Supplementary - Unsupervised Object Localization: Observing the Background to Discover Objects

Oriane Siméoni [1], Chloé Sekkat[1], Gilles Puy[1], Antonin Vobecky[1,2], Éloi Zablocki[1], Patrick Pérez[1]
[1]valeo.ai, Paris, France
[2]Czech Institute of Informatics, Robotics and Cybernetics, CTU, Prague, Czech Republic

Contents

A. Extra details

A.1. During learning

During training $\zeta()$ is applied at the image resolution. To do so, masks are upsampled to the original image size and the output refined masks are downsampled to the feature map size. The model is trained with the AdamW optimizer provided by PyTorch, with an initial learning rate of $5\mathrm{e}{-2}$. We use a simple step scheduler which applies a decay of 0.95 every 50 iterations.

A.2. Unsupervised saliency detection

We detail here the different metrics used in the task of unsupervised saliency detection.

The maximal F_β metric is the maximum F_β over various masks which have been binarized using different thresholds. Formally, F_β is the harmonic mean of precision (P) and recall (R) between a binary mask M and the ground-truth mask G, i.e.,

$$F_\beta = \frac{(1+\beta^2)\,\mathrm{P}\times\mathrm{R}}{\beta^2\,\mathrm{P}+\mathrm{R}}, \tag{1}$$

where β^2 is the precision weight, set at 0.3 following [6, 7, 9, 15]. The max F_β is computed by taking a soft predicted mask $M_p \in [0, 255]$ and binarizing it using 255 different thresholds between 0 and 254; max F_β is then the maximum value of F_β among all the generated binary masks, taken over the whole dataset (single optimal threshold). We noticed in SelfMask's code that the maximal F_β is computed with an optimal threshold found for each image rather than over the whole dataset. For this reason, and for a fair comparison, we do not report this original max F_β in our unsupervised saliency detection table.

The Intersection-over-Union measures the overlap between foreground regions of a predicted binary mask and the ground-truth mask, averaged over the entire dataset.

The pixel accuracy metric measures the pixel-wise accuracy between a predicted binary mask $M \in \{0,1\}^{H\times W}$ and the corresponding ground-truth mask $G \in \{0,1\}^{H\times W}$. Formally, it can be defined as:

$$\mathrm{Acc} = \frac{1}{H\times W}\sum_{i=1}^{H}\sum_{j=1}^{W}\delta_{G_{ij},M_{ij}}, \tag{2}$$

with δ being the Kroneker-delta function and G_{ij}, M_{ij} being the value of the ground-truth and predicted masks at position $(i,j) \in \{1\cdots H\}\times\{1\ldots W\}$.

A.3. Different setups for FreeSOLO

FreeSOLO [13] is a class-agnostic instance segmentation method and outputs several instance masks per image, making it different to other baselines. In order to compare it to our method, we use the code provided online. We follow

the original paper to get the prediction masks, i.e., we apply matrix non-maximum suppresion (NMS) [12] and keep masks with a maskness score above 0.7.

Unsupervised object discovery We present in Sec. 4.1 of the main paper our unsupervised object discovery protocol. The extraction of the single object box is straightforward for all methods but FreeSOLO [13]. For this method we have considered three setups: (a) merging all instance masks into a single one; (b) keeping only the mask with the highest maskness score; (c) keeping only the mask containing the largest connected component. Best results were achieved with (a) and are reported in the main paper.

Semantic segmentation retrieval We have performed similar tests with FreeSOLO in the semantic segmentation retrieval task. Additionally to the evaluation setups described in the main paper, we have experimented using two or more instances but without improvements of the results.

A.4. Semantic segmentation retrieval

In the task of unsupervised semantic segmentation retrieval, we consider two setups. One considers that the predicted mask highlights a single object, while the other splits the mask into connected components and treats each component as individual object. In both cases, we compute a per-object feature vector averaged over the pixels of the considered mask. Given a (flattened) binary mask $M \in \{0,1\}^{HW\times 1}$ and corresponding feature tensor $F \in \mathbb{R}^{C\times HW}$ with C the number of channels, we obtain a prototype $P \in \mathbb{R}^{C}$ as

$$P = FM. \tag{3}$$

These prototypes are first extracted for all train samples and serve as an index for retrieval. Then, to get a label for each val sample, we compute the sample prototype, find nearest neighbors in the train prototypes, and assign it the corresponding label.

