Supplementary Materials for "Interpretable Explanations of Black Boxes by Meaningful Perturbation"

Ruth C. Fong
University of Oxford
ruthfong@robots.ox.ac.uk

Andrea Vedaldi
University of Oxford
vedaldi@robots.ox.ac.uk

Contents

1. Extensions 1

2. Mask Initialization 1

3. Comparison Visualization Details 1

4. Pointing game implementation details 2
   4.1. Occlusion variants 2

5. Experimental results using default hyperparameters to minimize one target class 2

6. More examples of learned masks 3
   6.1. Randomly selected examples 3
   6.2. Examples of the top5 hyperparameter setting 3
   6.3. Examples for Different Network Architectures 3
   6.4. Examples For Visualizing Different Layers . 3
   6.5. Examples Comparing the Preservation vs. Deletion Loss 3
   6.6. Failure Examples on Imagenet 7
   6.7. Examples on COCO 7

7. Testing hypotheses: animal parts saliency 7

1. Extensions

To play the "preservation" game and aim to maximize a class score rather than minimize it, eq. 6 becomes:

$$\min_{m\in[0,1]^\Lambda} \lambda_1\|m\|_1 + \lambda_2 \sum_{u\in\Lambda} \|\nabla m(u)\|_\beta^\beta - \mathbb{E}_\tau[f_c(\Phi(x_0(\cdot-\tau), M))], \quad (1)$$

where the jitter parameter $\tau \in \text{Unif}(-\alpha, \alpha)^2$ is drawn from a 2D uniform random variable, $x_0(\cdot-\tau)$ denotes the original image x_0 translated by τ, $M(v) = \sum_u g_{\sigma_m}(v/s - u)m(u)$ is the upsampled mask, and g_{σ_m} is a 2D Gaussian kernel.

To minimize multiple class scores, as we do in the weak localization experiment, eq. 6 becomes:

$$\min_{m\in[0,1]^\Lambda} \lambda_1\|1-m\|_1 + \lambda_2 \sum_{u\in\Lambda} \|\nabla m(u)\|_\beta^\beta + \mathbb{E}_\tau[\sum_{c\in C} f_c(\Phi(x_0(\cdot-\tau), M))], \quad (2)$$

where C is the set of classes we are interested in minimizing.

2. Mask Initialization

We experimented with a few different ways to initialize our masks (e.g. uniformly random from [0,1], all set to 0.5, 0, or 1). We found it was best to initialize the mask with a mask that roughly covered the object enough to suppress the target score. Thus, we initialized our masks to be circular and centered on the image and chose the smallest radius [0:5:175 pixels] that suppressed the normalized score by at least 99% (or preserved the normalized score by at least 99% for the preservation game). If none did so, the mask was set to fully perturb the whole image initially. Finally, the initial circular mask is blurred by $g_{\sigma_m=10}$ to soften the disc edge so as not to trigger artifacts and down-sampled to the final mask size.

3. Comparison Visualization Details

When visualizing and testing other heatmap methods, we use the following settings, usually per their original default implementations:

1. Gradient [6]: Backpropagate error signal to the input data layer and take the inf-norm (this is calculated by taking the maximum value over the color channels per pixel from the absolute value of the gradient, $|\frac{dz}{dx}|$).

2. Guided Backprop [8]: Same as gradient, except use [8]'s modified ReLU backwards rule (i.e., for ReLU layers, apply a ReLU to the backprogated signal).

3. Excitation backprop [11]: Backpropagate signal to "pool2:3x3_s2" layer in Googlenet [9] using the excitation backprop rule and take the sum of the backpropagated signal over the feature channels.

4. Contrast excitation backprop [12]: Same as excitation backprop, except compute and subtract out the contrastive signal at "pool5:7x7_s1" in Googlenet and take the sum of the backpropagated signal over the feature channels after it has been passed through a ReLU.

5. Grad-CAM [5]: Backpropagate gradient to "inception_5b:output" in Googlenet, apply global average pooling (GAP), compute the sum of activations weighted by the GAP of the gradient.

6. Occlusion [10]: Following [5]'s of occlusion size, we use 35×35 mean-pixel value occlusion masks with stride 8. We normalize the resulting change in logit score heatmap by the number of masks applied per spatial location.

For all techniques, we backpropagate the error signal (or measure the normalized change in output score in the case of occlusion [10]) with respect to pre-softmax activations, as per convention for the above methods.

