CamouFinder: Finding Camouflaged Instances in Images

Trung-Nghia Le[1*], Vuong Nguyen[2, 3], Cong Le[2, 3], Tan-Cong Nguyen[2, 4], Minh-Triet Tran[2, 3, 5], Tam V. Nguyen[6]

[1] National Institute of Informatics, Japan
[2] Vietnam National University, Ho Chi Minh City, Vietnam
[3] University of Science, Ho Chi Minh City, Vietnam
[4] University of Social Sciences and Humanities, Ho Chi Minh City, Vietnam
[5] John von Neumann Institute, VNU-HCM, Vietnam
[6] University of Dayton, U.S.A

Abstract

In this paper, we investigate the interesting yet challenging problem of camouflaged instance segmentation. To this end, we first annotate the available CAMO dataset at the instance level. We also embed the data augmentation in order to increase the number of training samples. Then, we train different state-of-the-art instance segmentation on the CAMO-instance data. Last but not least, we develop an interactive user interface which demonstrates the performance of different state-of-the-art instance segmentation methods on the task of camouflaged instance segmentation. The users are able to compare the results of different methods on the given input images. Our work is expected to push the envelope of the camouflage analysis problem.

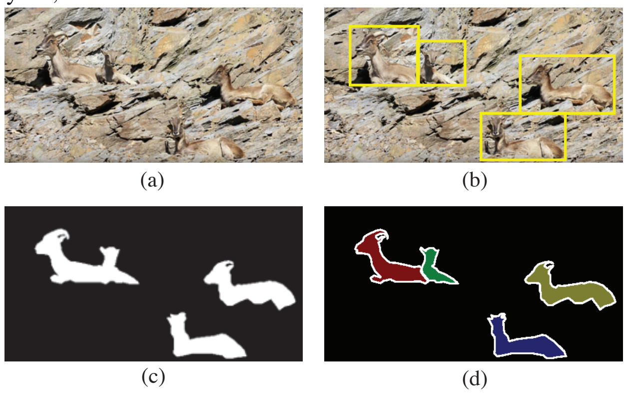

Figure 1: Illustrations of different tasks in camouflage analysis. From left to right: (a) input image, (b) camouflaged object detection, (c) camouflaged object segmentation, and (d) camouflaged instance segmentation.

Introduction

Camouflage is the combination of materials, coloration, or illumination for concealment which makes animals or objects difficult to be recognized (Singh, Dhawale, and Misra 2013). As discussed in (Le et al. 2019), there are naturally camouflaged objects such as leopard's spotted coat; and the artificially camouflaged objects such as the battledress of soldiers. In literature, finding salient objects and camouflaged object segmentation are really similar as they both use a binary mask as supervision. However, the goal of the former is to find regions whose contrast of foreground and background are more distinct, whereas the task of the latter is the opposite. The applications of salient object finding were fully discussed in (Nguyen, Zhao, and Yan 2018). Meanwhile, autonomously identifying camouflaged objects is beneficial in various fields of computer vision (search-and-rescue work; wild species discovery and preservation (Le et al. 2019)), COVID-19 infection identification from lung x-ray (Ucar and Korkmaz 2020)).

A few methods have been developed for camouflaged objects segmentation (Le et al. 2019; Yan et al. 2020). In addition, salient object detection methods (Le and Sugimoto 2015; Le and Sugimoto 2018, 2019; Le and Sugimoto 2017a,b; Nguyen 2015; Nguyen and Sepulveda 2015; Nguyen and Liu 2017; Nguyen, Nguyen, and Do 2019) also can be finetuned for the task of camouflaged objects segmentation. However, no existing work has been proposed to segment camouflaged objects at instance-level until the day.

In this work, we promote the new task of **camouflaged instance segmentation**. Camouflaged object is defined as all camouflaged pixels in an image without any further detail information such as the number of objects or semantic meaning (Yan et al. 2020). In contrast, *camouflaged instance consists of only meaningful pixels, which cover an object*. Camouflaged instance segmentation is more challenging than conventional camouflaged object segmentation in the sense that it *not only maps each pixels into labels but also sets instance identity for pixels*. Figure 1 illustrates different tasks in camouflage analysis. To our best knowledge, this is the first effort to showcase the camouflaged instance segmentation. Our work will be public at our project page[1].

