Salient object detection with pretrained Deeplab and k-means: Application to UAV-captured building imagery

Victor Megir[1], Giorgos Sfikas[1], Athanasios Mekras[2], Christophoros Nikou[1], Dimosthenis Ioannidis[2], and Dimitrios Tzovaras[2]

[1] Dpt. of Computer Science and Engineering
University of Ioannina, Greece
{megir,sfikas,cnikou}@uoi.gr
[2] Information Technologies Institute
CERTH, 57001 Thessaloniki, Greece
{athmekras,djoannid,Dimitrios.Tzovaras}@iti.gr

Abstract. We present a simple technique that can convert a pretrained segmentation neural network to a salient object detector. We show that the pretrained network can be agnostic to the semantic class of the object of interest, and no further training is required. Experiments were run on UAV-captured aerial imagery of the "smart home" structure located in the premises of the CERTH research center. Further experiments were also run on natural scenes. Our tests validate the usefulness of the proposed technique.

Keywords: Deep features · Deeplab · Salient object detection · Structure detection

1 Introduction

Computer vision and pattern recognition today play a role of steadily increasing importance in various fields of the industry. Constructions, structure surveillance, agricultural productivity applications, autonomous driving are only a few of the fields that benefit from new pattern recognition technology advances. For example, as automotive industry is progressing to fully autonomous vehicle, computer vision is a tool for the development of many intelligent systems and important safety measures. Lane departure warning and lane keeping assistance are two technologies that can identify vehicles entering the current lane and trigger warnings for the driver and activate automatic braking systems to avoid a collision. Similar systems can be used to prevent accidents by detecting and monitoring pedestrians, traffic lights and signs [17]. Another area that takes advantage of those technologies is the agriculture and food industry. Computer vision inspection systems can accelerate the increase in agricultural productivity. Pattern recognition can be applied to detect crop diseases, thus allowing timely intervention. Vision algorithms are being used to classify the healthy food from defective, resulting in faster and more reliable harvest [19].

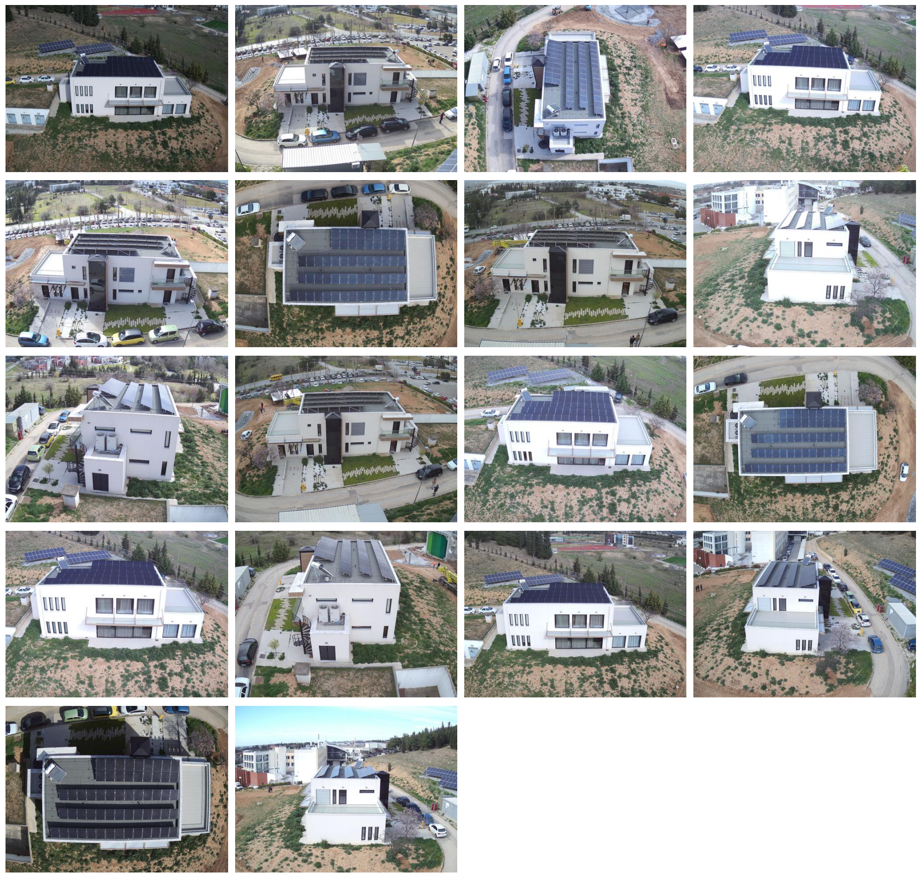

Fig. 1. Sample test images of the *CERTH* dataset.