4. Pointing game implementation details

The results from our re-implementation differ slightly from that of [11] because we resized all images to be 224×224 (we confirmed this source of difference by getting the same results when using [11]'s Pointing Game code with resized images). We also average the localization of the maximum points if there are more than one because our method clips a mask to [0,1]; the other methods aren't affected by this averaging because they don't saturate. Finally, due to limited computational power, we tested the Pointing Game on a subset of the COCO validation set ($N = 20,721$, about 50% of all validation images), in which we use at most the first 400 images per class. Without the last two modifications, results for center, gradient [6], guided backprop [8], and the excitation backprop [11] methods are within 1% of those reported in table 2 of the main text.

4.1. Occlusion variants

For blur occlusion, we used circles with diameter 35, softened their edges by applying a Gaussian blur to them ($\sigma = 10$), and used these circular masks to apply a blur perturbation to the image ($\sigma = 10$). This is in contrast to the standard occlusion implementation, which uses squares (in our case, with side lengths of 35) with hard edges to apply a constant perturbation (i.e., mean image pixel value). As we did for standard occlusion, we used a stride of 28 pixels as we did which we slide our disc masks. We hypothesized that our blur occlusion should perform better because it is less likely to cause artifacts thanks to its blur perturbation and mask softening.

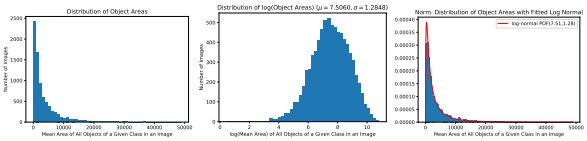

Figure 1. **(Left)** Empirical distribution of object area. **(Middle)** Log of empirical distribution. **(Right)** Log-Normal distribution ($\mu = 7.5060, \sigma = 1.2848$) fit to empirical distribution (left).

For variable occlusion, we drew samples for the radius and x- and y-center coordinates of circular masks that are then smoothed by blurring ($\sigma = 10$) and used to apply a blur perturbation ($\sigma = 10$) (this is similar to blur occlusion except that the masks are variable is size and randomly placed). Then, out of the top 10% most score-suppressing masks, we chose the one which had the smallest radius and used its center as our point (instead of generating a heatmap and picking its maximum point). We hypothesized that if a small, highly-suppressive mask was a good indication of saliency, variable occlusion should perform well.

To determine the distributions from which to sample the radii and center coordinates, we computed the empirical distribution of the object area and bounding box x- and y-center coordinates on a held-out COCO training set, which consisted of 100 images for each of the 80 classes ($N = 8000$). For a given image and class, we calculated the mean object area over all instances; fig. 1 shows this empirical distribution (left) and that a log normal distribution with $\mu = 7.5060$ and $\sigma = 1.2848$ fits well to the empirical distribution of object areas (right). We use this distribution A to draw radii R as follows: $R = \sqrt{\frac{A}{\pi}}$. We accumulate the x- and y- center coordinates of bounding boxes for all annotated objects in the heldout set; fig. 2 shows the empirical distribution (left) and that a normal distribution ($\mu = 116.0759, \sigma = 47.5054$) fits well to the empirical distribution of the y-coordinate (right). Due to the sharp, center peakiness and otherwise flatness of the empirical distribution for the center x-coordinate, a normal distribution did not fit it well and instead we use a uniform distribution so as to avoid a strong center bias. From these distributions, we drew the x- and y- center coordinates. We used the same number of masks used in standard and blur occlusion ($N = 784 = 28 \times 28$ stride).

5. Experimental results using default hyper-parameters to minimize one target class

For the weak localization experiment, using the default hyper-parameter settings to minimize the target class resulted in a accuracy rate of 41.9% on the training held-out

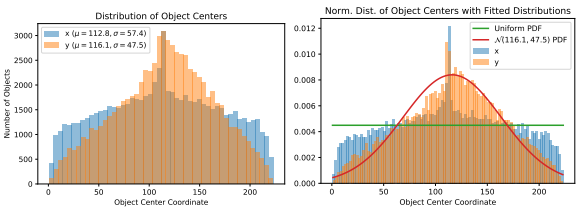

Figure 2. **(Left)** Empirical distribution of x- and y- center coordinates. **(Right)** Normal distribution, $\mathcal{N}(116.0759, 47.5054)$, and uniform distribution respectively fit to the y- and x- center empirical distributions.

