Supplementary Material of Fisher Kernel for Deep Neural Activations

Anonymous CVPR submission

Paper ID 12

1. Run-Time for Extracting Multi-scale Dense Activations

We measure a run-time for extracting multi-scale dense activations. We then compare the run-time of a naive way and that of the proposed way. The details of the each method are as follows.

Naive Extraction

The naive extraction is composed of two steps, preprocessing and feedforward. The preprocessing step includes cropping multi-scale dense patches and resizing each of them into the CNN input size (227×227 in [1]). For cropping and resizing, we use `imcrop` and `imresize` functions in MATLAB. In the feedforward step, each patch is fed into the CNN and an activation vector is obtained from a target layer (FC7). For fast extraction, we make multiple batches and feed each batch to the CNN, where each batch contains 256 patches.

Proposed Extraction

The proposed extraction is illustrated in Figure 2 in the submitted paper. It also includes preprocessing and feedforward steps. In the preprocessing step, we resize an image into multi-scale images, starting from 227×227 size as the smallest image scale. For resizing, we use `imresize` function in MATLAB. In the feedforward step, we feed each scaled image to the CNN and get multi-scale dense activation vectors.

The preprocessing step is run on a CPU of 2.6GHz Intel Xeon. The extraction part is implemented with MatConvNet toolbox [2] and run on a GPU of GTX TITAN Black. The Caffe reference model [1] is used for the extraction part and the activations are obtained from FC7 layer of the model. We measure the average run-time with 100 random images from PASCAL VOC 2007. The result is shown in Table 1. With the proposed method, *a substantial amount of rich activations are obtained with reasonable time*.

Combination of image scales	1∼4	1∼5	1∼6	1∼7
Number of activations	270	754	1,910	4,410
Performance (%) on MIT Indoor 67	68.58	71.34	74.33	**75.67**
Performance (%) on PASCAL VOC 2007	78.06	79.16	**79.65**	79.54
Performance (%) on Oxford 102 Flowers	87.88	89.79	90.86	**91.28**
Preprocessing time (sec) for the naive extraction	1.008	2.942	6.478	16.28
Preprocessing time (sec) for the proposed extraction	0.0457	0.0741	0.1016	0.1872
Feedforward time (sec) for the naive extraction	0.6934	1.998	4.932	11.36
Feedforward time (sec) for the proposed extraction	0.0312	0.0524	0.1405	0.2763
Total time (sec) for the naive extraction	1.702	4.941	11.41	27.64
Total time (sec) for the proposed extraction	**0.0769**	**0.1265**	**0.2420**	**0.4635**
Speedup (times)	×22.13	×39.06	×47.15	×59.63

Table 1. Average run-time for extracting multi-scale dense activations per image.

2. Performances with Different Target Layers

We show performances of classification with different target layers. We use Caffe reference model [1] and seven scales for image pyramid. The PCA dimension is 128 and the number of Gaussians is 256. For average pooling method, ten images are augmented from one image.

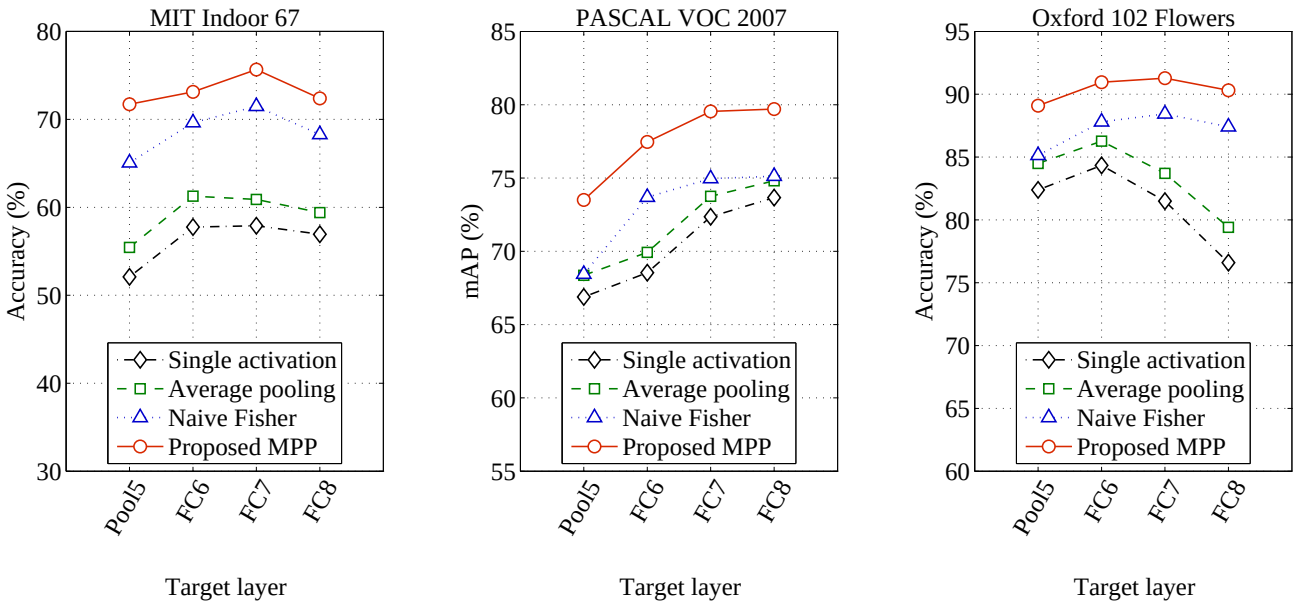


Figure 1. Performances with different target layers.

3. Performances with Different Fisher Kernel Parameters

Every experiment here uses activations from FC7 of the Caffe reference model [1] and seven scales for image pyramid. We decided to use PCA dimension of 128 and 256 Gaussians for efficient experiments in the submitted paper, because the two values (128, 256) show reasonably good performance in all benchmarks, as shown in Fig. 2 and Fig. 3.

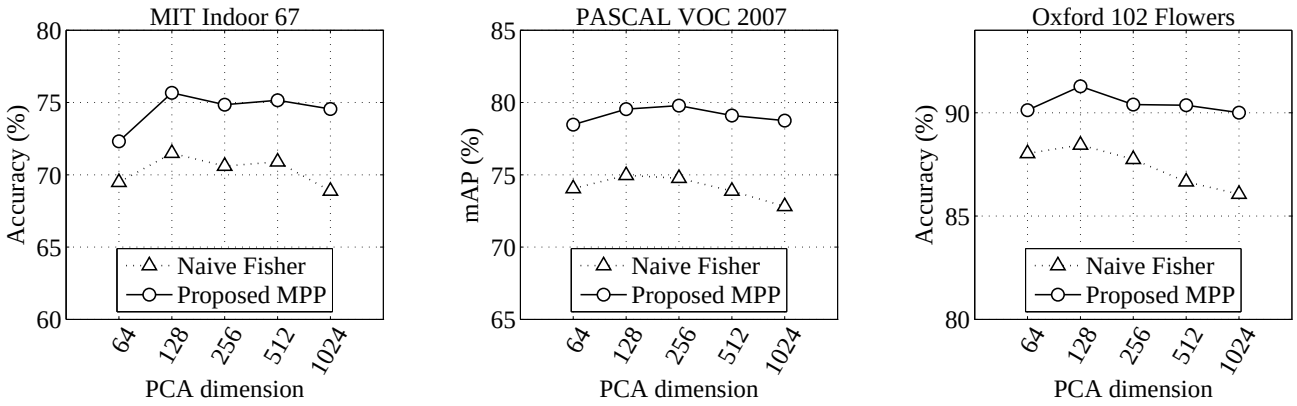


Figure 2. Performances with different PCA dimensions.

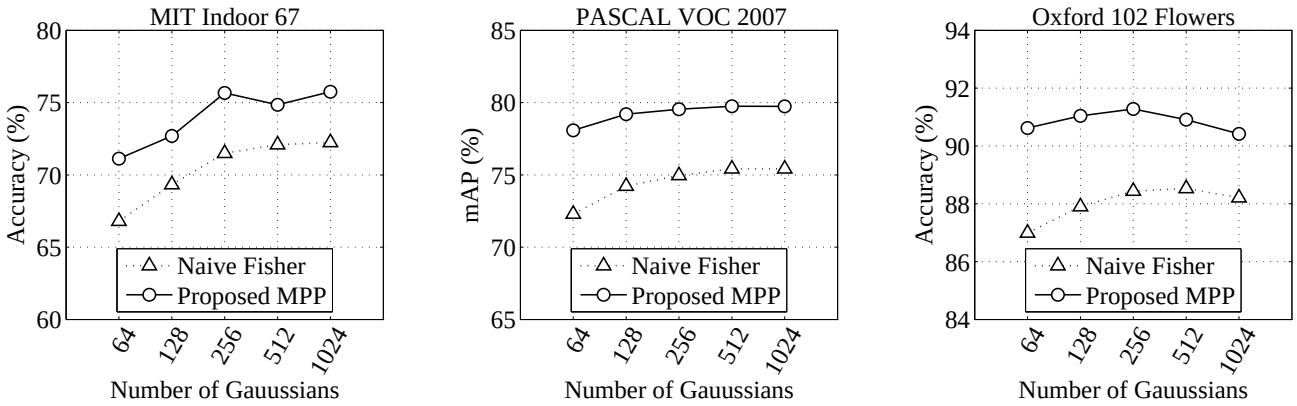


Figure 3. Performances with different number of Gaussians.

4. More Examples of Weakly-Supervised Object Confidence Maps

We show more examples of object confidence map from PASCAL VOC 2007. How we obtain the object confidence maps is described in Sec. 4.5 in the submitted paper. Note that *the following confidence maps are obtained by the supervision only with the class-level labels, without using object bounding box annotations*.

In Fig. 4 and Fig. 5, we show examples of a single image having different multiple classes. In Fig. 6 $\sim$ Fig. 25, we show good and bad examples of 20 different classes. Every image in the examples are test images, not training images. The examples start in the next page.