Applications in construction benefit ever-increasingly by solutions provided by computer vision and pattern recognition. Classification systems can assess video data from construction sites in real-time, preventing defects and mistakes in this industry that can be crucial for both saving resources and human lives. Cameras can track the arrival of materials on a site and compare them to a project schedule. This process can improving efficiency, quality and safety [18]. Remote cameras and unmanned aerial vehicles (UAVs) can offer non-contact solutions to civil infrastructure condition assessment. Aerial survey and inspection of buildings for the purpose of gathering useful information for their assets, such as solar panels or air condition units, can provide a more comprehensive picture of their energy performance. Analyzing those data, can contribute to the development of a framework for energy loss evaluation and building energy efficiency improvement [10]. Accurate detection of structures of interest can help determine properties that are related to auxiliary sensors, like thermographic cameras [8].

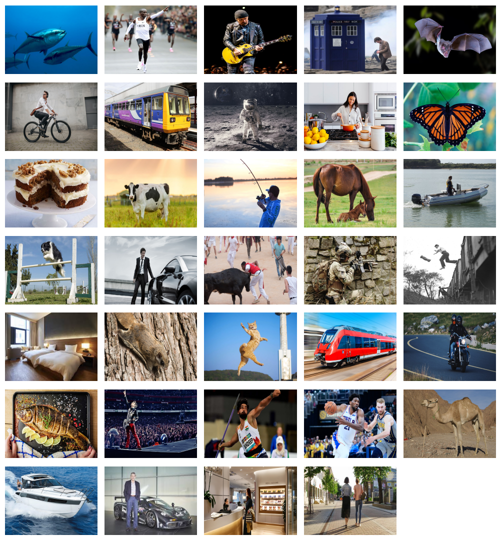

Fig. 2. Sample test images of the *Natural* dataset.

With this work we propose a method for salient object detection [2], tested to a setting where the object of interest is a structure. The base component of the method is a convolutional neural network that, pretrained for the task of semantic segmentation. We have used the popular Deeplab network [3], although in principle any off-the-shelf convolution segmentation network could be used in its place. Intermediate convolutional layer results, referred to in the literature as deep features [13] are used as powerful, high-level features. Deep features are in turn clustered, and the centermost cluster is selected as the salient object segment. This technique is shown to be succesfully applied where the object of interest is a building, as well as for a set of natural image scenes. Importantly, no extra training is required for the off-the-shelf base network. Furthermore, we show that our method works even when the base network is competely agnostic to the semantic class of the object of interest.

The paper is structured as follows. In section 2, we briefly review related work. Semantic segmentation using a state-of-the-art neural network is discussed in section 3, and in 4 we discuss the concept of deep features and how to use them for salient object segmentation. Experimental results, which compare the discussed segmentation methods, are presented in section 5, and close the paper with our conclusions in section 6.

2 Related work

The idea of using deep features as powerful image cues has been previously employed in a diverse range of computer vision tasks. The simplest variant of deep features is related to creating a holistic image descriptor by computing the activations of one of the network fully connected layers, typically situated on the head of convolutional networks [1]. Convolutional layers have also been used in this regard, creating pixel-level cues out of convolutional layer pointwise activations. This concept has been further extended to concatenating activations of a set of convolutional layers, after being resized and interpolated to a common height and width if required. This type of deep features has been referred to as hypercolumns [16, 12, 9]. These local cues can then be optionally encoded, with operations as simple as sum pooling leading to powerful image descriptors [16].

Variants of the idea of using deep features with clustering have been previously used as a data processing pipeline component to the end of producing Demand Response potential estimates [8, 15]. In [8], deep features are extracted from pretrained Deeplab, and used to create a binarization mask with an infrared input. This idea is demonstrated on a set of synthetic infrared images, created on the basis of infrared image statistics. In [15], the same concept is validated on aligned pairs of colour and real infrared images, captured using a dual camera mounted on a UAV. In both works, in contrast to the present work, two important issues are not covered, which are: a) what is the role and effect of the choice of the dataset used to create the pretrained model weights? b) if the pretrained model has seen the object class of interest during training, how does the proposed deeplab-based technique fare in comparison to using Deeplab output? With the current work, we discuss and attempt to answer these issues.