	Value	Energy	Mean
α^*	0.25	0.90	1.0
Val Err	48.4%	46.1%	46.0%

Table 1. Weakly supervised localization results on validation set ($N = 50,000$) when using value, energy, and mean thresholding when the default hyper-parameters to minimize the ground truth class was used.

set ($N = 5000$) when using mean thresholding ($\alpha^* = 1.0$). By comparison, the top-5 hyper-parameter setting yielded a held-out accuracy rate of 39.6% ($\alpha^* = 0.5$, Table 1 in main text). Thus, only the top-5 hyper-parameter setting was used for the weak localization experiments. Table 1 shows the localization results on the validation set of using the default hyper-parameter settings to minimize the target class; there results have 3-4% higher than those for using the top-5 hyper-parameter setting.

For the pointing game, using the default settings to minimize the target class resulted in the following precision rates, which are 1% and 2% lower than that of the top-5 hyper-parameters setting reported in the main text: 36.40% (all) and 28.21% (difficult).

6. More examples of learned masks

6.1. Randomly selected examples

Figure 3 shows randomly selected examples of learned masks and compares them to visualizations from other saliency methods.

6.2. Examples of the top5 hyperparameter setting

For the weakly supervised object localization task, in order to get more object coverage, we minimized the softmax scores of the top 5 predicted classes (eq. (2)). Furthermore, in order to get sharper, less coarsely smoothed masks, we used $\lambda_1 = 10^{-3}$ and $\beta = 2.0$ and otherwise default parameters. Figure 4 shows randomly selected examples of masks using the default settings and the settings used for the weak localization experiment.

Figure 5 shows all 50 ImageNet [4] validation examples for "sunglasses" with the mask learned using default parameters overlaid, while fig. 6 shows those examples with the mask learned using the weak localization parameters, which minimize the top 5 predicted classes and thus show what the network was "distracted" by. Because "sunglasses" had the second lowest validation classification rate of 8%, our method illustrates in fig. 5 that the network has learned a reasonable understanding of the object by highlighting glasses in most images.

6.3. Examples for Different Network Architectures

Figure 8 compares masks learned by our method on the Alexnet [1], VGG-16 [7], and Googlenet [9] architectures. The default hyper-parameters, which were selected for Googlenet, were used, with the exception that for Alexnet, the mask scaling factor was equal to $7.5\bar{6}$ in order to rescale its slightly larger default input size of 227×227 to the down-sampled 32×32 mask size. Qualitatively, masks for all three networks generally identify the same or similar parts of the image, though the Googlenet results look slightly better; this is likely due to the fact that the hyper-parameters used were tuned to Googlenet.

6.4. Examples For Visualizing Different Layers

For visualizing intermediate layers, we used the following parameters: $N = 300$ iterations, learning rate $\gamma = 10^{-1}$, TV $\beta = 3$. The mask is not upsampled, blurred, or jittered ($\delta = 1, g_{\sigma_m=0}, \tau = 0$) and constant perturbation, which equivalent to drop-out in intermediate layers, is used instead of blur.

For AlexNet [1], we used the following parameters to visualize these layers: pool1 (27×27): L1 $\lambda_1 = 10^{-4}$, TV $\lambda_2 = 10^{-2}$, pool2 (13×13): $\lambda_1 = 5 \times 10^{-3}, \lambda_2 = 10^{-2}$, relu3 ($13 \times 13$): $\lambda_1 = 5 \times 10^{-3}, \lambda_2 = 10^{-2}$, relu4 ($13 \times 13$): $\lambda_1 = 5 \times 10^{-3}, \lambda_2 = 10^{-2}$, ahd pool5 ($6 \times 6$): $\lambda_1 = 10^{-2}, \lambda_2 = 10^{-3}$.

For VGG-16 [7], we used the following layers and parameters: pool1 (112×112): $\lambda_1 = 5 \times 10^{-5}, \lambda_2 = 10^{-2}$, pool2 ($56 \times 56$): $\lambda_1 = 10^{-4}, \lambda_2 = 10^{-2}$, pool3 ($28 \times 28$): $\lambda_1 = 10^{-4}, \lambda_2 = 10^{-2}$, pool4 ($13 \times 13$): $\lambda_1 = 5 \times 10^{-3}, \lambda_2 = 10^{-2}$, ahd pool5 ($7 \times 7$): $\lambda_1 = 10^{-2}, \lambda_2 = 10^{-3}$.