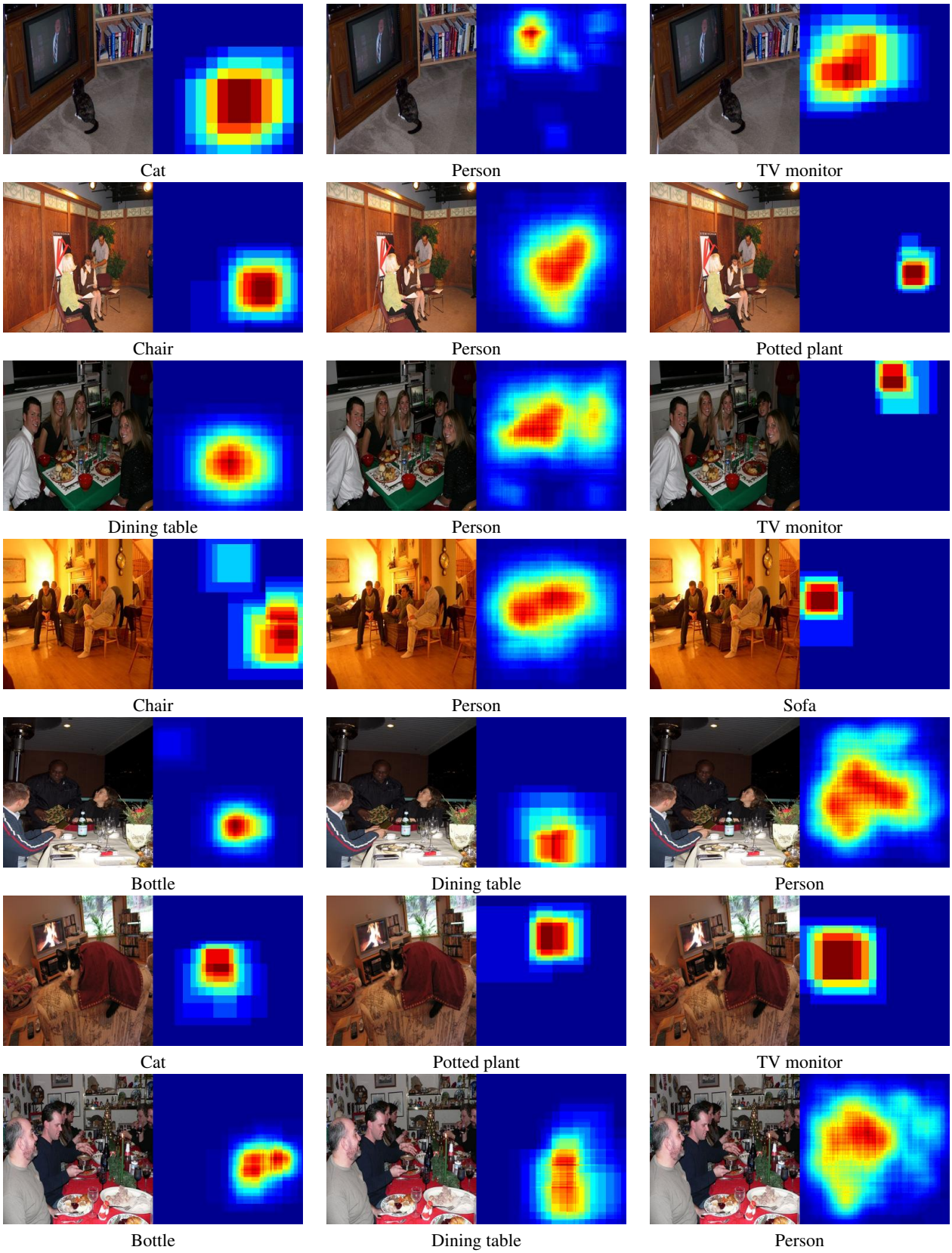


Figure 4. Examples of multi-class object confidence map of indoor scenes.

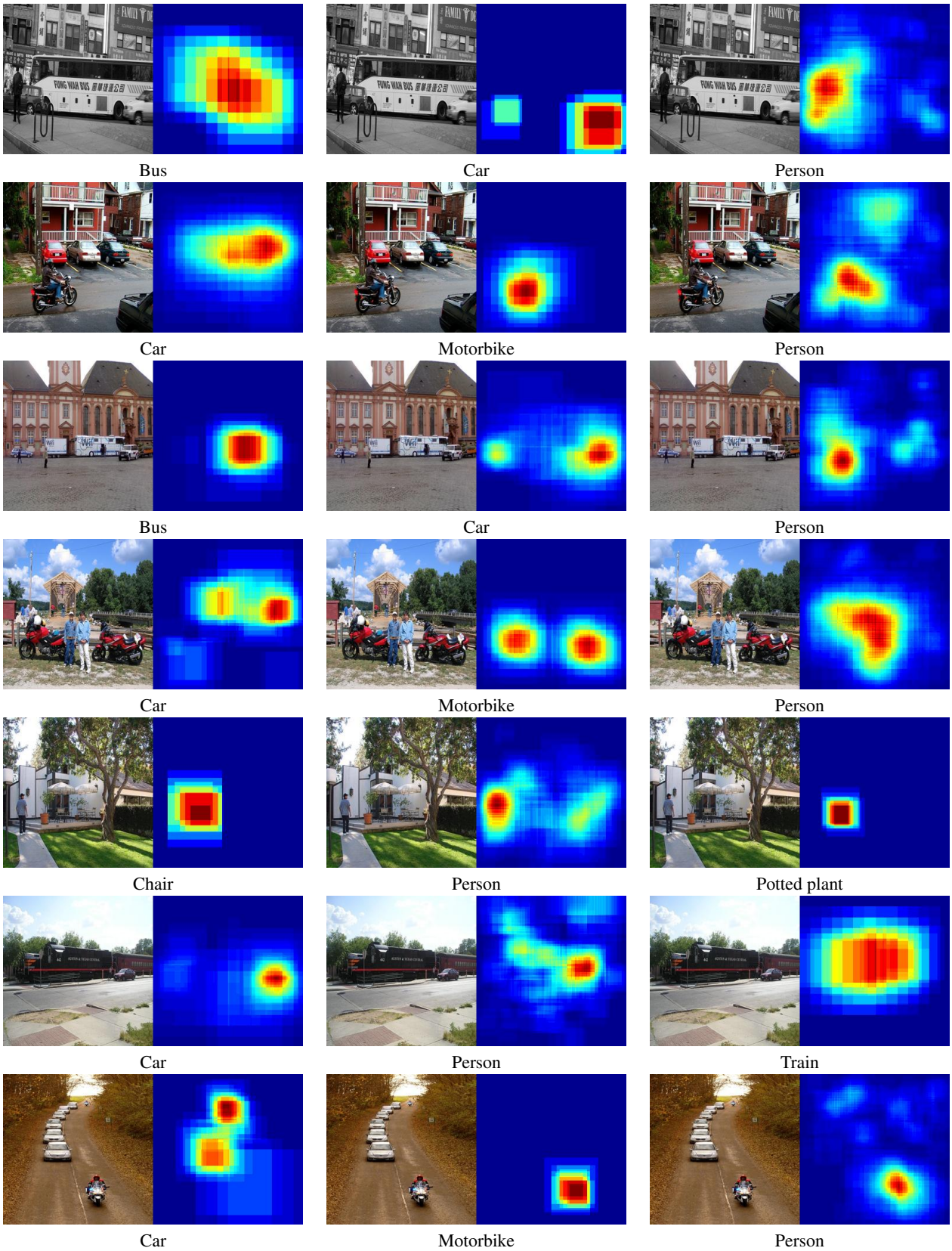
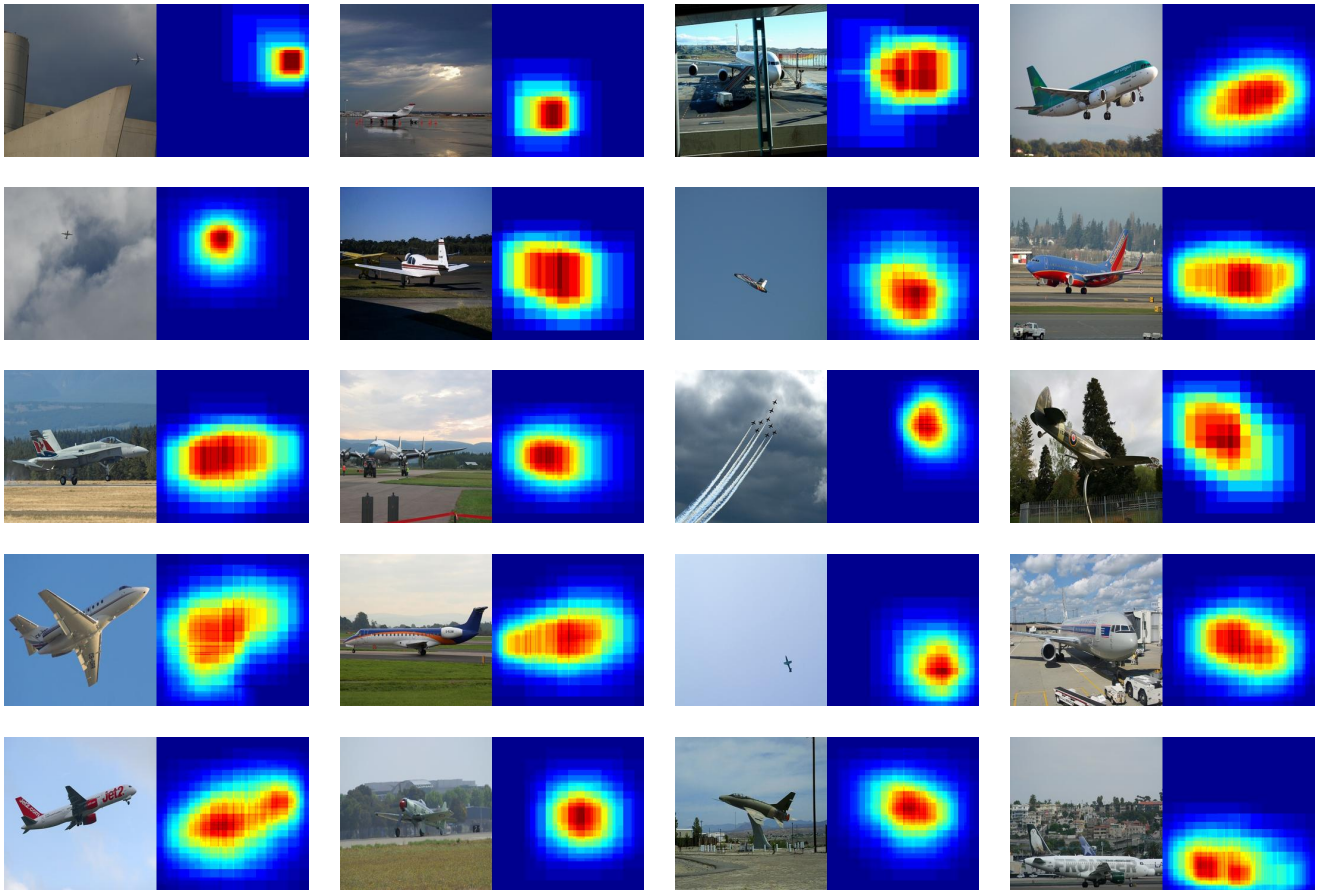
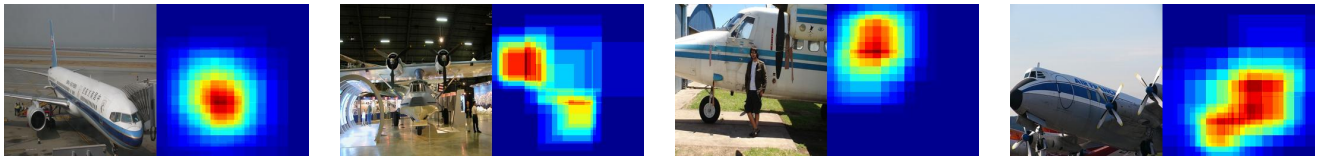


Figure 5. Examples of multi-class object confidence map of outdoor scenes.



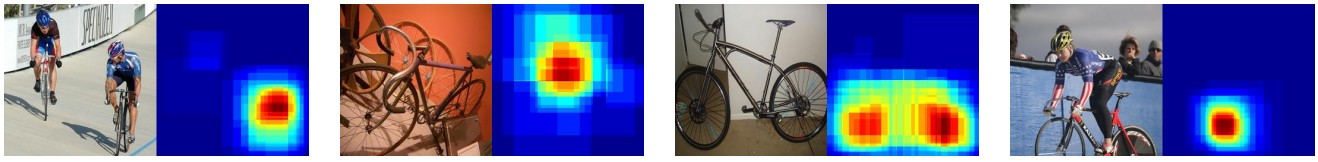
(a) True examples of the “aeroplane” class.



(b) False examples of the “aeroplane” class.