3 Semantic segmentation with Deeplab

Semantic segmentation as a task combines two long-standing problems in computer vision, that of image segmentation and image classification. In particular, it is defined as a simultaneous segmentation of the image and classification of each pixel to a semantic category. DeepLab is a state-of-the-art semantic segmentation model designed and open-sourced by Google. The original network components [3], employed and popularized "atrous" convolutional and deconvolutional layers, which act by inserting holes between nonzero filter taps. The final result was refined using a conditional random field model. In this work, we use one of the latest reiterations of the Deeplab, namely Deeplab v3+ [4]. This version of Deeplab uses atrous convolution and an encoder-decoder architecture, plus a refined decoder module to detect object boundaries. The output layer is topped by a pointwise softmax layer, giving class predictions for each pixel.

4 Segmentation using Deep features

Deep features are defined as features that are computed as the result of intermediate neural network layers, when fed with the test input on which we need to compute features $[12, 13, 11, 7]$. In other words, instead of using a feed-forward pass to compute the output of a neural network, which would of course require computation of all intermediate layers, we instead aim to compute the activation or pre-activation of a layer that stands between the input layer and between the network output. Powerful features can be obtained in this manner, after having ensured that the network has been trained on a specific task. In the context of image processing, convolutional neural networks (CNNs) are the norm when neural net-based methods are considered. CNNs comprise three main type of layers that we are interested in with respect to computing deep features: convolutional layers, deconvolutional layers, and dense or fully-connected layers. Dense layers feature a standard number of neurons, sharing no specific topology. On the contrary, convolutional and deconvolutional layers result in feature maps that consist of neurons arranged in an image-like 2D topology. Deep features are relevant to either dense or convolutional / deconvolutional layers, with dense layers leading to holistic descriptors and convolutional / deconvolutional layers leading to localized cues. This stems from the aforementioned topological properties of the neuron set associated with each type of layer. More formally, we can extract deep features from a dense layer in the form of a vector $x \in \Re^D$, where D is the number of layer neurons. On the other hand, deep features from a convolutional or deconvolutional layer come in the form $I \in \Re^{H \times W \times C}$, where H and W are the height and width of the associated feature map, and C is the number of channels resulting from the number of convolutional / deconvolutional layer depth. In other words, we obtain a vector $x \in \Re^C$ for each point $(i, j) \in [1, H] \times [1, W]$ in the feature map I. Note that these local cues can be combined to create a holistic feature using an encoding method [16].

What is of note is the type of deep features obtained as a function of the "distance" of the associated layer from the input and the output layers. Layers that are close to the input tend to produce deep features that are more generic and have little dependence on the training set with the which the network has been trained with. On the contrary, as our choice of layer moves toward the output, features become more and more training set-specific. It is frequently observed that the first convolutional layers after training completes typically correspond to low-level filters such as edge and blob detectors; subsequent layers encode more complex filters such as texture detectors; finally, layers that are situated close to the output act in effect as filters that produce cues with highly semantic content. The content of these features tends to be a close function of the training set characteristics, with filters that can be interpreted as e.g. object part detectors.

In figures 3 and 4 we show visualizations of deep features on sets of images (images of a structure, natural scenes). These visualizations were created by applying Principal Component Analysis (PCA) on points $x \in \Re^C$ obtained as deep features by a convolutional feature map. PCA is set to reduce dimension to 3, and each principal component is set to correspond to one of the RGB channels. Two different pretrainings are considered (PASCAL VOC20, ADE20k sets), with both using the same layer to obtain cues ($decoder/decoder\_conv1\_pointwise/Relu$:

0). (Pseudo-)color in these visualizations is perhaps difficult to interpret in absolute terms; however, what is highly interpretable is comparing pseudocolors of one image are to the other. Indeed, areas with similar pseudocolor are perceived by the network also to be similar in content. This rule also applies vice versa, with dissimilar color denoting objects/content that is also dissimilar.

In this work, we use Deeplab after assuming that it has already been trained on a specific set comprising K semantic classes. We aim to exploit the fact that, while the Deeplab output is a point-wise softmax that produces a K-size probability vector, deep features from the same model are more flexible; flexibility here is to be understood in the sense that deep features are not constrained to be tied to one or more of K semantic classes. Our strategy is first to cluster (convolutional) deep features into a set of n_s segments. We use k-means++ as our clustering method, after which we choose the cluster that is situated the most towards the center of the image as the salient object [15]. If the size of the feature map is different to that of the input, we resize the former to match the dimensions of the latter. The rest of the pixels are marked as background.

5 Experiments

Our experiments consist of salient object detection trials, executed on a number of different setups. The setup of the trials is chosen so as to check whether our main premise holds under a variability of experimental parameters, which is whether it is beneficial to use Deeplab "as-is", or use the proposed Deeplab deep feature-based technique.