Not that the default parameters were chosen for a 28×28 mask; the above parameters are correlated to activation size, which equals mask size for these layers. As the activation size decreases, a stronger L1 regularization (λ_1) and weaker TV norm regularization (λ_2) are needed.

6.5. Examples Comparing the Preservation vs. Deletion Loss

Figure 9 shows masks learned using the preservation loss (eq. (1)) and the deletion loss (main text, eq. (6)). Qualitatively, the deletion loss results in better heat maps; this is because there is always some positive evidence for the target class that can be deleted, whereas for a difficult image,

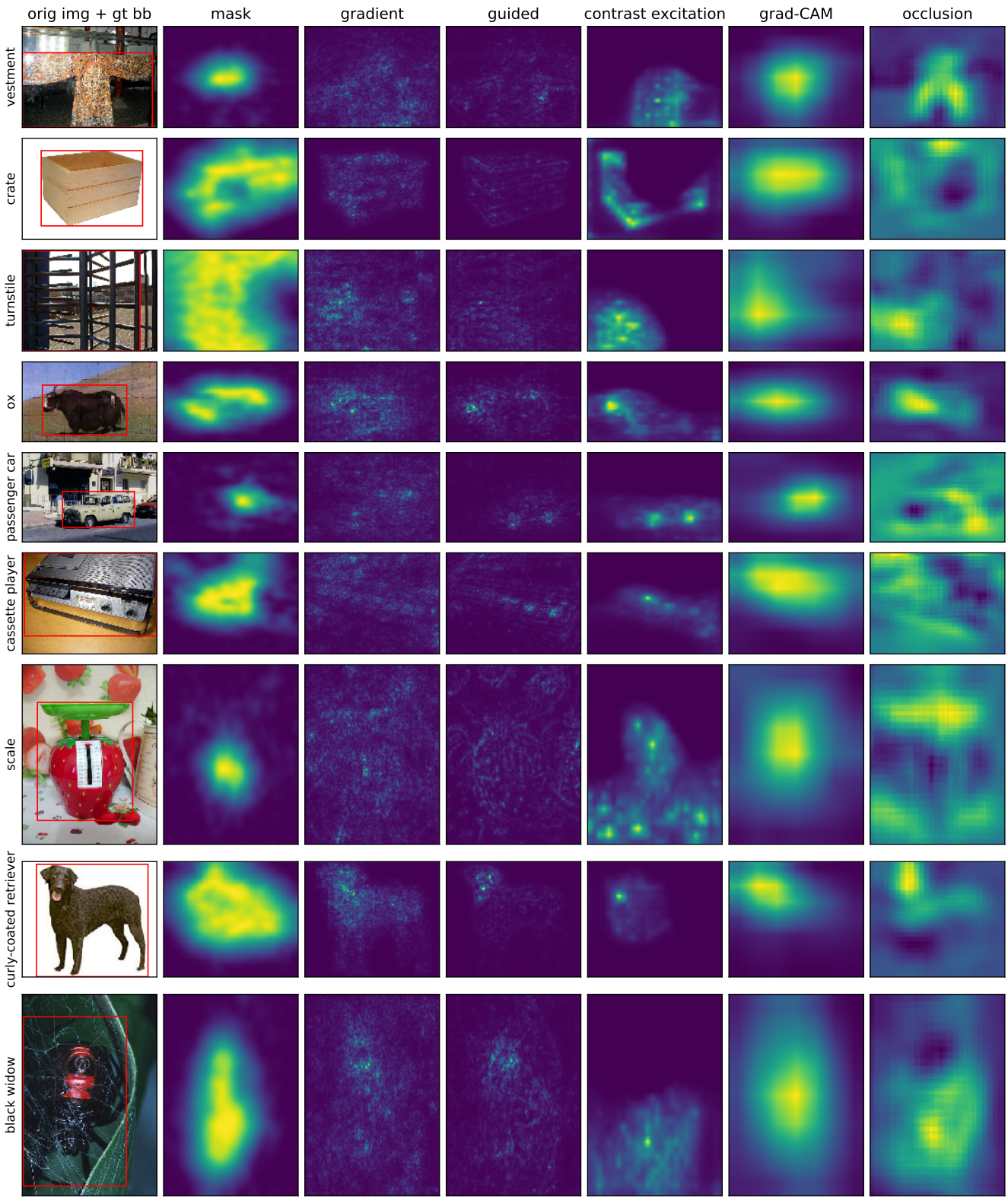

Figure 3. Comparison with other saliency methods (images randomly selected). From left to right: original image with ground truth bounding box; learned mask subtracted from 1 (our method); gradient [6]; guided backprop [8, 3]; contrastive excitation backprop [11]; grad-CAM [5]; and occlusion [10].