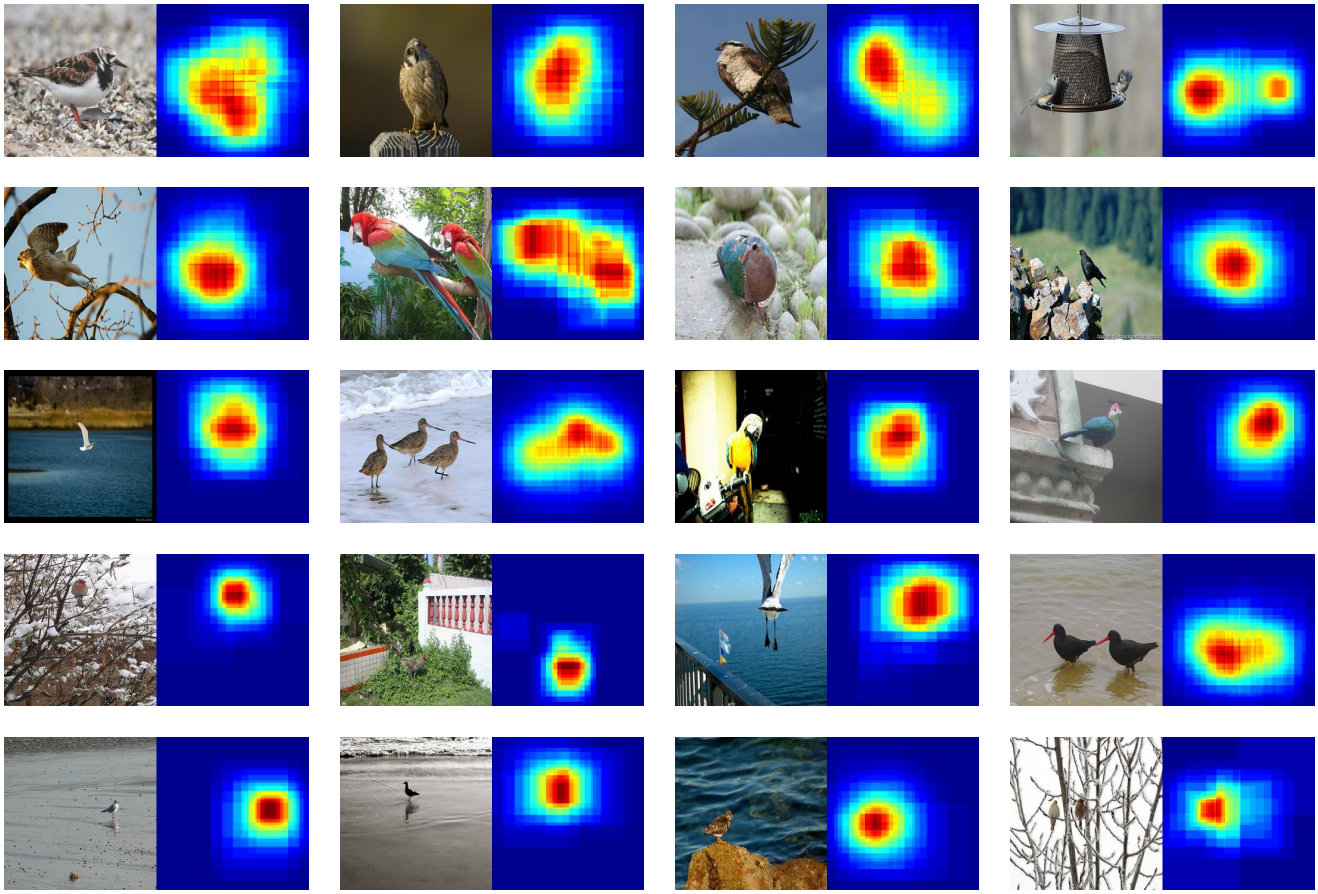
Figure 6. Examples of object confidence map of “aeroplane” class in the PASCAL VOC 2007 test image set.

(a) True examples of the “bicycle” class.

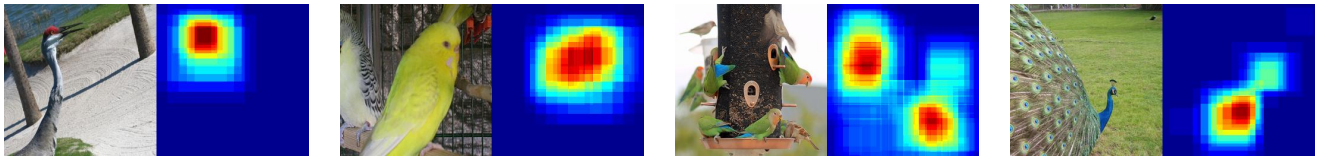


(b) False examples of the “bicycle” class.

Figure 7. Examples of object confidence map of “bicycle” class in the PASCAL VOC 2007 test image set.

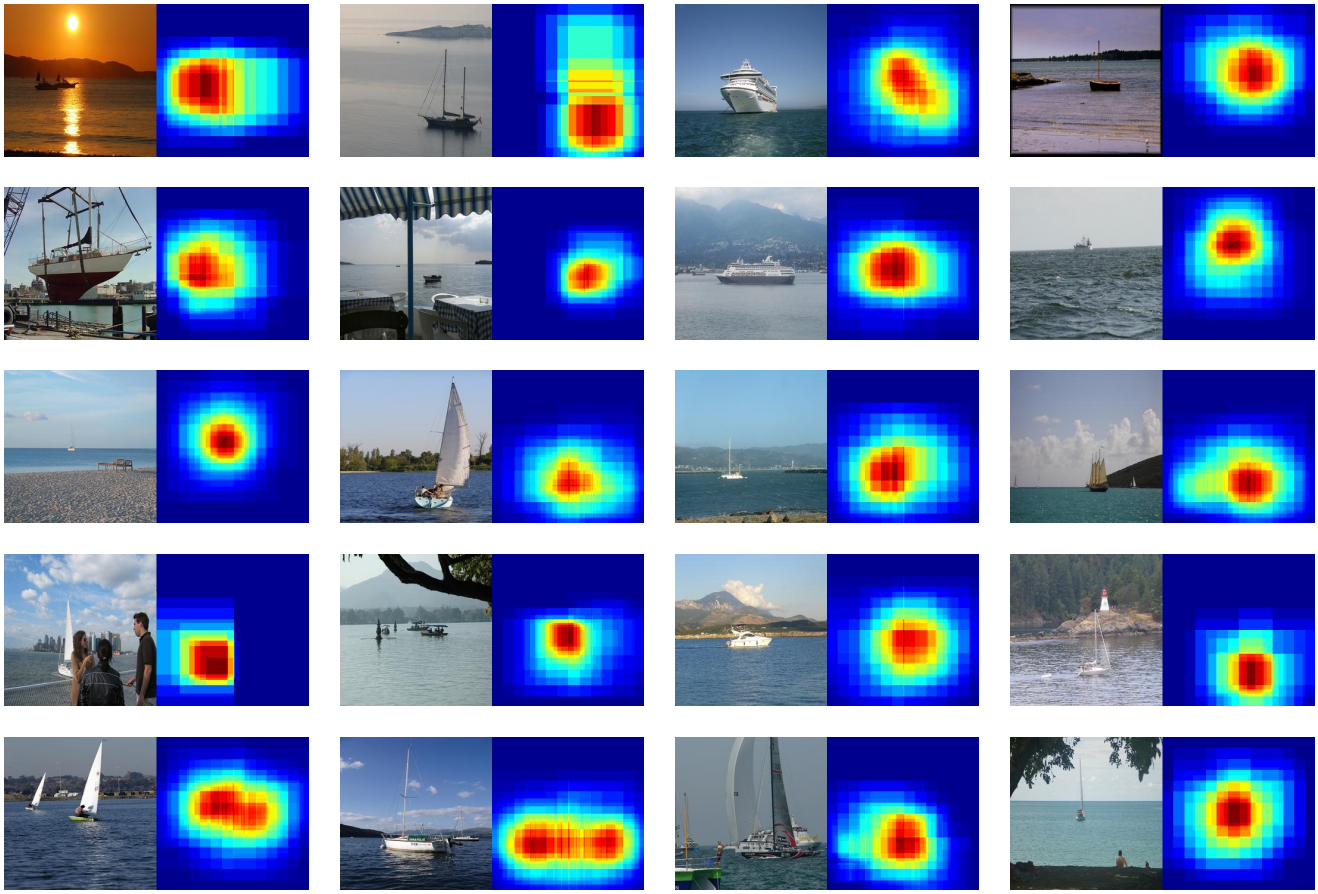


(a) True examples of the “bird” class.

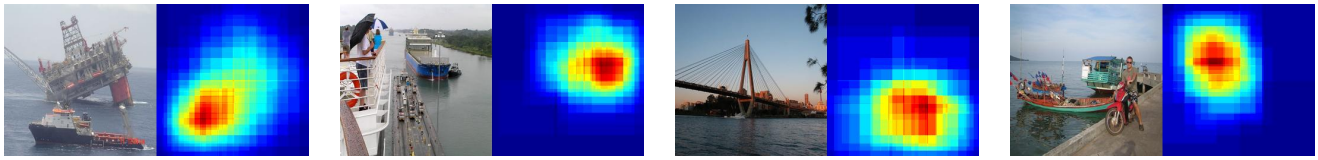


(b) False examples of the “bird” class.

Figure 8. Examples of object confidence map of “bird” class in the PASCAL VOC 2007 test image set.

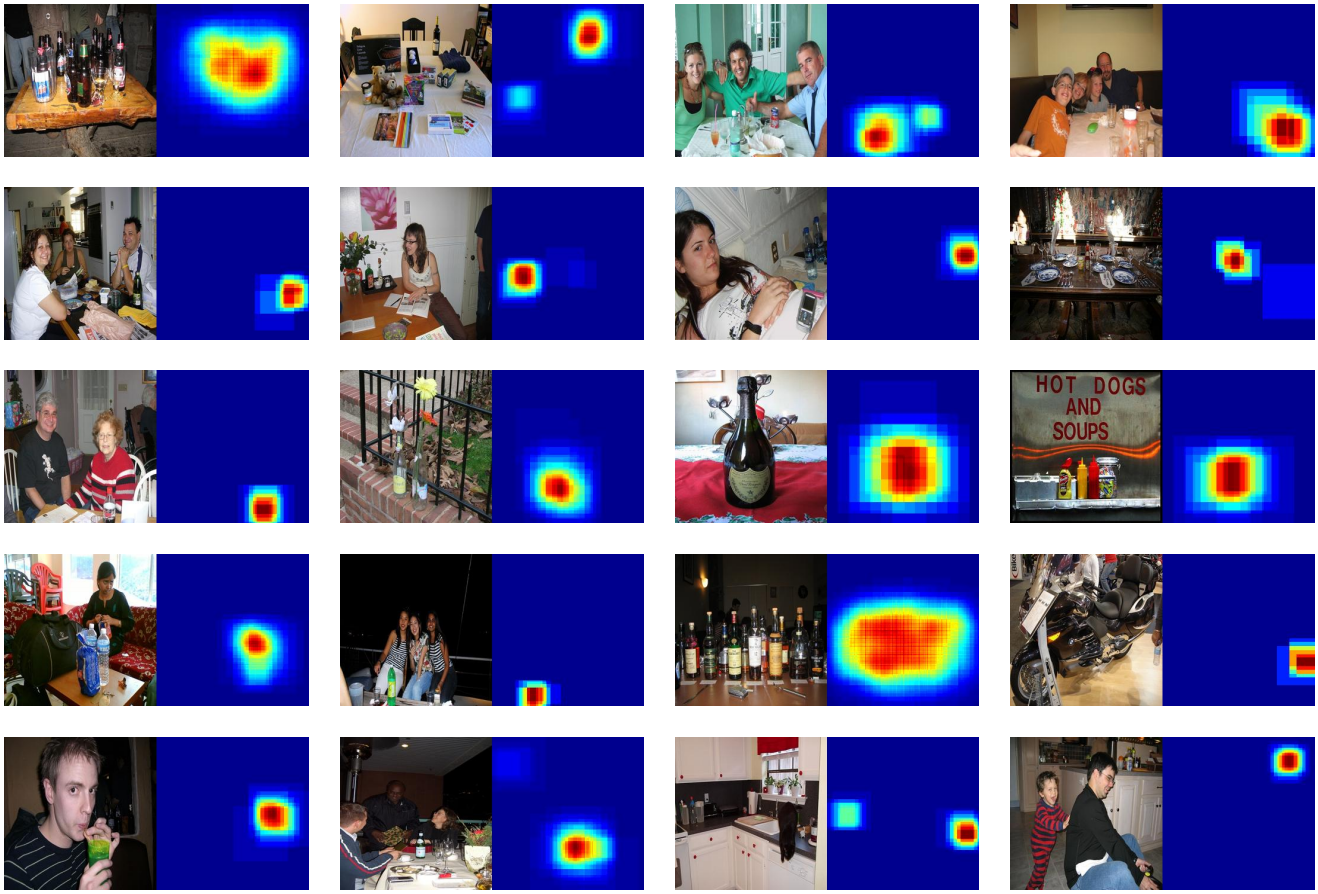


(a) True examples of the “boat” class.