We have compared standard Deeplab segmentation against the proposed technique on two datasets. The first dataset, which we refer to as *CERTH* in this paper, is composed of 18 images of the "smart home" building, found in the premises of the CERTH-ITI institute in Thessaloniki, Greece. All images depict the same structure, captured using a camera mounted on a DJI Unmanned Aerial Vehicle. The second dataset, which we refer to as *Natural*, is made up of 34 images that depict a variety of objects in various different scenes and layouts. We use the *CERTH* dataset as proof-of-concept that our premise is applicable in the context of structure segmentation, while the *Natural* dataset is used to test the method on a different, heterogeneous dataset. Samples of the two datasets can be seen in Fig. 1 and Fig. 2 respectively. All images have been manually annotated with corresponding pixel-level binary masks. In this manner, a ground-truth binarization to 'salient object' and 'background/rest' classes was available for evaluation of salient object segmentation.

Two variants of Deeplab were used, with the difference being in the dataset on which the network was trained at each time. In all cases the Xception architecture was used as the Deeplab backbone [5], employing depthwise separable convolutions. The two training sets used were the PASCAL-VOC20 dataset [6] the ADE20k datasets [21]. These datasets contain a total of 20 and 150 classes (plus background). What is important for our experiments here is that the PASCAL VOC20 dataset does not have a structure or building class, while ADE20k does. This means that a version of Deeplab (or any similar convolutional semantic segmentation network) that is trained on PASCAL VOC20 cannot recognize image areas that correspond to a building or structure correctly.

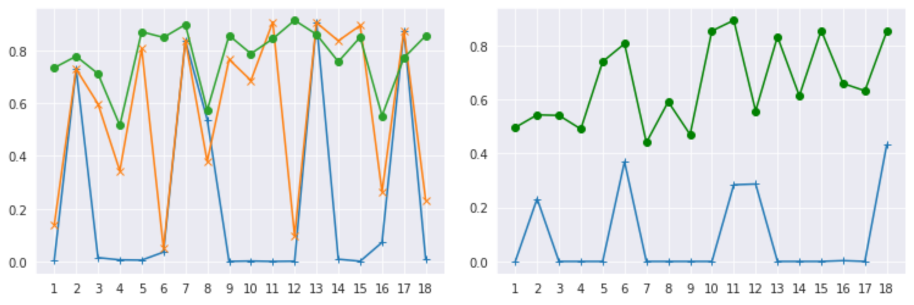

Fig. 6. Plots for results on *CERTH*, using ADE20k (left) and PASCAL VOC20 pretrained models (right). Horizontal axis corresponds to image id (1 to 18), vertical axis corresponds to IoU result (higher is better). Depicted curves are the proposed technique (green curve with circle markers), Deeplab output for ADE20k building class (orange curve with 'x' markers), Deeplab output for the "centermost" class (blue curve with '+' markers).

The proposed method was compared against simply using the output of Deeplab. In all our experiments, for both pretrained models considered we use the $decoder/decoder\_conv1\_pointwise/Relu:0$. We have chosen this layer as it is situated close to the output, therefore it is highly semantically "charged". The Intersection over Union (IoU) metric was used to evaluate each segmentation.

Visual and numerical results for trials over the *CERTH* dataset can be observed in Figure 5. Two plots comparing results over all *CERTH* dataset considered can be examined in Figure 6. One observation that is notable is that the proposed technique outperforms vanilla Deeplab, even when the dataset used to create the pretrained model weight is ADE20k, which *does* contain a structure class (this is clearer in the reported statistics of table 1). As PASCAL VOC20, which does not contain a structure class could not be directly compared in this manner, we have compared our techinique against the Deeplab output by considering the Deeplab cluster that is situated the most near the center (this is denoted in table 1 as "Deeplab centermost class"). Again, the proposed technique outperformed vanilla Deeplab. Commenting on the results between the pretrained model weights that correspond to the two considered datasets, ADE20k fares better than PASCAL VOC20. This is perhaps unsurprising, as ADE20k is considerably larger and contains more semantic classes than PASCAL VOC20.

We have run supplementary trials on the *Natural* dataset, of which visual and numerical results can be examined in Figure 7. Note that not all classes were detectable by the softmax output when used with PASCAL; (the available classes were 'aeroplane', 'bicycle', 'bird', 'boat', 'bottle', 'bus', 'car', 'cat', 'chair', 'cow', 'diningtable', 'dog', 'horse', 'motorbike', 'person', 'pottedplant', 'sheep', 'sofa', 'train', 'tv', plus background). Despite this fact however, our method could correctly detect object classes that would otherwise be uncategorizable by this version of the Deeplab model. Finally, statistics over results the *CERTH* and *Natural* datasets can be examined in table 1.