B. Sensitivity to masking method

B.1. Sensitivity to background threshold τ

We investigate here the impact of the background parameter τ on final results. We report in Fig. 5 saliency detection results. We observe that FOUND is stable to changes of $\tau \in [0.1, 0.5]$, with saliency scores varying by at most 0.2 percentage pts on DUT-OMRON and not at all on ECSSD.

B.2. Using masks from other methods

We investigate here the performance of our method when considering different mask generators. In particular, we consider the well-known object discovery methods TokenCut [14] and LOST [10] with which we extract the masks

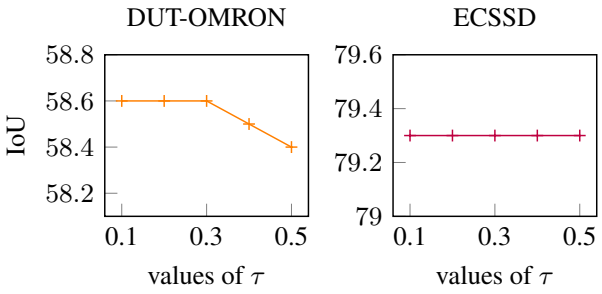


Figure 5. **Sensitivity to background threshold τ.** We report saliency detection results measured on the datasets DUT-OMRON (*left*) and ECSSD (*right*) with the IoU metric.

method	VOC07	VOC12	COCO20k
TokenCut [14] + T	72.3	75.9	62.7
LOST [10] + T	72.3	**76.1**	62.8
FOUND (ours)	**72.5**	**76.1**	**62.9**

Table 6. **Sensitivity to mask generation method.** Unsupervised object discovery results (measured using the CorLoc metric) when using different mask generation strategy to generate the masks $\mathbf{M}^f$ refined in our training process. T denotes the training of our segmentation head with the masks $\mathbf{M}^f$.

$\mathbf{M}^f$ that are then refined in our training process (following Sec. 3.2 of the main paper). We present the corresponding unsupervised object discovery results in Tab. 6. They show that our method is agnostic to the mask generator but still performs slightly better with our foreground masks — the complement of the background masks described in Sec. 3.1. It is also to be noted that our method is much faster than TokenCut because we do not need the computation of eigenvectors.