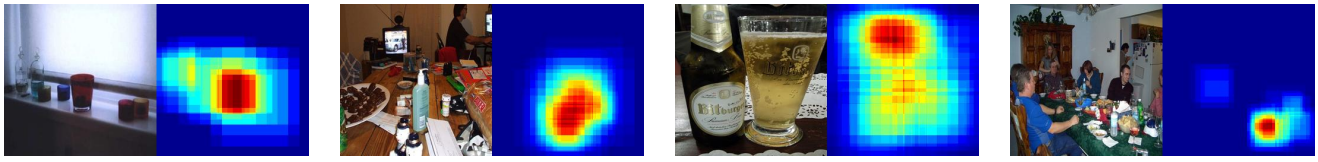


(b) False examples of the “boat” class.

Figure 9. Examples of object confidence map of “boat” class in the PASCAL VOC 2007 test image set.

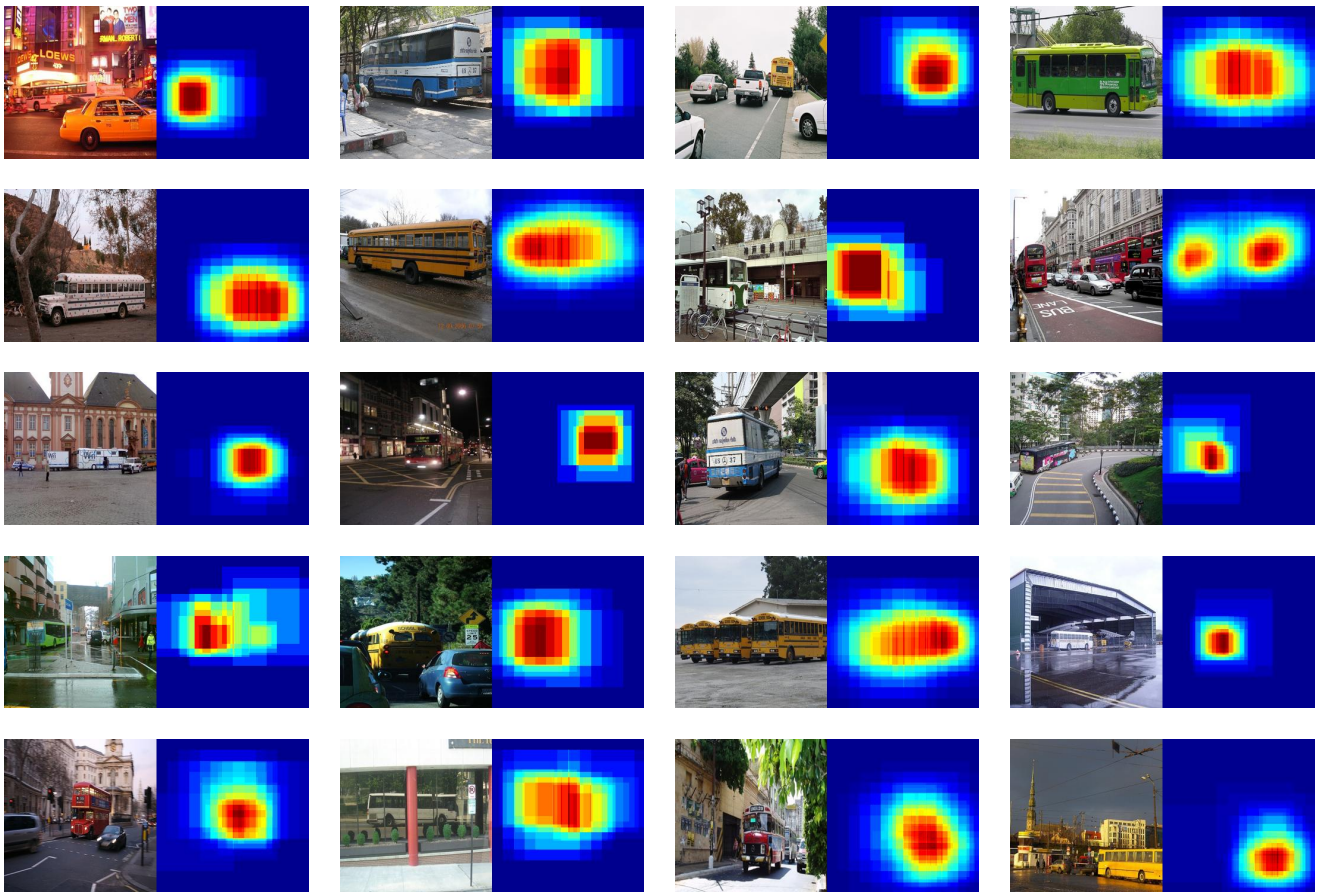


(a) True examples of the “bottle” class.

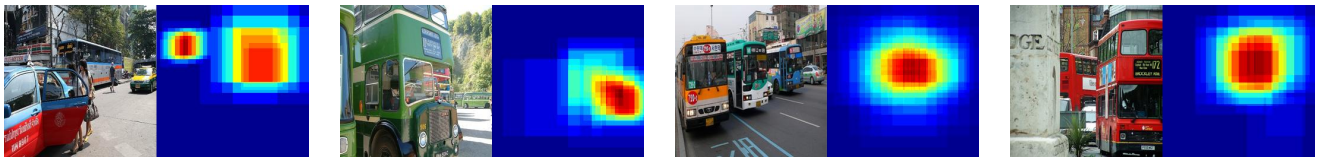


(b) False examples of the “bottle” class.

Figure 10. Examples of object confidence map of “bottle” class in the PASCAL VOC 2007 test image set.

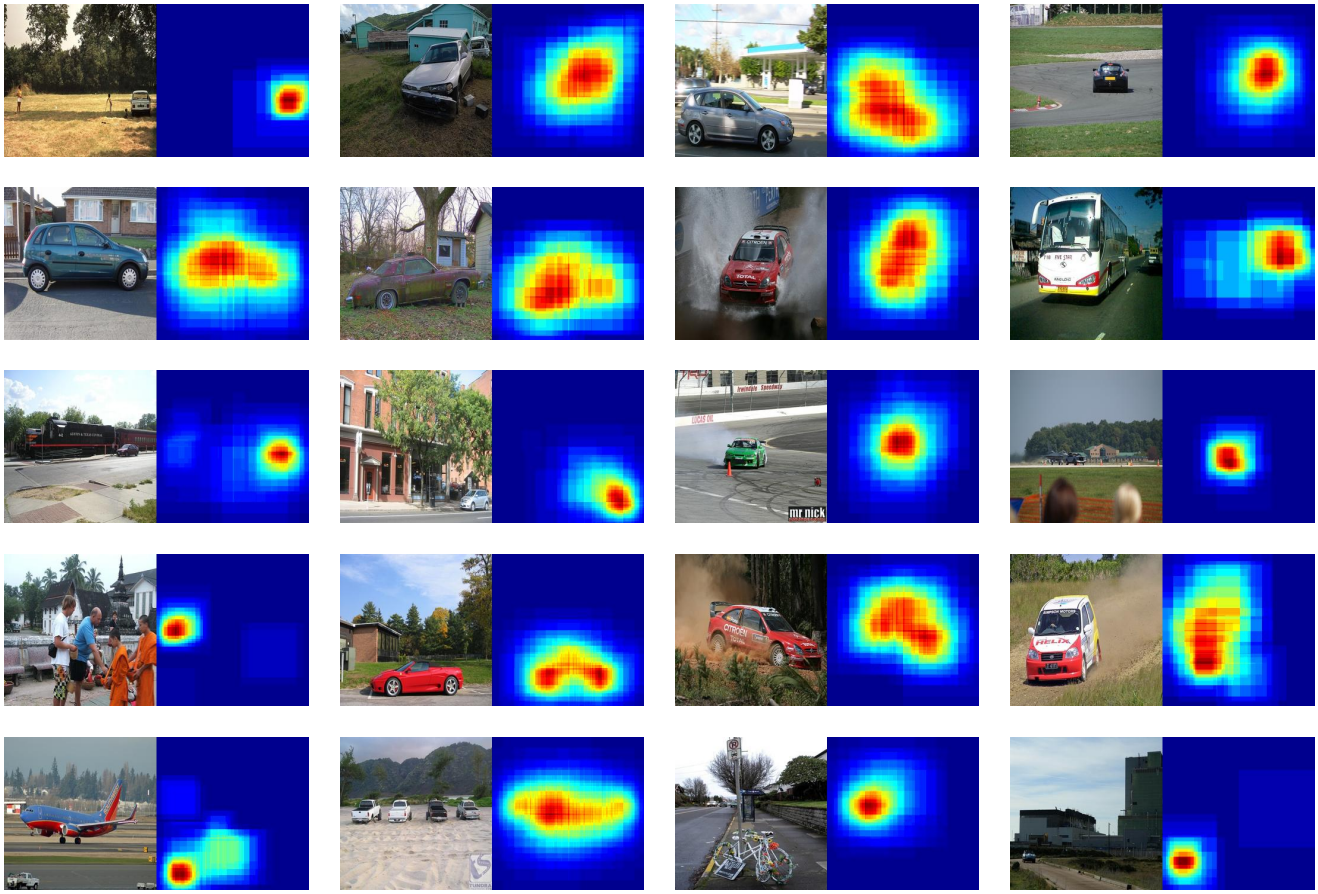


(a) True examples of the “bus” class.

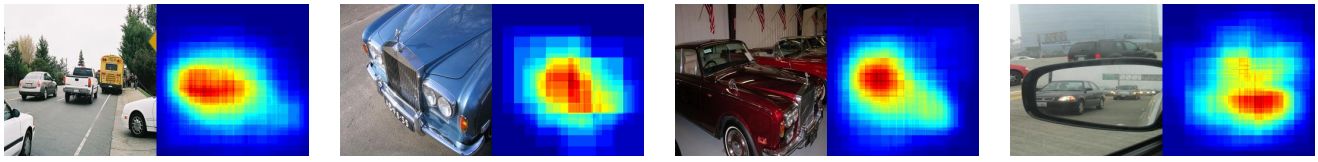


(b) False examples of the “bus” class.

Figure 11. Examples of object confidence map of “bus” class in the PASCAL VOC 2007 test image set.