We develop an interactive user interface to demonstrate and visualize the performance of state-of-the-art methods on the task of camouflaged instance segmentation. The users are able to examine the results of different methods on the given input images. Our demonstration is expected to be

*E-mail address: *ltnghia@nii.ac.jp*

[1]https://sites.google.com/view/ltnghia/research/camo

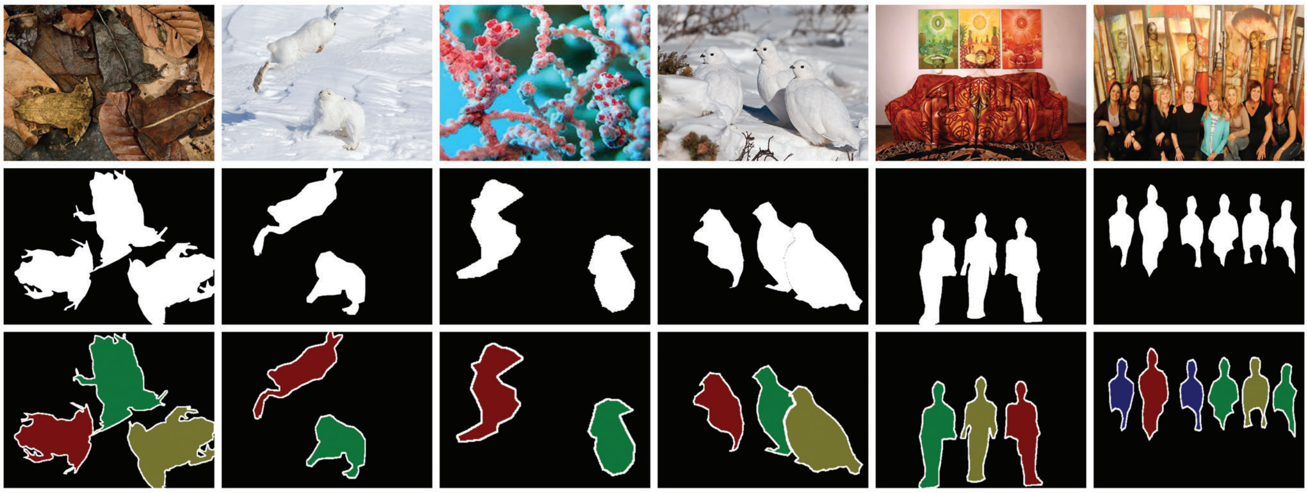

Figure 2: Exemplary samples of our data collection. From top to bottom: the original images, object ground-truth maps, instance ground-truth maps.

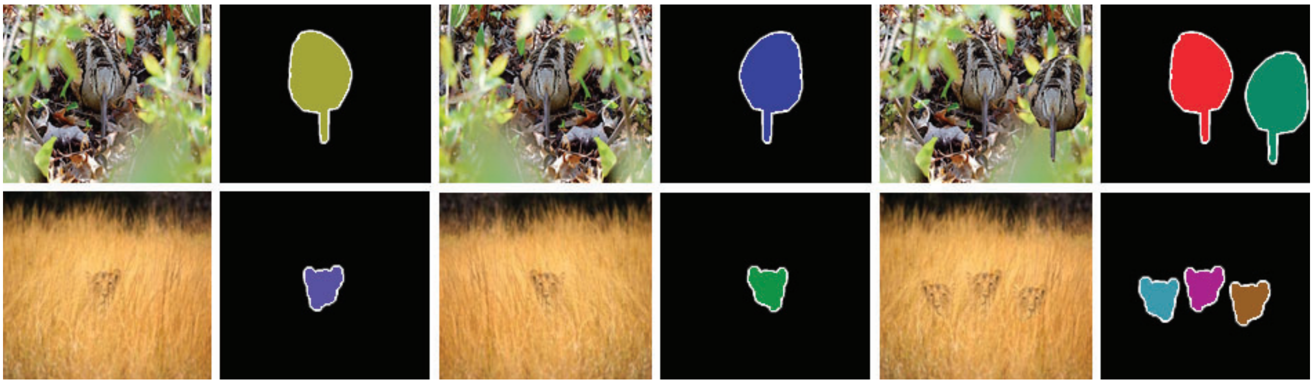

Figure 3: Exemplary samples of our data augmentation. From left to right: the original images with the ground-truth maps, the flipped images, the cloning instance images.

helpful to attract attention from the community to look into the research of camouflaged instance segmentation.