Figure 4. Comparison between two mask settings, e.g. minimizing target class and top 5 predicted classes, with the latter using $\lambda_1 = 10^{-3}$ and $\beta = 2.0$ (images randomly selected).

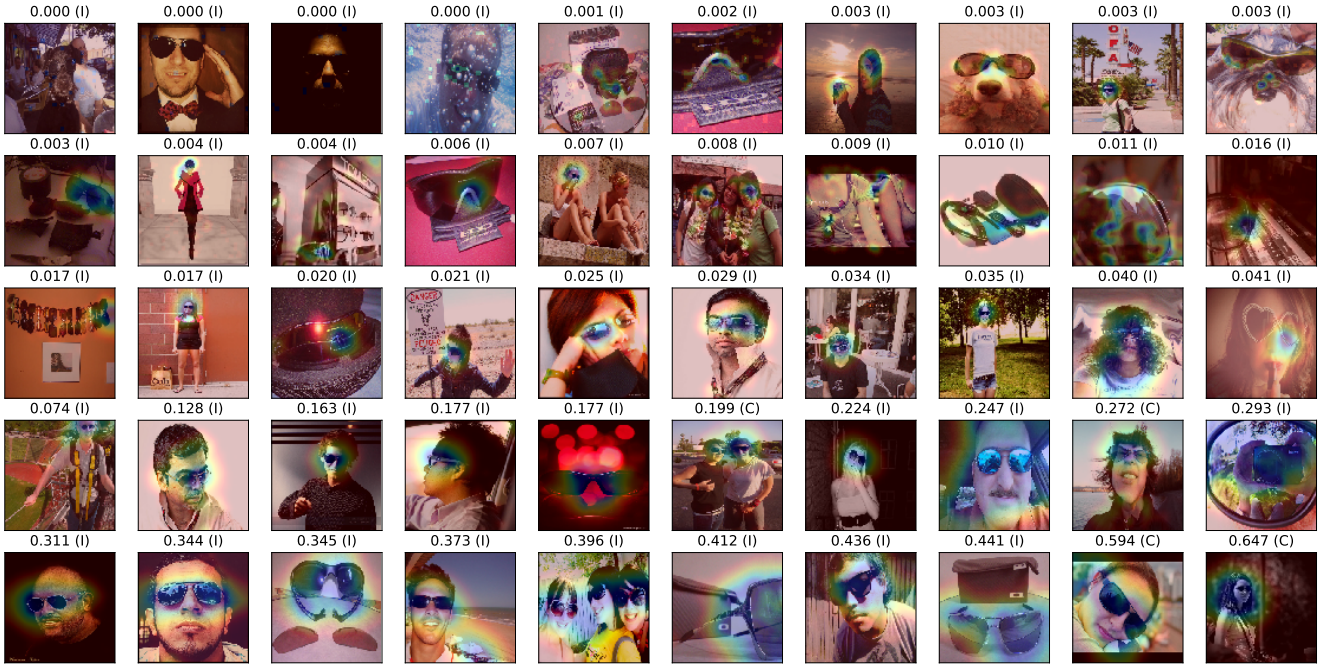

Figure 5. 50 ImageNet [4] "sunglasses" validation examples with mask using default settings overlaid (Googlenet [9] "sunglasses" softmax probability and whether it was classified correctly (C) or incorrectly (I) above). Despite having a validation classification accuracy of 8%, these visualizations show that the network had a reasonable concept of a pair of "sunglasses" by consistently highlighting glasses in most images.