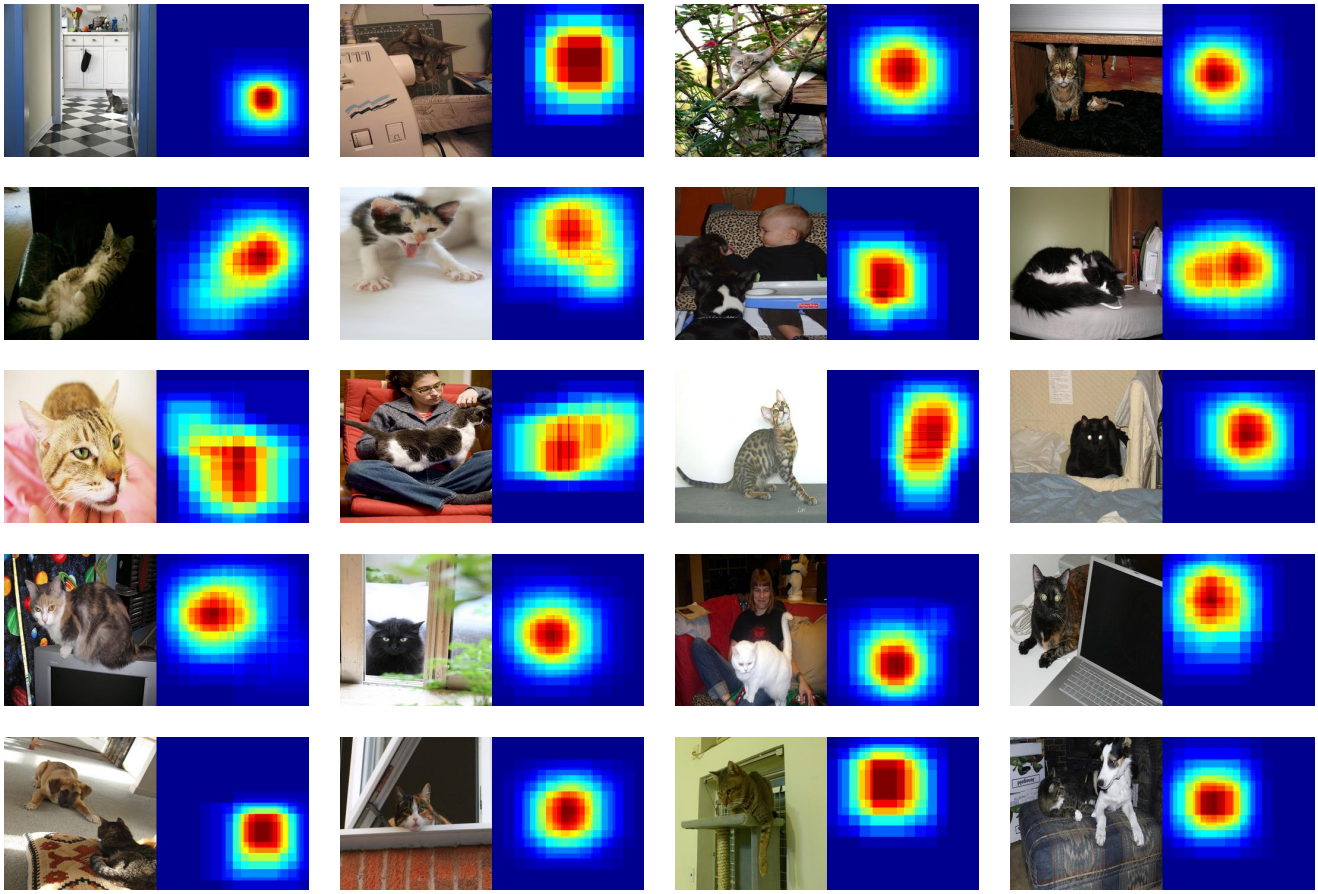


(a) True examples of the “car” class.

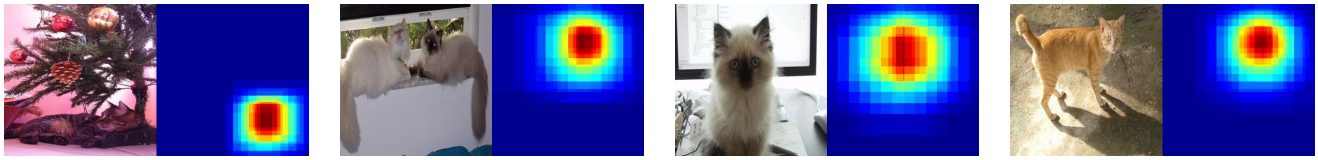


(b) False examples of the “car” class.

Figure 12. Examples of object confidence map of “car” class in the PASCAL VOC 2007 test image set.

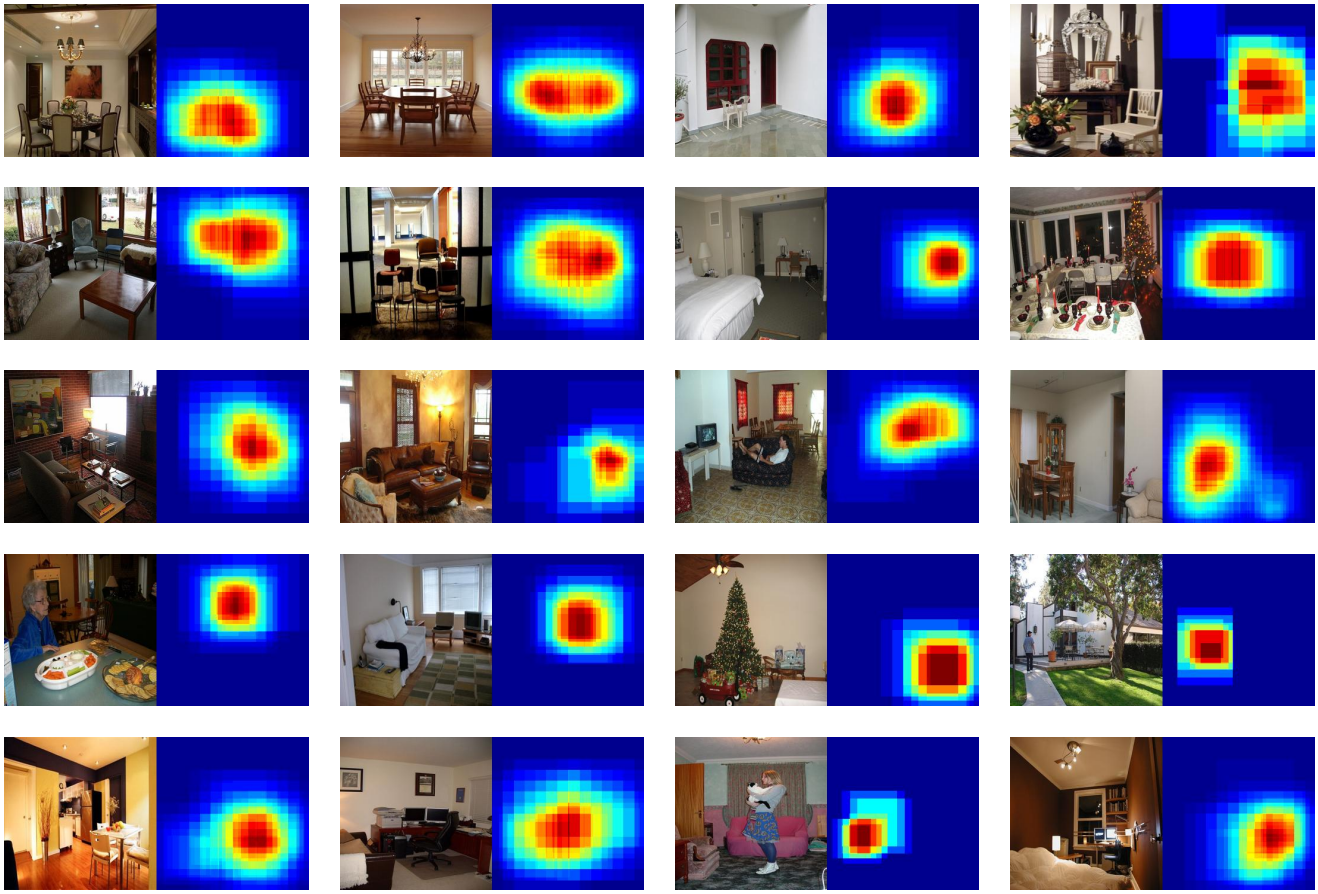


(a) True examples of the “cat” class.

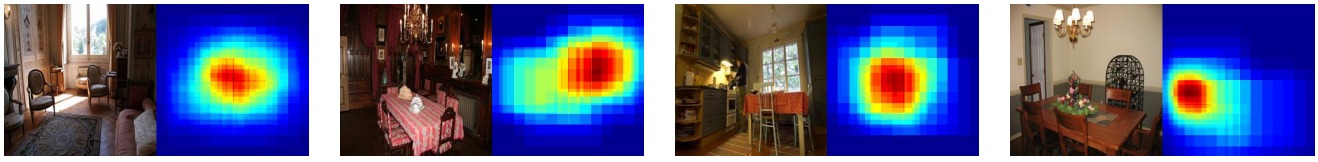


(b) False examples of the “cat” class.

Figure 13. Examples of object confidence map of “cat” class in the PASCAL VOC 2007 test image set.



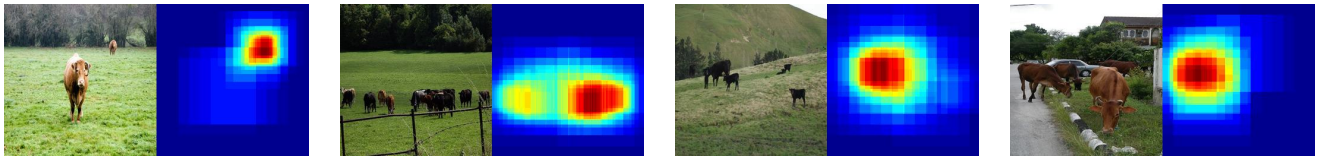
(a) True examples of the “chair” class.



(b) False examples of the “chair” class.

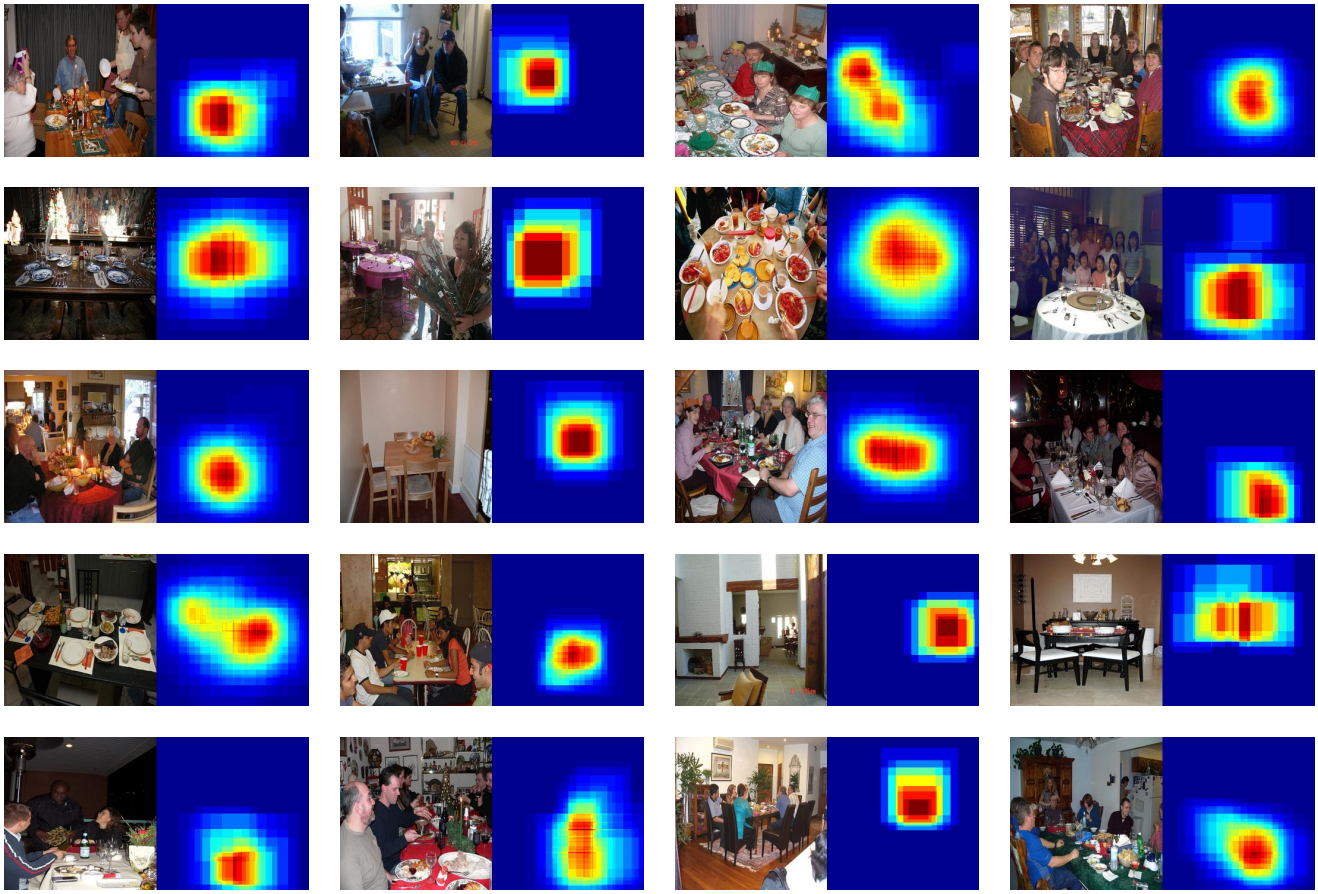
Figure 14. Examples of object confidence map of “chair” class in the PASCAL VOC 2007 test image set.