Proposed Framework

Data Collection

Since non-camouflaged object segmentation attracts most attention compared to camouflaged object segmentation, there are only a few relevant datasets, and most of them have the problem of too few samples. Therefore, we adopt CAMO dataset (Le et al. 2019) proposed and benchmarked for camouflaged object segmentation, for the training of instance segmentation framework. The dataset is divided into camouflage and non-camouflage categories, each containing 1,000 training and 250 test set, and a total of 2,500 manually annotated ground truths. Most of the dataset images are mammals, insects, birds, and aquatic animals, each with approximately similar proportions, and a small number of reptiles, human art, soldiers, and amphibians. The diversity of species in this dataset makes our framework adaptable, but it must be pointed out that it also has insufficient samples compared to mainstream datasets like COCO (Lin et al. 2014). We first compute the number of connected components on the binary ground truth maps of CAMO dataset (the training part). Then, we compute the bounding boxes for the components in the images. Figure 2 shows some samples of camouflaged instances on the CAMO dataset.

Data Augmentation

We later increase the number of training samples by using transformation methods such as cropping and flipping. It is essentially different from data augmentation for non-camouflaged objects, because to determinate whether an object is camouflaged is not only depends on its own features, but also its surroundings. Inspired by (Yan et al. 2020), we clone the object instances and place them onto different image regions with a small color difference in the background. We increase the number of training samples, alleviating the problem of insufficient data. Figure 3 shows samples of augmented data.