Table 1. Numerical results for the *CERTH* and *Natural* dataset trials. All figures are mean +- st.deviation for IoU scores (higher mean is better).

	CERTH		Natural	
	ADE20k	PASCAL	ADE20k	PASCAL
Deeplab centermost class	0.22 ± 0.4	0.09 ± 0.1	–	–
Deeplab building class	0.57 ± 0.3	N/A	–	–
Proposed method	0.78 ± 0.1	0.66 ± 0.2	–	0.70 ± 0.2

6 Conclusion

We have discussed a simple technique in order to perform salient object segmentation, that is based on a pretrained semantic segmentation network. What we believe that is important here, is that the pretrained network can be agnostic to the object class that we require to detect. This can be an advantage also when the object of interest is an outlier in its own class in terms of appearance, hence the network will normally have a hard time detecting it correctly. We have shown that the proposed technique can effectively be used to adapt Deeplab for salient object segmentation. The technique was tested succesfully on our dataset containing aerial photography of a structure of interest. As future work, we plan to test our method more extensively and to explore and adapt more advanced segmentation techniques (e.g. [14, 20]) to apply over deep features.

Acknowledgements

This research has been partially co-financed by the EU and Greek national funds through the Operational Program Competitiveness, Entrepreneurship and Innovation, under the call OPEN INNOVATION IN CULTURE, project *Bessarion* (T6YB*Π*-00214). It has also been partially co-financed by the EUs Horizon 2020 framework programme for research and innovation, project *eDREAM* (grant agreement No 774478).

References

1. Babenko, A., Slesarev, A., Chigorin, A., Lempitsky, V.: Neural codes for image retrieval. In: European conference on computer vision. pp. 584–599. Springer (2014)

2. Borji, A., Cheng, M.M., Hou, Q., Jiang, H., Li, J.: Salient object detection: A survey. Computational visual media pp. 1–34 (2019)
3. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. IEEE transactions on pattern analysis and machine intelligence **40**(4), 834–848 (2017)
4. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: Proceedings of the European conference on computer vision (ECCV). pp. 801–818 (2018)
5. Chollet, F.: Xception: Deep learning with depthwise separable convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1251–1258 (2017)
6. Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes (VOC) challenge. International journal of computer vision **88**(2), 303–338 (2010)
7. Goodfellow, I., Bengio, Y., Courville, A., Bengio, Y.: Deep learning, vol. 1. MIT press Cambridge (2016)
8. G.Sfikas, A.Noula, J.Patacas, D.Ioannidis, D.Tzovaras: Building thermal output determination using visible spectrum and infrared inputs. In: International Conference on Energy and Sustainable Futures (ICESF) (2019)
9. Hariharan, B., Arbeláez, P., Girshick, R., Malik, J.: Hypercolumns for object segmentation and fine-grained localization. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 447–456 (2015)
10. Kanellakis, C., Nikolakopoulos, G.: Survey on computer vision for uavs: Current developments and trends. Journal of Intelligent & Robotic Systems (01 2017). https://doi.org/10.1007/s10846-017-0483-z
11. Retsinas, G., Louloudis, G., Stamatopoulos, N., Sfikas, G., Gatos, B.: An alternative deep feature approach to line level keyword spotting. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 12658–12666 (2019)
12. Retsinas, G., Sfikas, G., Gatos, B.: Transferable deep features for keyword spotting. In: Multidisciplinary Digital Publishing Institute Proceedings. vol. 2, p. 89 (2018)
13. Retsinas, G., Sfikas, G., Louloudis, G., Stamatopoulos, N., Gatos, B.: Compact deep descriptors for keyword spotting. In: 2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR). pp. 315–320. IEEE (2018)
14. Sfikas, G., Nikou, C., Galatsanos, N., Heinrich, C.: Majorization-minimization mixture model determination in image segmentation. In: CVPR 2011. pp. 2169–2176. IEEE (2011)
15. Sfikas, G., Patacas, J., Psarros, C., Noula, A., Ioannidis, D., Tzovaras, D.: A deep neural network-based method for the detection and accurate thermography statistics estimation of aerially surveyed structures. In: 19th International Conference on Construction Applications of Virtual Reality (2019)
16. Sfikas, G., Retsinas, G., Gatos, B.: Zoning aggregated hypercolumns for keyword spotting. In: 2016 15th international conference on frontiers in handwriting recognition (ICFHR). pp. 283–288. IEEE (2016)
17. Stein, G.P., Rushinek, E., Hayun, G., Shashua, A.: A computer vision system on a chip: a case study from the automotive domain. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Workshops. pp. 130–130 (2005)