(a) True examples of the “cow” class.

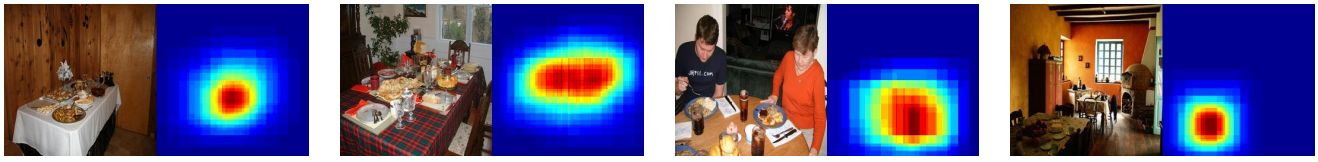


(b) False examples of the “cow” class.

Figure 15. Examples of object confidence map of “cow” class in the PASCAL VOC 2007 test image set.



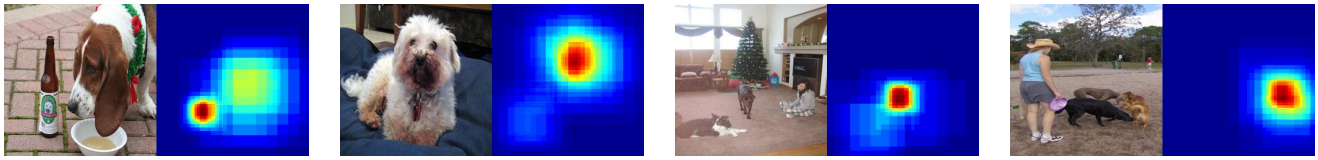
(a) True examples of the “diningtable” class.



(b) False examples of the “diningtable” class.

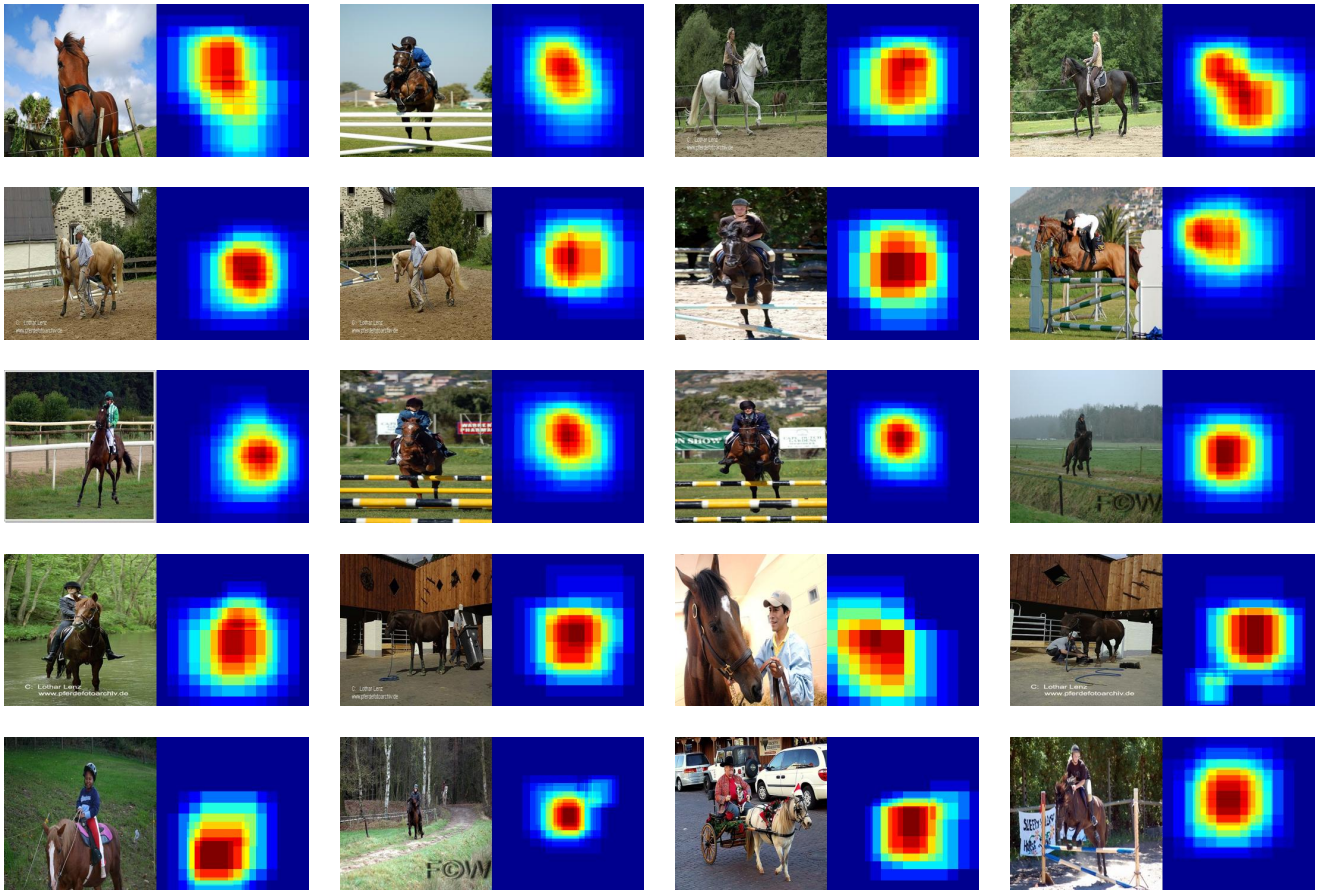
Figure 16. Examples of object confidence map of “diningtable” class in the PASCAL VOC 2007 test image set.

(a) True examples of the “dog” class.

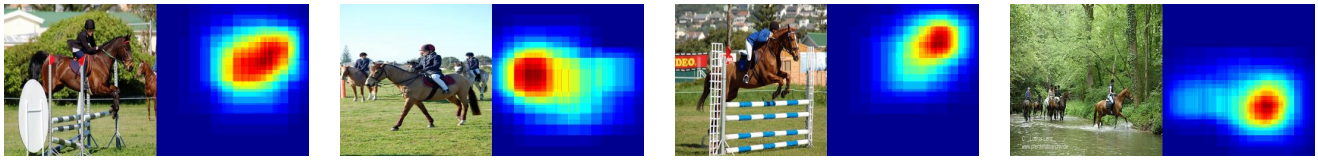


(b) False examples of the “dog” class.

Figure 17. Examples of object confidence map of “dog” class in the PASCAL VOC 2007 test image set.



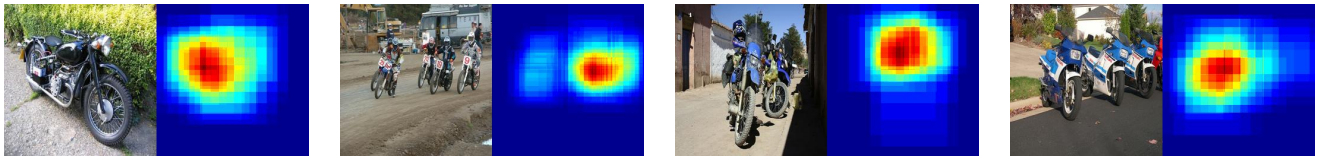
(a) True examples of the “horse” class.



(b) False examples of the “horse” class.

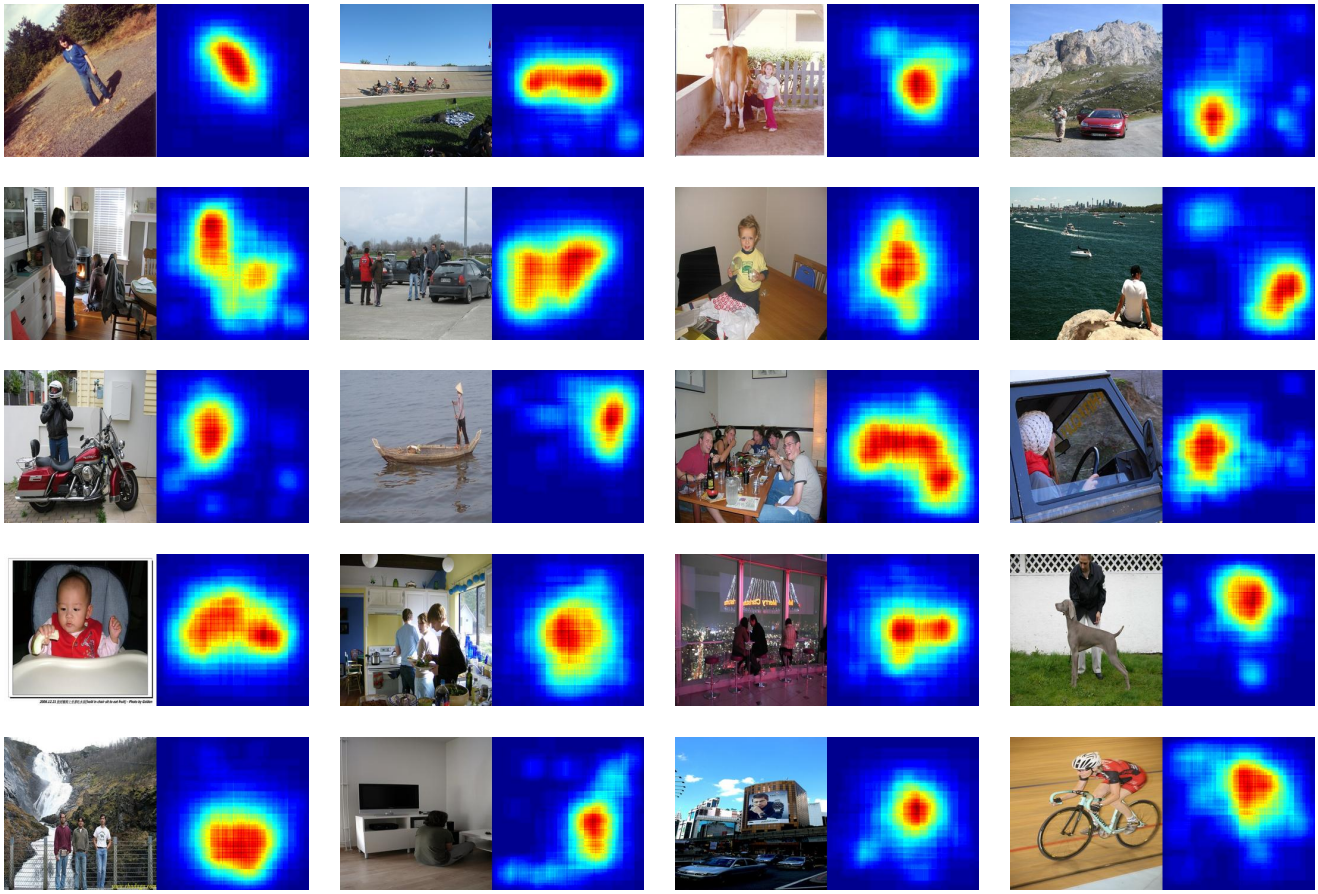
Figure 18. Examples of object confidence map of “horse” class in the PASCAL VOC 2007 test image set.

(a) True examples of the “motorbike” class.

