [51], (e) DFF [6], (f) NLDF [41], and (g) PRM [65], respectively.

method	year	trained	COCO-VOC		COCO-NONVOC		VOC12		SOC	
			$mAP^r_{0.25}$	$mAP^r_{0.5}$	$mAP^r_{0.25}$	$mAP^r_{0.5}$	$mAP^r_{0.25}$	$mAP^r_{0.5}$	$mAP^r_{0.25}$	$mAP^r_{0.5}$
CLRW [51]	CVPR 2014	×	33.3	13.7	24.6	10.7	29.2	10.5	34.9	15.6
UODL [5]	CVPR 2015	×	9.6	2.2	8.5	1.8	9.4	2.0	11.0	2.7
DDT [58]	IJCAI 2017	×	31.4	10.1	25.7	9.7	30.7	8.8	43.0	25.7
DDT+ [59]	arXiv 2017	×	31.7	10.6	26.0	10.1	33.6	9.4	39.6	22.4
DFF [6]	ECCV 2018	×	30.8	11.6	22.6	7.3	27.7	13.7	42.3	17.0
NLDF [41]	CVPR 2017	√	39.1	18.2	23.9	8.5	34.3	12.7	49.5	21.6
C2S-Net [34]	ECCV 2018	√	39.6	13.4	25.1	7.6	30.1	10.7	37.0	12.5
PRM [65]	CVPR 2018	√	44.9	14.6	-	-	45.3	14.8	-	-
Ours	-	×	52.6	21.1	35.3	12.3	45.6	16.7	54.2	26.0

Table 2. Performance of instance co-segmentation on the four collected datasets. The numbers in red and green show the best and the second best results, respectively. The column "trained" indicates whether additional training data are used.

Figure 5. Seven examples, one in each row, of the co-localization results by our method on the COCO-NONVOC dataset.

Object Co-localization

method	year	trained	COCO-VOC	COCO-NONVOC	VOC12	SOC
CLRW [51]	CVPR 2014	×	33.4	31.6	29.9	30.9
UODL [5]	CVPR 2015	×	12.3	12.7	9.5	10.3
DDT [58]	IJCAI 2017	×	30.0	27.4	25.0	16.7
DDT+ [59]	PR 2019	×	29.5	25.8	23.7	18.4
DFF [6]	ECCV 2018	×	32.3	30.5	28.7	22.9
NLDF [41]	CVPR 2017	√	51.2	31.0	39.2	42.0
C2S-Net [34]	ECCV 2018	√	39.0	28.4	31.1	32.9
PRM [65]	CVPR 2018	√	18.1	-	23.3	-
Ours	-	×	49.6	34.3	39.2	43.1

Table 4. Performance of object co-localization on the four datasets. The numbers in red and green indicate the best and the second best results, respectively. The column "trained" indicates whether additional training data are used.

Strengths

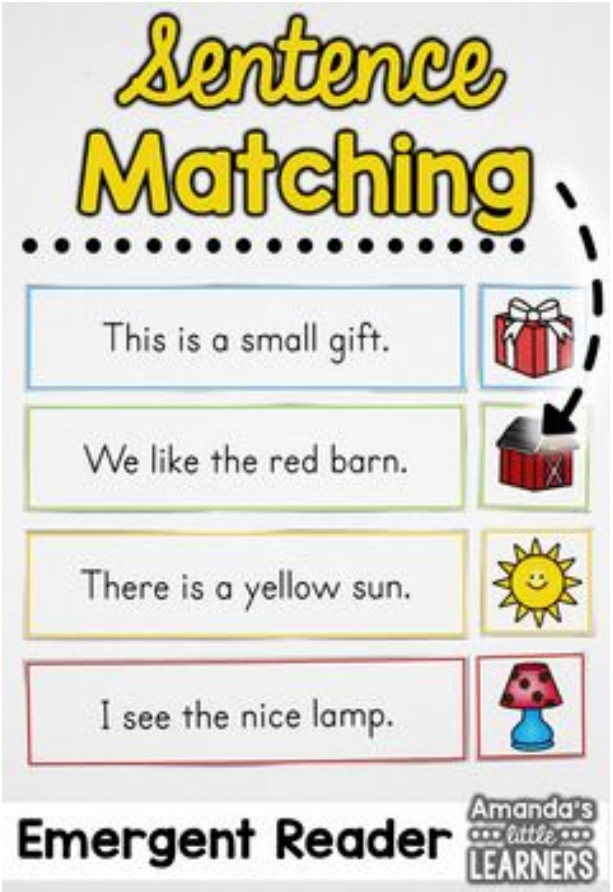
- Minimal annotations needed in training data
- First to do instance CO-segmentation
- Performs better than state-of-the-art instance segmentation

Weaknesses

- Instance co-segmentation doesn’t tell us the class
- Some of their coined terms hard to understand

Applications/Future

- Autonomous driving
- Visual question-answering
- Image and sentence matching



Questions?

Thank You 😊