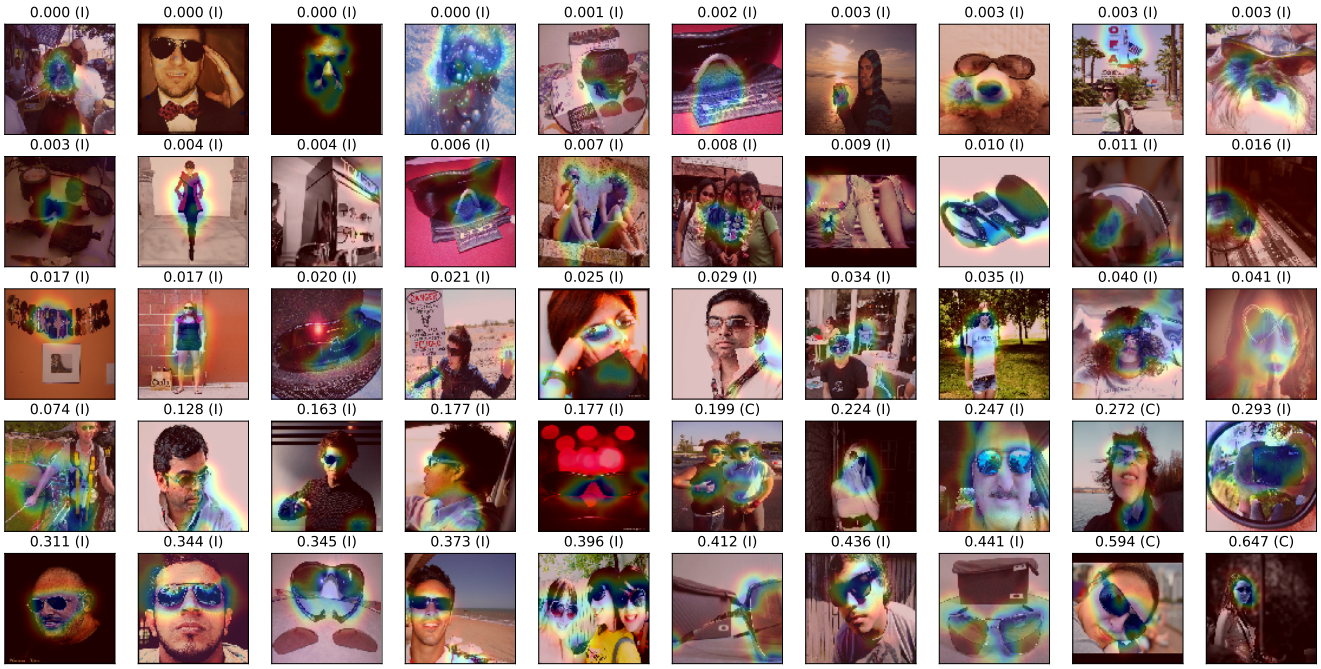

Figure 6. 50 ImageNet [4] 'sunglasses' validation examples with mask overlaid using localization parameters (minimize top 5 predicted classes, $\lambda_1 = 10^{-3}$, $\beta = 2.0$). In comparison to fig. 5, this figure shows what the network was "distracted" by.