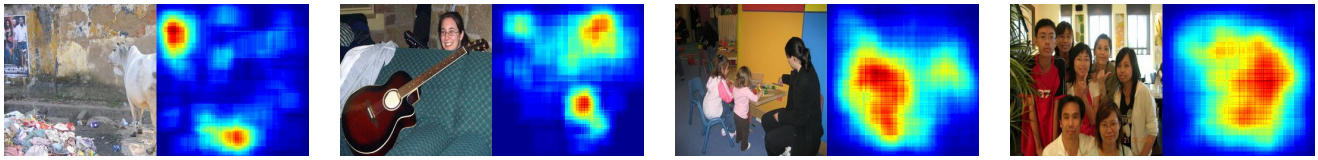


(b) False examples of the “motorbike” class.

Figure 19. Examples of object confidence map of “motorbike” class in the PASCAL VOC 2007 test image set.

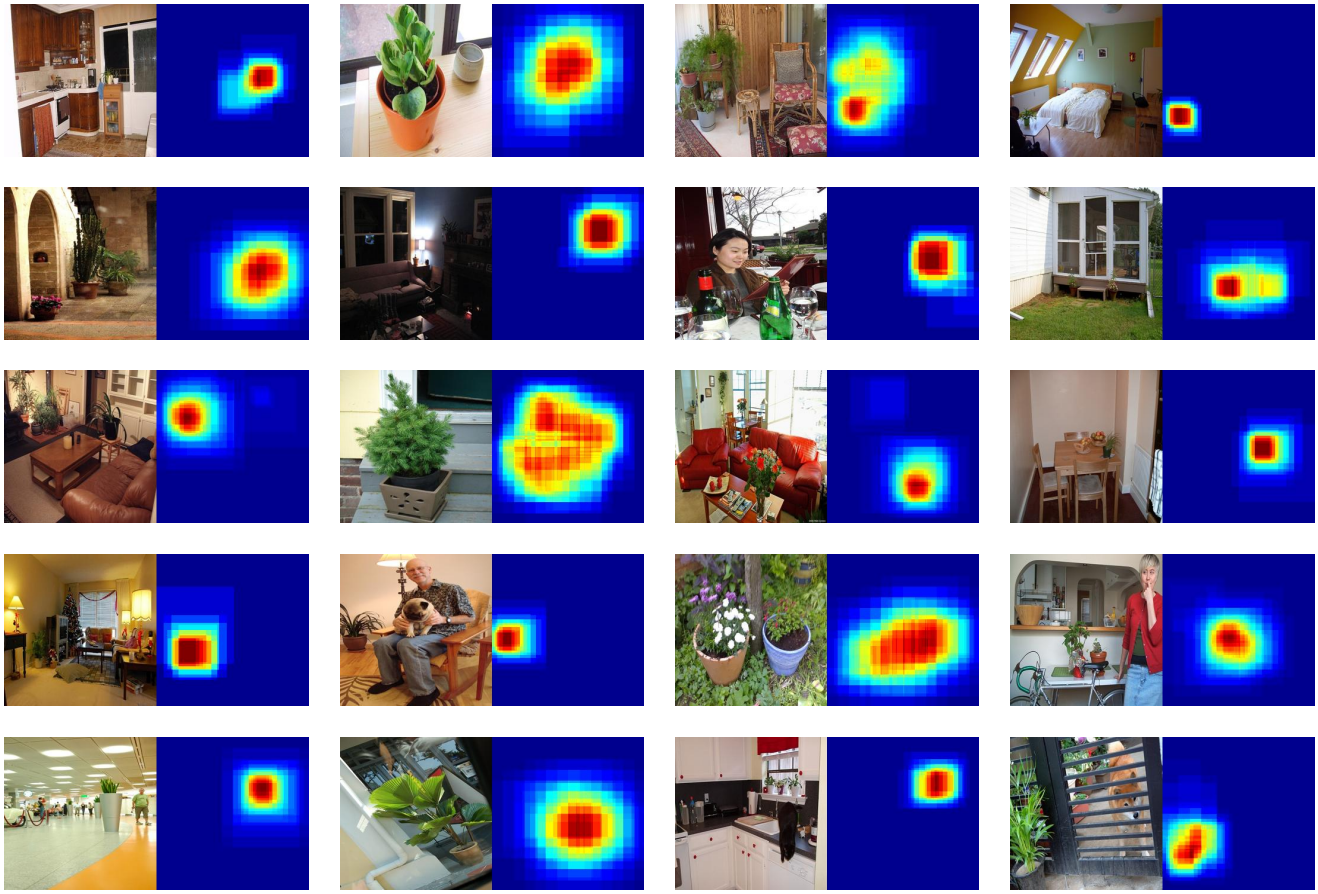


(a) True examples of the “person” class.



(b) False examples of the “person” class.

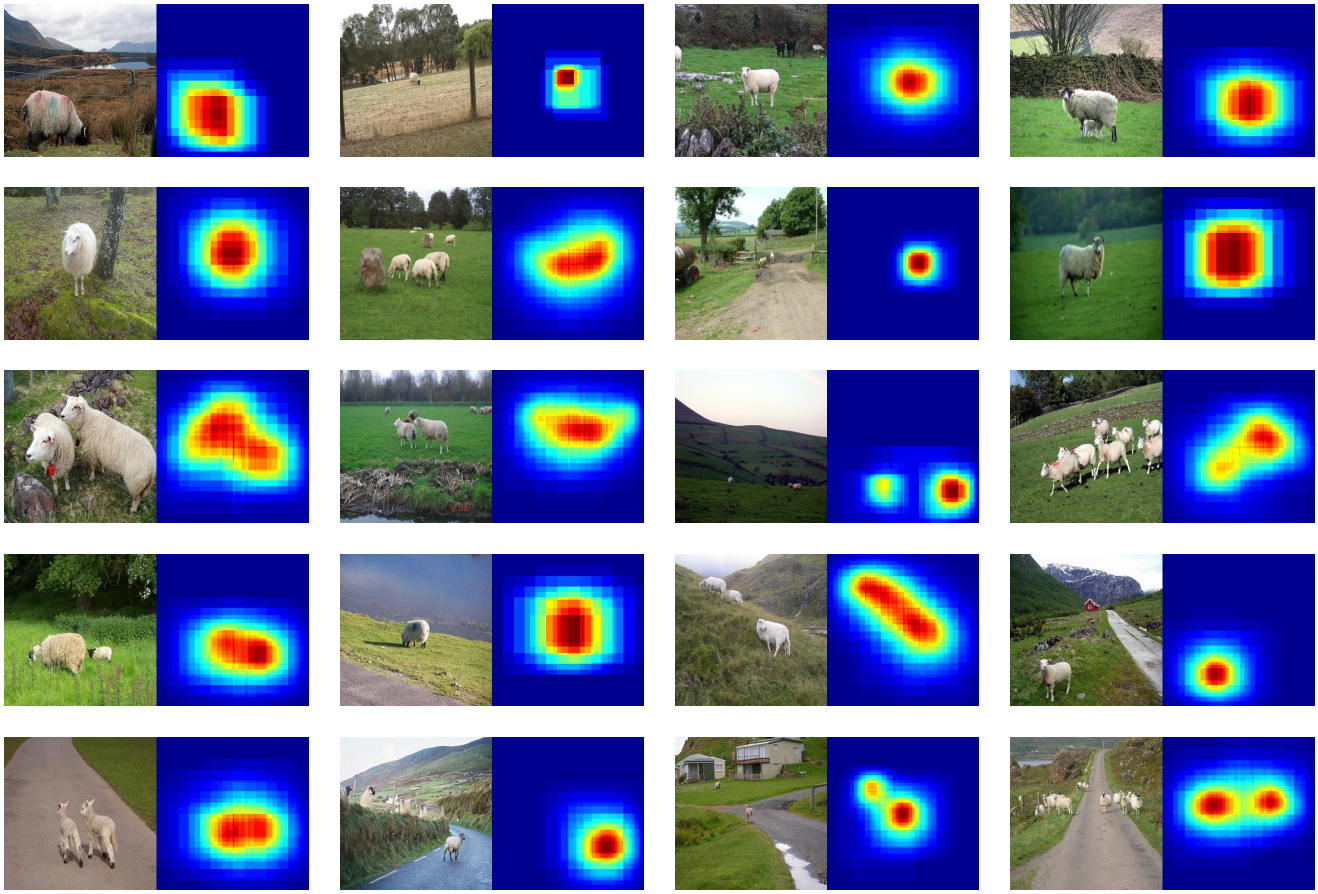
Figure 20. Examples of object confidence map of “person” class in the PASCAL VOC 2007 test image set.



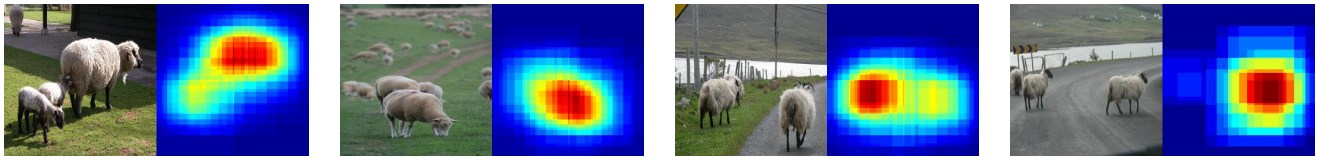
(a) True examples of the “pottedplant” class.

(b) False examples of the “pottedplant” class.

Figure 21. Examples of object confidence map of “pottedplant” class in the PASCAL VOC 2007 test image set.

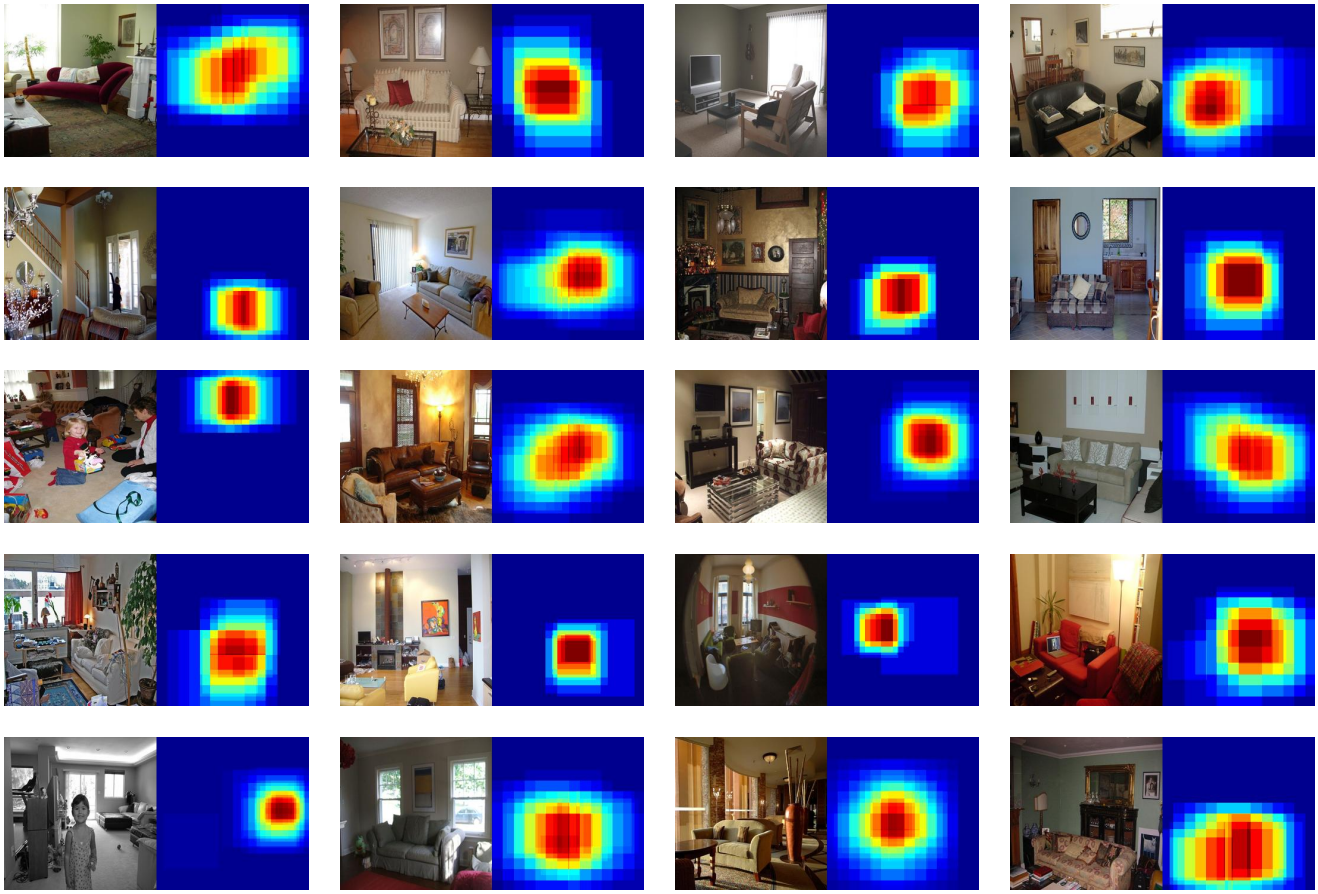


(a) True examples of the “sheep” class.