18. Tay, J., Shi, P., He, Y., Nath, T.: Application of computer vision in the construction industry. SSRN Electronic Journal (01 2019). https://doi.org/10.2139/ssrn.3487394
19. Tian, H., Wang, T., Liu, Y., Qiao, X., Li, Y.: Computer vision technology in agricultural automation a review. Information Processing in Agriculture **7** (09 2019). https://doi.org/10.1016/j.inpa.2019.09.006
20. Uziel, R., Ronen, M., Freifeld, O.: Bayesian adaptive superpixel segmentation. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 8470–8479 (2019)
21. Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., Torralba, A.: Scene parsing through ADE20k dataset. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 633–641 (2017)

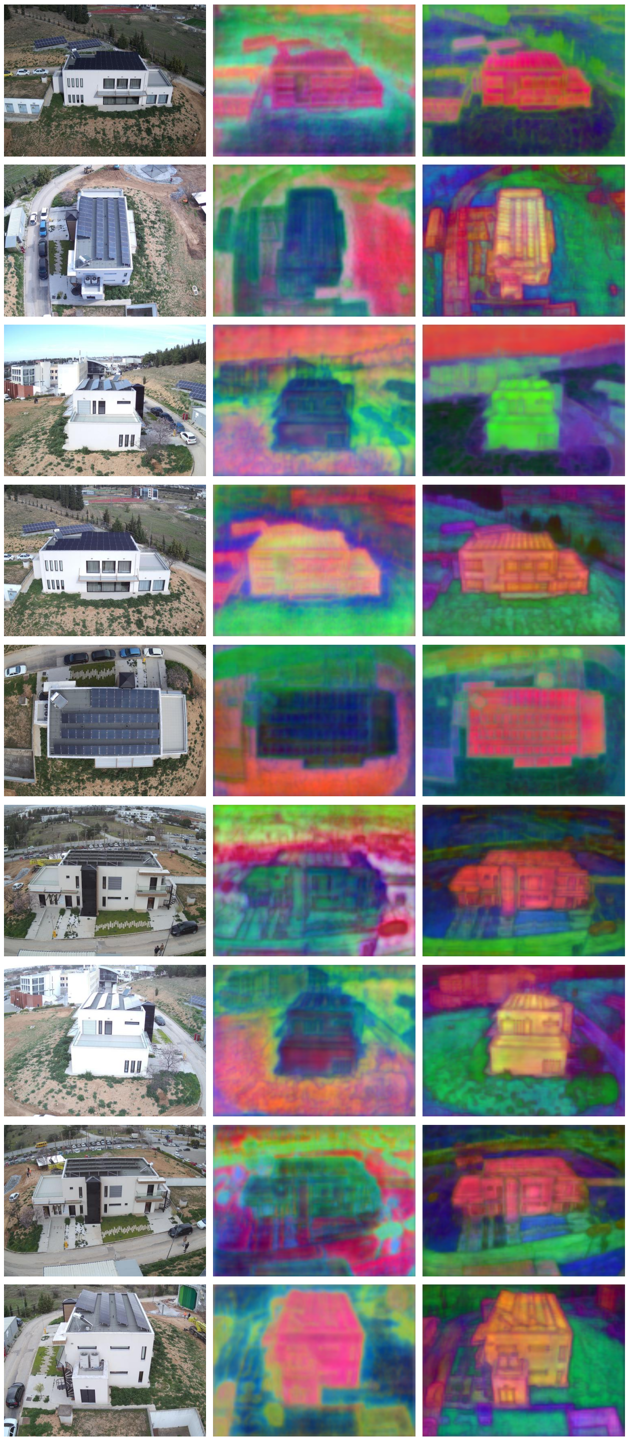

Fig. 3. Deep feature visualizations on images from the CERTH dataset. From left to right column: Original image, result with PASCAL VOC20 pretrained weights, result with ADE20k pretrained weights.