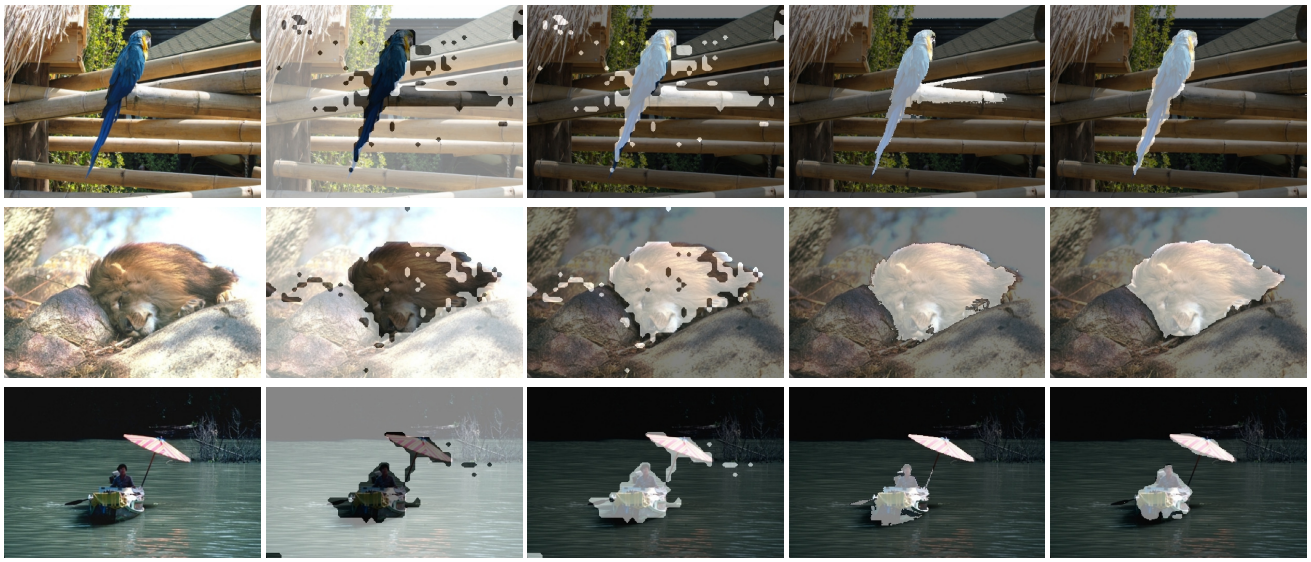
C. Additional qualitative results

We present in this section more visualizations of FOUND results, first on more challenging images (Sec. C.1) and at the different step of our process (Sec. C.2). We then motivate the interest of reweighting the transformer heads (Sec. C.3) via visual illustration. Following we show examples where the application of the bilateral solver impacts negatively the results (Sec. C.4) and some more general failure cases of FOUND (Sec. C.5). We finally provide example of discovered objects as performed in the task of unsupervised object discovery (Sec. C.6).

C.1. Results on generic images from the Internet

We present in Fig. 6 some results of FOUND random images taken from the Internet. These results show the ability of FOUND to discover multiple and diverse objects, both in terms of classes and scales. In particular, dinosaurs

Figure 6. **Visualization of FOUND results on images taken from the Internet**. Objects out of the domain of ImageNet [3] and DUT-TR [11] (datasets used for training the backbone and our segmentation head), of different scales, and of different shapes are correctly localized.



(a) Input image (b) Coarse background mask $\mathbf{M}^b$ (c) Coarse foreground mask $\mathbf{M}^f$ (d) Refined $\zeta(\mathbf{M}^f)$ (e) Predicted $\mathbf{M}^s$ (FOUND)

Figure 7. **More visualizations of masks generated on images from ECSSD [8] at different stages of our method**. We show (a) the input image, (b) the mask $\mathbf{M}^b$ extracted using our background discovery step, (c) its inverse $\mathbf{M}^f$ used as foreground mask to train our segmenter head, (d) the version refined using a bilateral solver $\zeta(\mathbf{M}^f)$, and (e) the final output of our trained segmentation head $\mathbf{M}^s$.

and spaceships are not depicted in ImageNet [3] nor DUT-TR [11] and yet FOUND can detect them, showing the ability to discover objects which "are not background." Moreover, the selected images here are non-object centric and out-of-domain showing the capacity of FOUND to go behond ImageNet-like images.

C.2. Visualization of masks at different steps

We provide in Fig. 7 additional visualizations of the masks generated at different steps of our method. We can observe that each step brings an improvement over the previous one. The right-most column presents the final output of FOUND without any refinement.

C.3. Reweighting the attention heads

We provide in Fig. 8 a visualization of the self-attention maps extracted from the last layer of our model. We show the self-attention obtained over the six heads; we can observe that the 4^{th} head is noisy. When looking for the background *seed*, we are looking for the pixel with least attention. Our reweighting scheme helps in reducing the weight given to such noisy heads automatically and improves re-