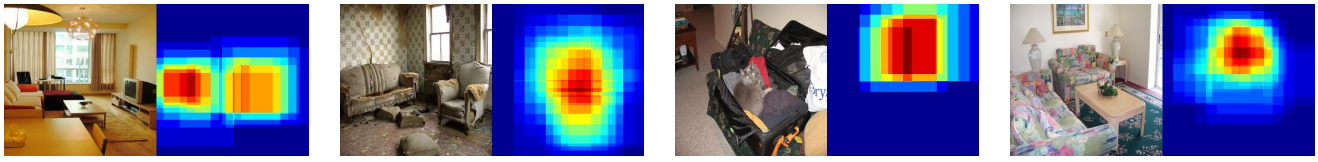


(b) False examples of the “sheep” class.

Figure 22. Examples of object confidence map of “sheep” class in the PASCAL VOC 2007 test image set.

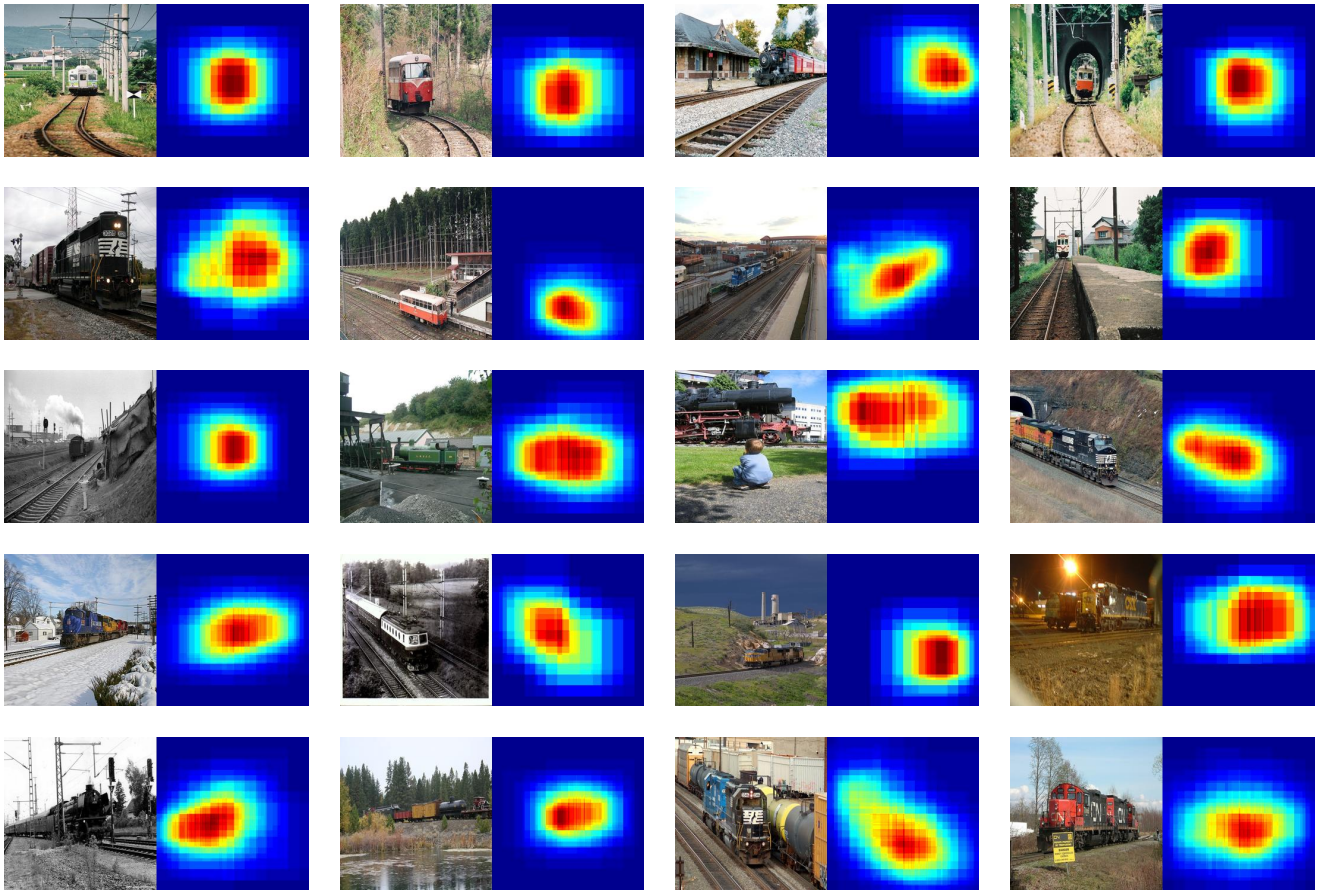


(a) True examples of the “sofa” class.

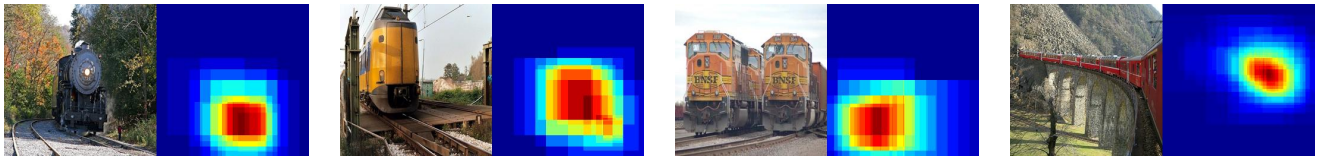


(b) False examples of the “sofa” class.

Figure 23. Examples of object confidence map of “sofa” class in the PASCAL VOC 2007 test image set.

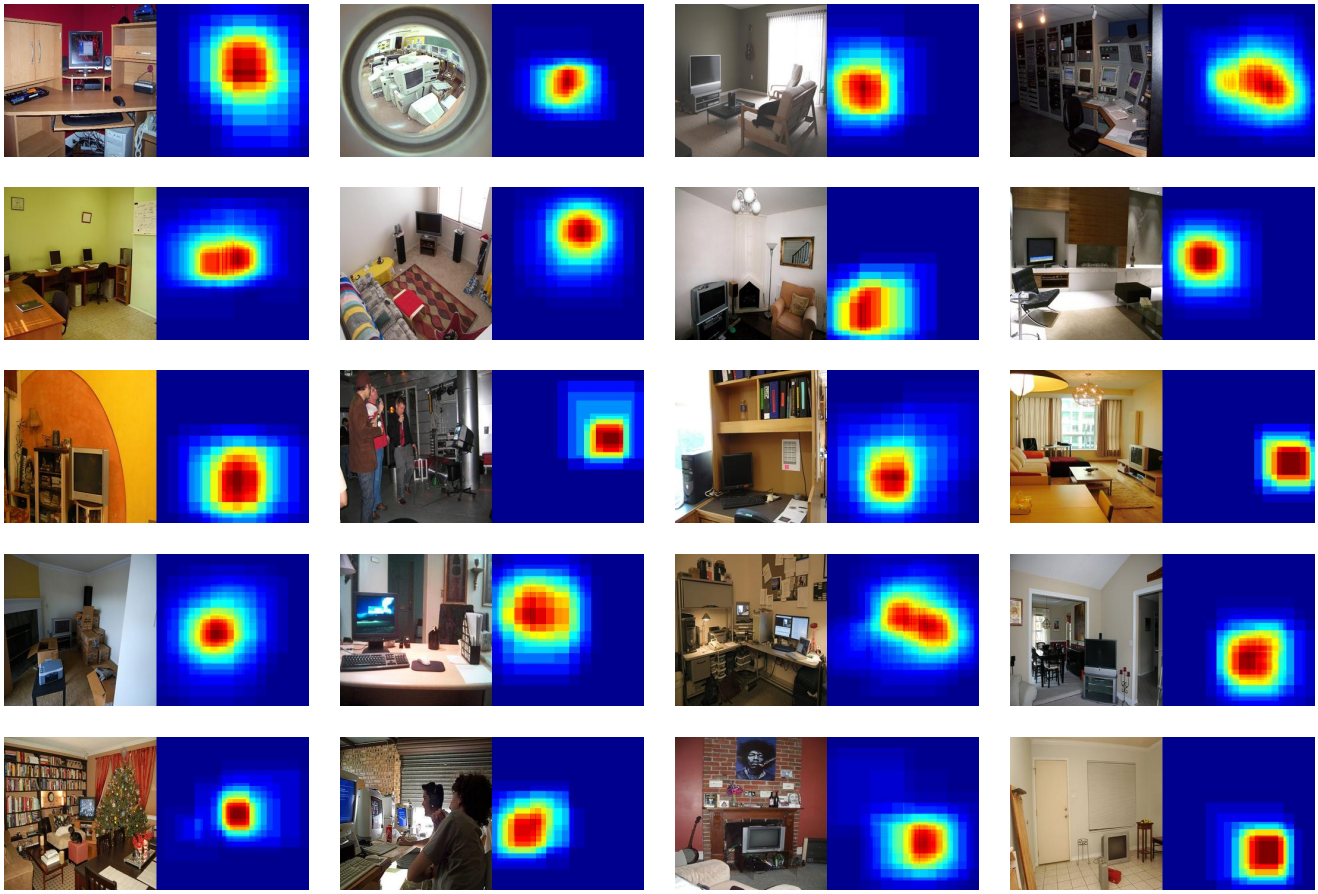


(a) True examples of the “train” class.

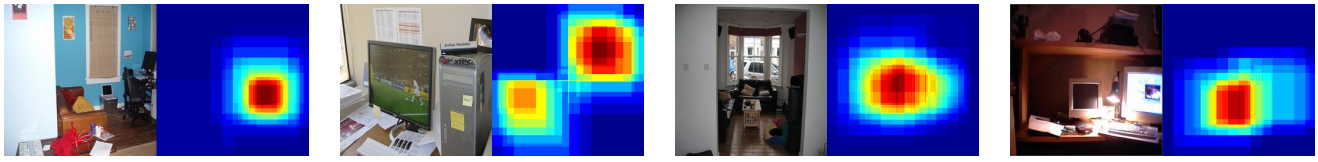


(b) False examples of the “train” class.

Figure 24. Examples of object confidence map of “train” class in the PASCAL VOC 2007 test image set.



(a) True examples of the “tvmonitor” class.



(b) False examples of the “tvmonitor” class.

Figure 25. Examples of object confidence map of “tvmonitor” class in the PASCAL VOC 2007 test image set.

References

[1] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014. 1, 2, 3

[2] A. Vedaldi and K. Lenc. MatConvNet: Convolutional neural networks for matlab. http://www.vlfeat.org/matconvnet/, 2014. 1