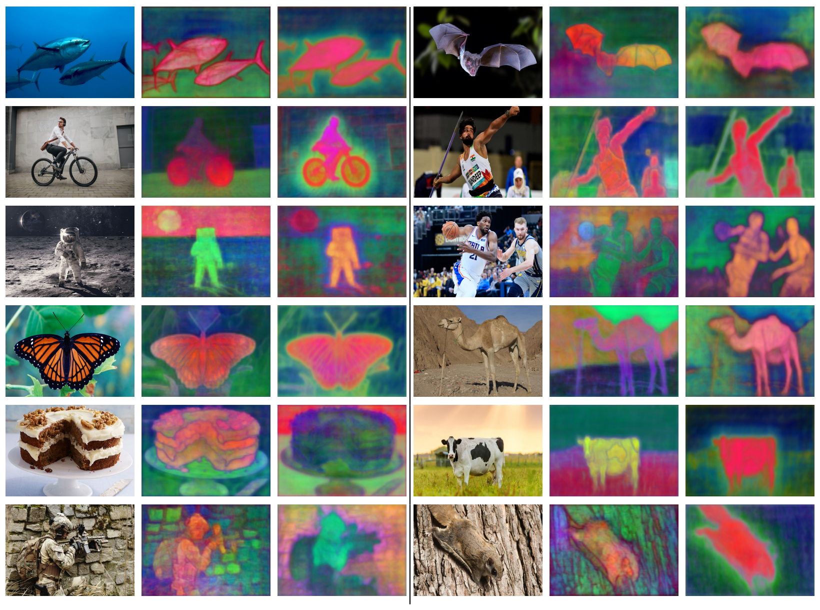

Fig. 4. Deep feature visualizations on images from the Natural dataset. From left to right column: Original image, result with PASCAL VOC20 pretrained weights, result with ADE20k pretrained weights.