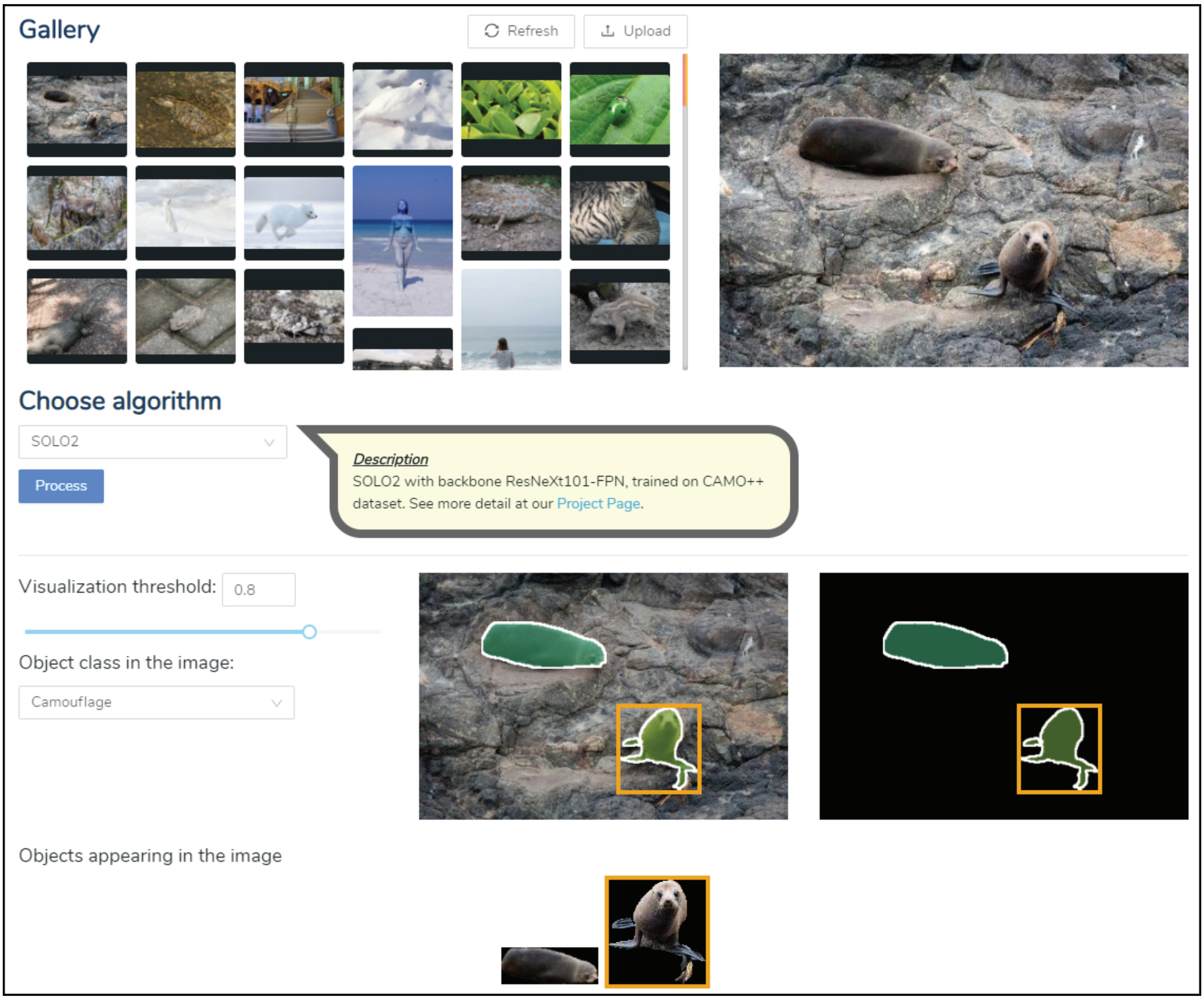

Figure 4: The interactive user interface of our camouflaged instance segmentation framework. The users are able to examine the results of different methods on the given input images.

Camouflage Instance Segmentation Methods

In order to segment the camouflaged instances, we trained and validated various instance segmentation methods (*i.e.*, Mask RCNN (He et al. 2017), Cascade RCNN (Cai and Vasconcelos 2018), Mask Scoring RCNN (Huang et al. 2019), RetinaMask (Fu, Shvets, and Berg 2019), YOLACT (Bolya et al. 2019b), YOLACT++ (Bolya et al. 2019a), CenterMask (Lee and Park 2020), BlendMask (Chen et al. 2020), SOLO (Wang et al. 2020a), SOLO2 (Wang et al. 2020b), and CondInst (Tian, Shen, and Chen 2020)) on the data mentioned above. These methods are categorized into single-stage and two-stage approaches. Two-stage methods follow detect-then-segment approach. These methods first perform object detection to extract bounding-boxes around each instance object, and then perform binary segmentation inside each bounding-box to separate the foreground (object) and the background. Meanwhile, single-stage methods are inspired by anchor-free object detection methods (such as CenterNet (Zhou, Wang, and Krähenbühl 2019) and FCOS (Tian et al. 2019)). Generally these single-stage methods are faster than two-stage methods.

Interactive User Interface

Then we develop an interactive user interface which demonstrates the performance of different state-of-the-art instance segmentation methods on the task of camouflaged instance segmentation. Our proposed systems consists of a front-end web-based interface and back-end web-services.

For front-end, we build a friendly web-based interface. We use Ant Design, Virtualized List, and CSS Position to layout the website and control interaction. ReactJS and Redux are utilized to manage state consistently in data flow.

For back-end, we deploy deep-learning models on Google Colab using deep-learning libraries (*e.g.*, PyTorch and TensorFlow). Python Flask and Ngrok support for data storage platform in Google Drive. We develop APIs to run our web-service and call them from the front-end interface. As seen in Figure 4, users are able to examine the results of different methods on the given input images. The users are also able to vary the score prediction thresholds for the instance filtering.

Conclusion and Future Work

In this paper, we introduce a framework for the challenging task of camouflaged instance segmentation. For the training data, we annotate the available CAMO dataset at the instance level and increase the number of training samples via data augmentation. Then, we train various state-of-the-art instance segmentation on the CAMO-instance data. In addition, we develop an interactive user interface for visualization of different methods on the input images. The users are able to examine the results of different methods on the given input images. We believe this work will attract and encourage more research in the camouflaged instance segmentation analysis problem.

Acknowledgements

This research is in part granted by National Science Foundation (NSF) under Grant No. 2025234, University of Dayton SEED Grant, JSPS KAKENHI Grant Number 20K23355, and Vingroup Innovation Foundation (VINIF) in project code VINIF.2019.DA19.

References

Bolya, D.; Zhou, C.; Xiao, F.; and Lee, Y. J. 2019a. YOLACT++: Better Real-time Instance Segmentation. In *arXiv preprint arXiv: 1912.06218*.

Bolya, D.; Zhou, C.; Xiao, F.; and Lee, Y. J. 2019b. YOLACT: Real-time Instance Segmentation. In *ICCV*.

Cai, Z.; and Vasconcelos, N. 2018. Cascade R-CNN: Delving into High Quality Object Detection. In *CVPR*.