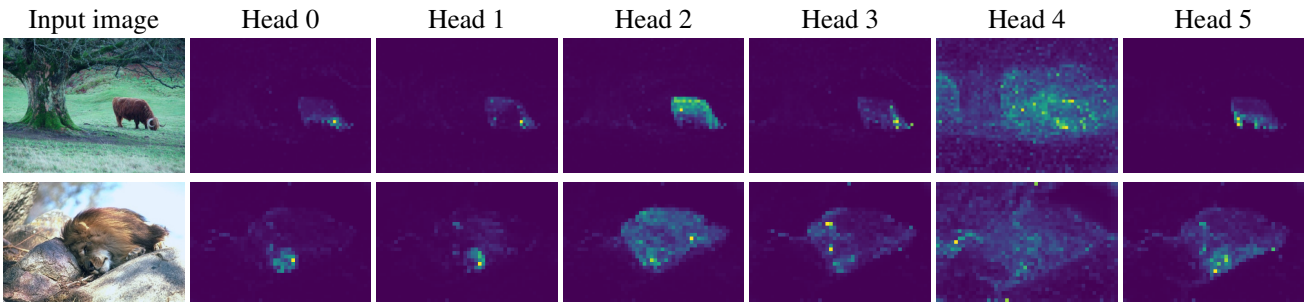


Figure 8. **Visualization of self-attention maps** obtained with the six different heads in the last attention layer. Results are obtained with a ViT/S-8 trained using DINO [2] applied on an image from VOC07 [4] (first row) and ECSSD [8].

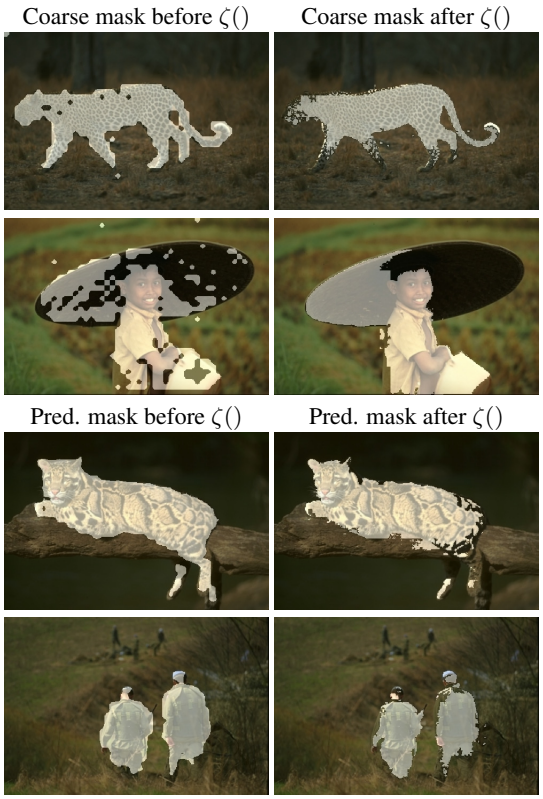


Figure 9. **Visualization of the negative impact of the bilateral solver** on different ECSSD [8] images.

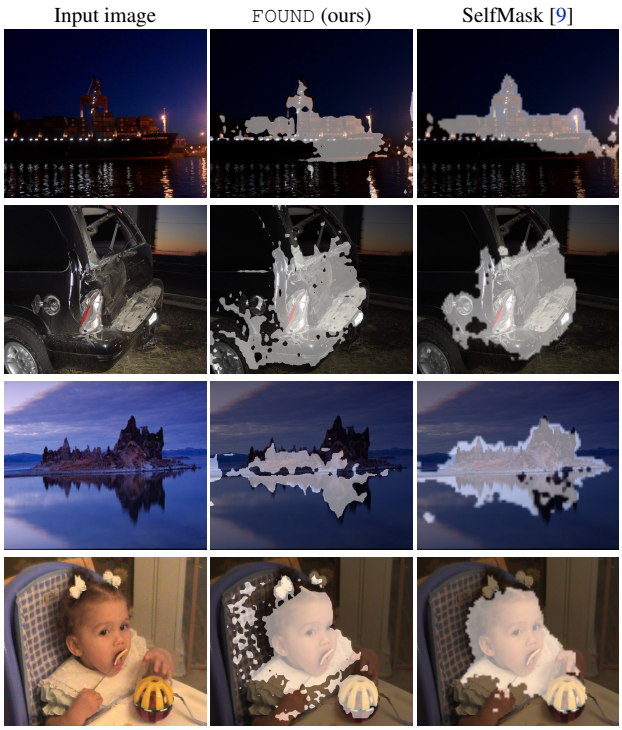


Figure 10. **Visualization of failure cases for object localization** on images from ECSSD [8], PASCAL VOC07 & VOC12 [4,5] and DUTS-TE [11] datasets along side results obtained with SelfMask method. No refinement step is applied.

sults, as shown in Tab. 5 of the main paper.