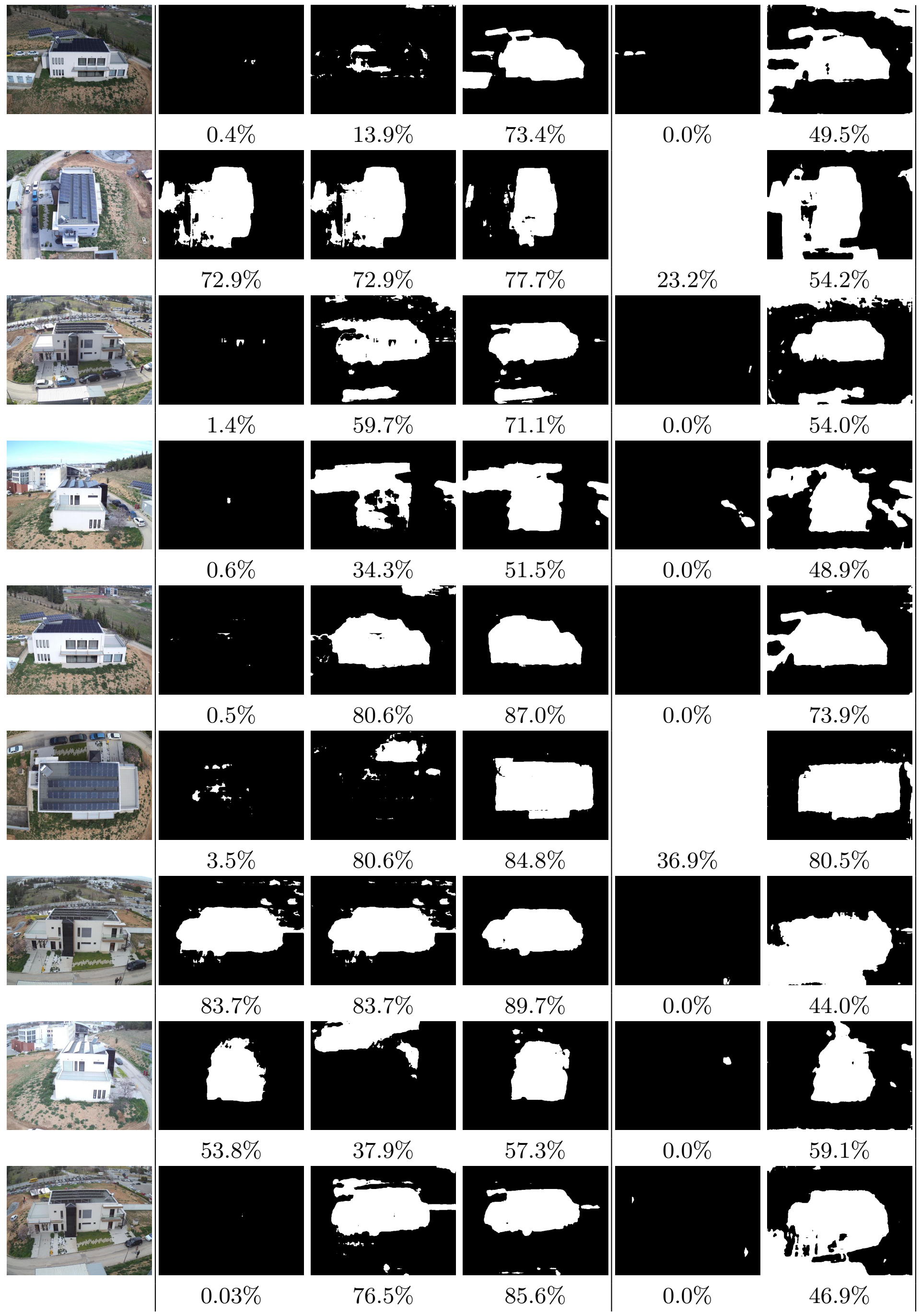

Fig. 5. Results on the *CERTH* dataset. From left to right, in each row the following images are shown: original image, ADE20k result by taking the centermost segment, ADE20k result by taking the estimated building class, ADE20k result by using the proposed method, PASCAL VOC20 result by taking the centermost segment, PASCAL VOC20 result by using the proposed method. Below each row IoU scores are also presented for the corresponding segmentation estimates.

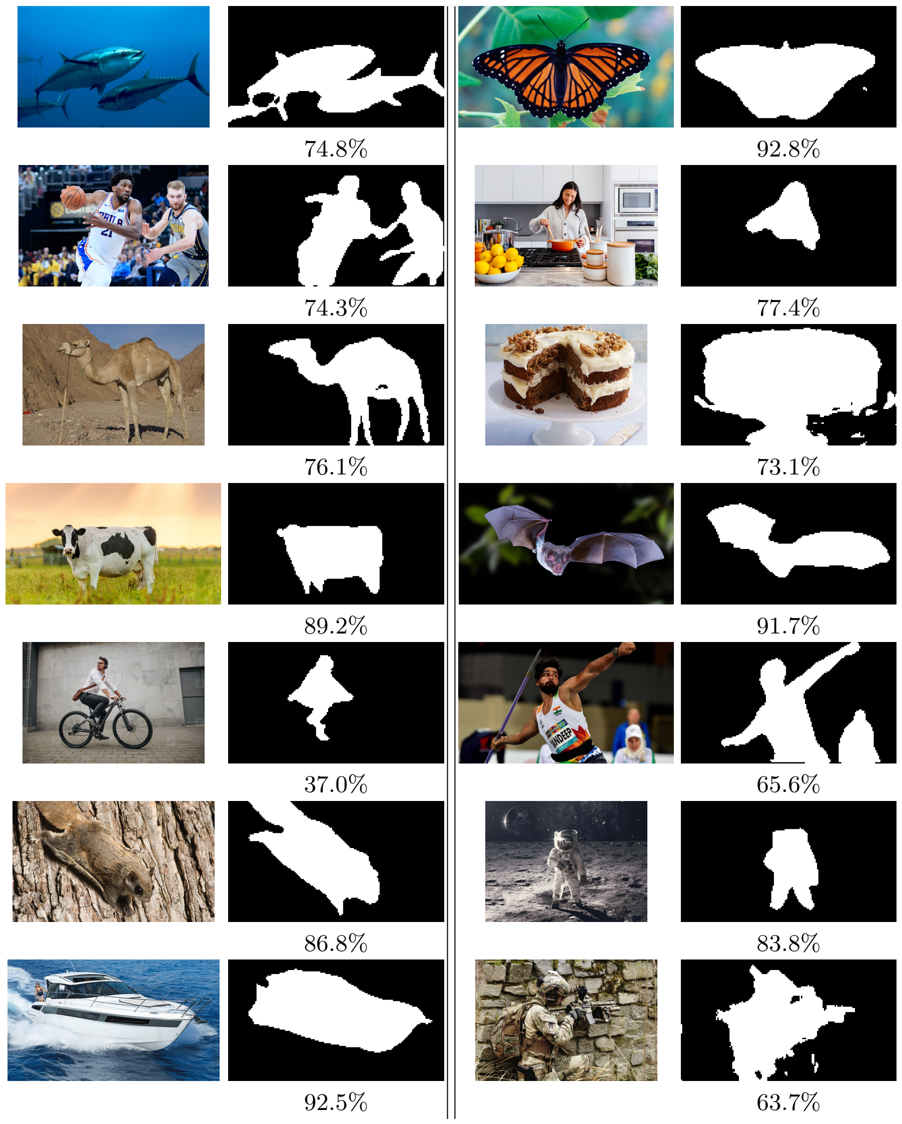

Fig. 7. Results on the *Natural* dataset. From left to right, in each row the following images are shown: original image, PASCAL VOC20 result, IoU score for the result of the current row. Note that objects that would otherwise be undetectable as Deeplab outputs (i.e. softmax classes) are correctly detected (see text for details).