Chen, H.; Sun, K.; Tian, Z.; Shen, C.; Huang, Y.; and Yan, Y. 2020. BlendMask: Top-Down Meets Bottom-Up for Instance Segmentation. In *CVPR*.

Fu, C.-Y.; Shvets, M.; and Berg, A. C. 2019. RetinaMask: Learning to predict masks improves state-of-the-art single-shot detection for free. In *arXiv preprint arXiv:1901.03353*.

He, K.; Gkioxari, G.; Dollár, P.; and Girshick, R. 2017. Mask r-cnn. In *ICCV*, 2980–2988.

Huang, Z.; Huang, L.; Gong, Y.; Huang, C.; and Wang, X. 2019. Mask Scoring R-CNN. In *CVPR*.

Le, T.; and Sugimoto, A. 2018. Video Salient Object Detection Using Spatiotemporal Deep Features. *IEEE Transactions on Image Processing* 27(10): 5002–5015.

Le, T.; and Sugimoto, A. 2019. Semantic Instance Meets Salient Object: Study on Video Semantic Salient Instance Segmentation. In *WACV*, 1779–1788.

Le, T.-N.; Nguyen, T. V.; Nie, Z.; Tran, M.-T.; and Sugimoto, A. 2019. Anabranch Network for Camouflaged Object Segmentation. *Journal of Computer Vision and Image Understanding* 184: 45–56.

Le, T.-N.; and Sugimoto, A. 2015. Contrast Based Hierarchical Spatial-Temporal Saliency for Video. In *PSIVT*, 734–748.

Le, T.-N.; and Sugimoto, A. 2017a. Deeply Supervised 3D Recurrent FCN for Salient Object Detection in Videos. In *BMVC*.

Le, T.-N.; and Sugimoto, A. 2017b. Spatiotemporal utilization of feep features for video saliency detection. In *ICME Workshop*.

Lee, Y.; and Park, J. 2020. CenterMask: Real-Time Anchor-Free Instance Segmentation. In *CVPR*.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft COCO: Common Objects in Context. In *ECCV*, 740–755.

Nguyen, T. V. 2015. Salient Object Detection via Objectness Proposals. In *AAAI*, 4286–4287.

Nguyen, T. V.; and Liu, L. 2017. Salient Object Detection with Semantic Priors. In *IJCAI*, 4499–4505.

Nguyen, T. V.; Nguyen, K.; and Do, T. 2019. Semantic Prior Analysis for Salient Object Detection. *IEEE Transactions on Image Processing* 28(6): 3130–3141.

Nguyen, T. V.; and Sepulveda, J. 2015. Salient Object Detection via Augmented Hypotheses. In *IJCAI*, 2176–2182.

Nguyen, T. V.; Zhao, Q.; and Yan, S. 2018. Attentive Systems: A Survey. *International Journal of Computer Vision* 126(1): 86–110.

Singh, S.; Dhawale, C.; and Misra, S. 2013. Survey of Object Detection Methods in Camouflaged Image. *IERI Procedia* 4: 351 – 357.

Tian, Z.; Shen, C.; and Chen, H. 2020. Conditional Convolutions for Instance Segmentation. In *ECCV*.

Tian, Z.; Shen, C.; Chen, H.; and He, T. 2019. FCOS: Fully Convolutional One-Stage Object Detection. In *ICCV*.

Ucar, F.; and Korkmaz, D. 2020. COVIDiagnosis-Net: Deep Bayes-SqueezeNet based diagnosis of the coronavirus disease 2019 (COVID-19) from X-ray images. In *Medical Hypotheses*.

Wang, X.; Kong, T.; Shen, C.; Jiang, Y.; and Li, L. 2020a. SOLO: Segmenting Objects by Locations. In *ECCV*.

Wang, X.; Zhang, R.; Kong, T.; Li, L.; and Shen, C. 2020b. SOLOv2: Dynamic, Faster and Stronger. *arXiv preprint arXiv:2003.10152* .

Yan, J.; Le, T.-N.; Nguyen, K.-D.; Tran, M.-T.; Do, T.-T.; and Nguyen, T. V. 2020. MirrorNet: Bio-Inspired Camouflaged Object Segmentation. *arXiv preprint arXiv:2007.12881* .

Zhou, X.; Wang, D.; and Krähenbühl, P. 2019. Objects as Points. In *arXiv preprint arXiv:1904.07850*.