C.4. Potential negative effect of the bilateral solver

While the application of $\zeta()$, the bilateral solver [1], improves results in general (see Fig. 7), there are cases where $\zeta()$ actually degrades the mask quality. We show examples of such cases in Fig. 9 both on coarse masks (rows 1 and 2) and on the final outputs (rows 3 and 4). We can observe that the function amplifies the under-segmentation, *e.g.*, on the hat and the leopard head and legs (row 1 and 2). Moreover, long and thin segments can disappear, *e.g.*, human and animal legs or arms (row 3). Correcting this behaviour would help improving our training and is left for future work.

C.5. Examples of failures cases

We show some failure cases of FOUND in Fig. 10. For these cases, we also present the results obtained with one of the best competitor: SelfMask [9]. We observe that night or dark scenes are challenging (first two rows). Our method tends to under-segment objects but SelfMask has also dif-

ficulties in segmenting correctly the main objects in these situation. FOUND, just like SelfMask, is also not robust to reflection on water (third row). Finally, we observe that both methods fail to segment the hair in the fourth columns.

C.6. Unsupervised object discovery results

We present in Fig. 11, qualitative results for the unsupervised single object discovery task (no refinement is applied to the masks). We draw the extracted bounding box on top of the corresponding predicted mask. The conclusions here are similar to those discussed in the main paper. Overall our method segments the objects of interest better and provides cleaner boundaries.

References

[1] Jonathan T. Barron and Ben Poole. The fast bilateral solver. In *ECCV*, 2016. 4

[2] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021. 4

[3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009. 3

[4] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html, 2007. 4

[5] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html, 2012. 4, 6

[6] Jiang-Jiang Liu, Qibin Hou, Ming-Ming Cheng, Jiashi Feng, and Jianmin Jiang. A simple pooling-based design for real-time salient object detection. In *CVPR*, 2019. 1

[7] Luke Melas-Kyriazi, Christian Rupprecht, Iro Laina, and Andrea Vedaldi. Finding an unsupervised image segmenter in each of your deep generative models. In *ICLR*, 2022. 1

[8] Jianping Shi, Qiong Yan, Li Xu, and Jiaya Jia. Hierarchical image saliency detection on extended CSSD. *IEEE TPAMI*, 2016. 3, 4

[9] Gyungin Shin, Samuel Albanie, and Weidi Xie. Unsupervised salient object detection with spectral cluster voting. In *CVPRW*, 2022. 1, 4, 6

[10] Oriane Siméoni, Gilles Puy, Huy V. Vo, Simon Roburin, Spyros Gidaris, Andrei Bursuc, Patrick Pérez, Renaud Marlet, and Jean Ponce. Localizing objects with self-supervised transformers and no labels. In *BMVC*, 2021. 2

[11] Lijun Wang, Huchuan Lu, Yifan Wang, Mengyang Feng, Dong Wang, Baocai Yin, and Xiang Ruan. Learning to detect salient objects with image-level supervision. In *CVPR*, 2017. 3, 4

[12] Xinlong Wang, Tao Kong, Chunhua Shen, Yuning Jiang, and Lei Li. Solo: Segmenting objects by locations. In *ECCV*, 2020. 2

[13] Xinlong Wang, Zhiding Yu, Shalini De Mello, Jan Kautz, Anima Anandkumar, Chunhua Shen, and Jose M. Alvarez. Freesolo: Learning to segment objects without annotations. In *CVPR*, 2022. 1, 2, 6

[14] Yangtao Wang, Xi Shen, Shell Xu Hu, Yuan Yuan, James L. Crowley, and Dominique Vaufreydaz. Self-supervised transformers for unsupervised object discovery using normalized cut. In *CVPR*, 2022. 2, 6

[15] Jiaxing Zhao, Jiang-Jiang Liu, Deng-Ping Fan, Yang Cao, Jufeng Yang, and Ming-Ming Cheng. Egnet: Edge guidance network for salient object detection. In *ICCV*, 2019. 1

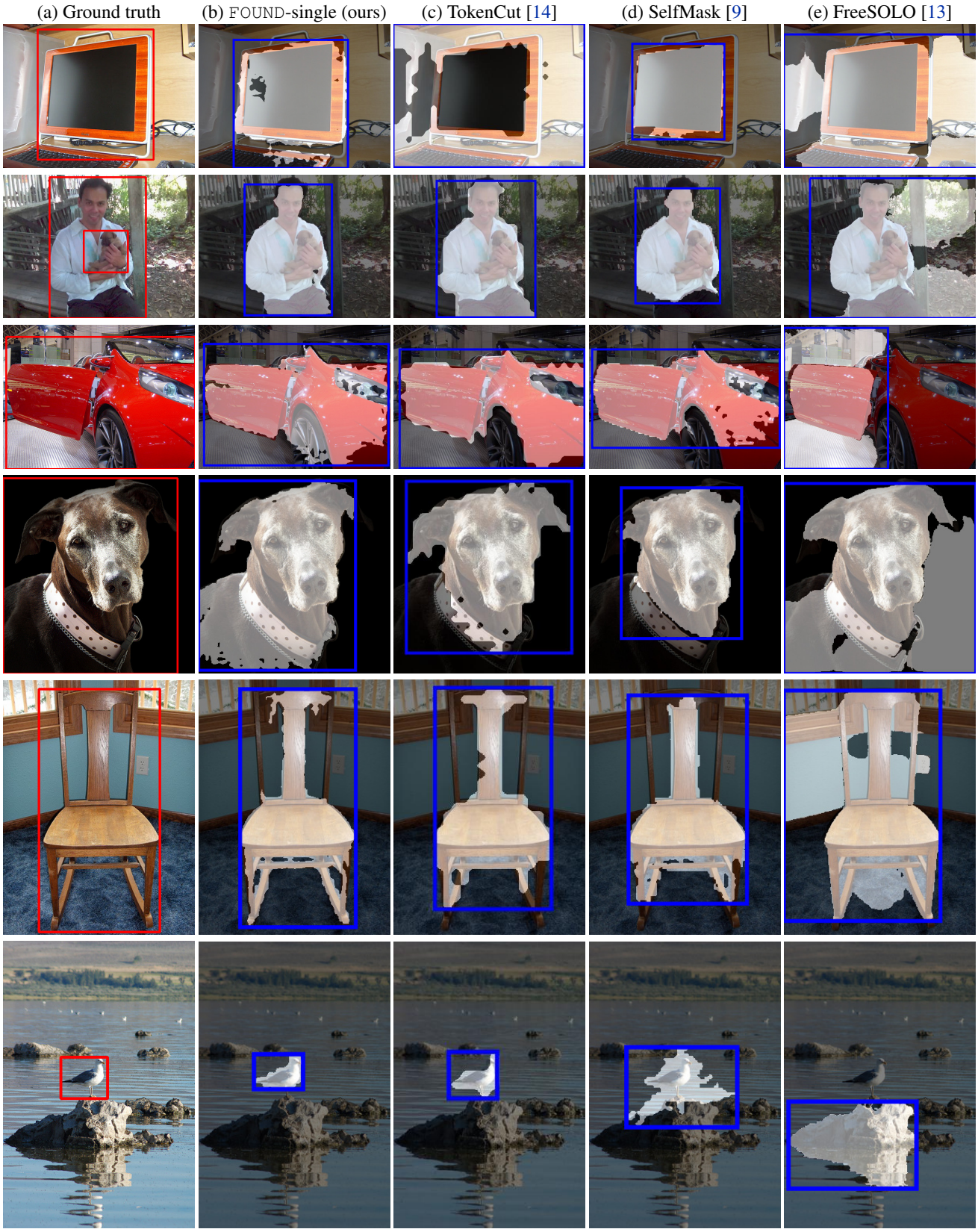


Figure 11. **Qualitative results** for the task of *unsupervised single object discovery* on PASCAL VOC12 dataset [5]. We show here masks and boxes extracted as defined in Sec. 4.1. In particular, FOUND is in the *single* setup (FOUND – single). No refinement step is applied on the masks.