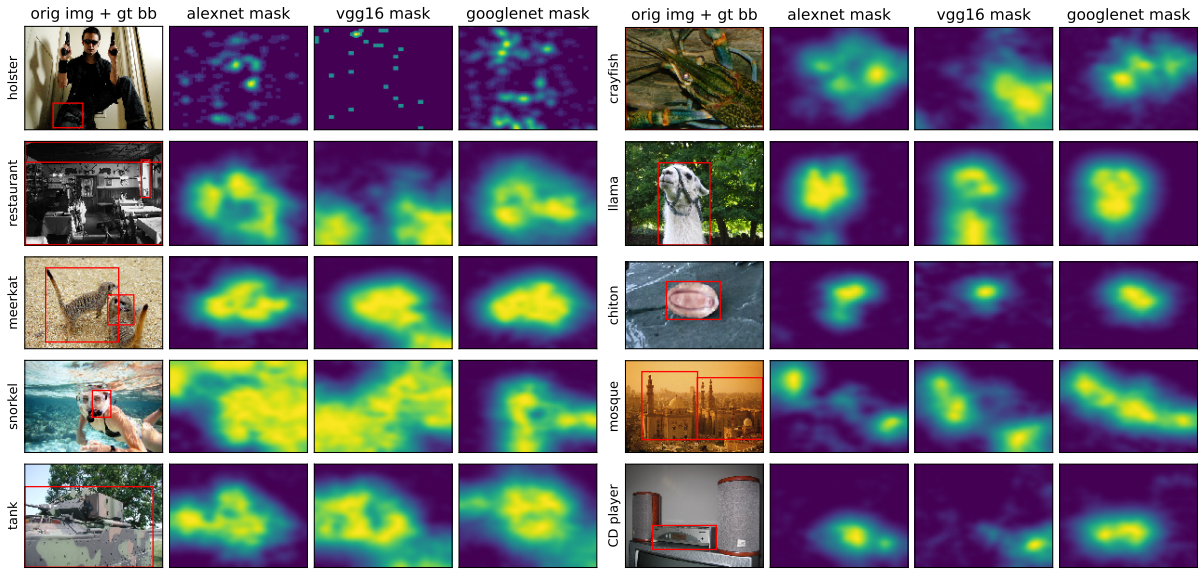

Figure 7. Comparison among the following network architectures: AlexNet [1], VGG-16 [7], and GoogLeNet [9] (images randomly selected). Masks were generated using default parameters, except the mask scaling factor was equal to $7.5\bar{6}$ for AlexNet in order to rescale its slightly larger default input size of 227×227 to the down-sampled 32×32 mask size.

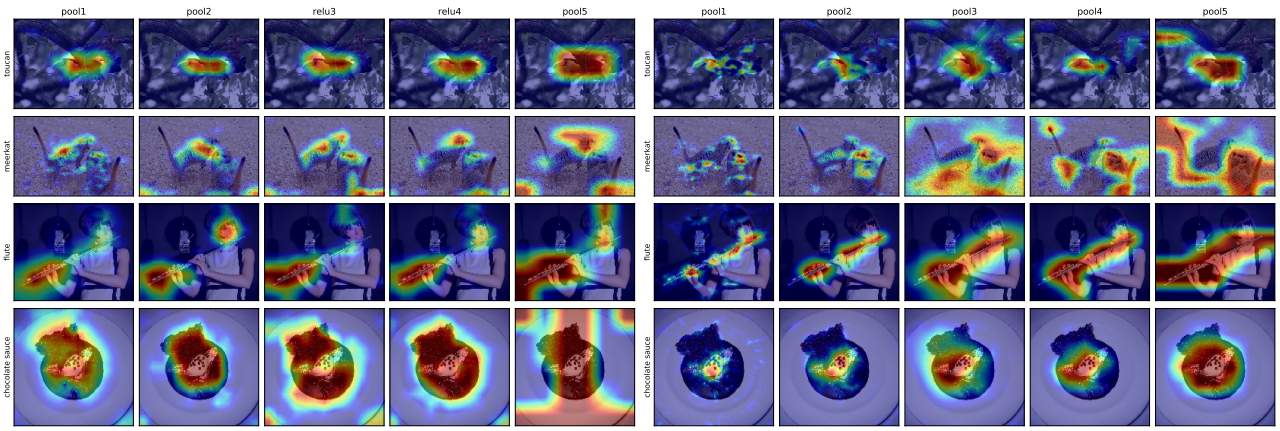

Figure 8. Visualizing different layers. (Left) AlexNet [1]; (Right) VGG-16 [7]. See section 6.4 for parameters used; masks are subtracted from 1, so red corresponds to an original mask value close to 0 (e.g., where deletion occurs).

in which the target score is already quite low in the original image, it may be difficult to amplify the existing positive evidence using a mask without introducing artifacts.

6.6. Failure Examples on Imagenet

The masks for the "vestment" and "passenger car" images in rows 1 and 5 of fig. 3 cover only a small portion of the object while the top-5 mask for "picnic fence" in fig. 4, left row 5 highlights grass.

Another rare failure case occurs when numerical instabilities cause the optimization to not converge; interestingly, fig. 8 left row 1 shows an example of a particularly difficult "holster" image, in which the "holster" is hard to discern even for a human, for which numerical instability affects masks learned for several architectures. This suggests that convergence failure may not just be a failure case but may also provide useful insight to the quality of an image. Instability is also observed in a few masks learned using preservation loss (fig. 9).

6.7. Examples on COCO

Figure 10 shows learned masks on the COCO dataset [2]; qualitatively, the COCO masks look better for simpler images (e.g., those more similar to ImageNet) and worse for small and/or occluded objects, though it does surprisingly well for very small traffic lights.

7. Testing hypotheses: animal parts saliency

Figure 11 shows the mean intensity around each foot and eye keypoint for each of the 76 animal classes and demon-

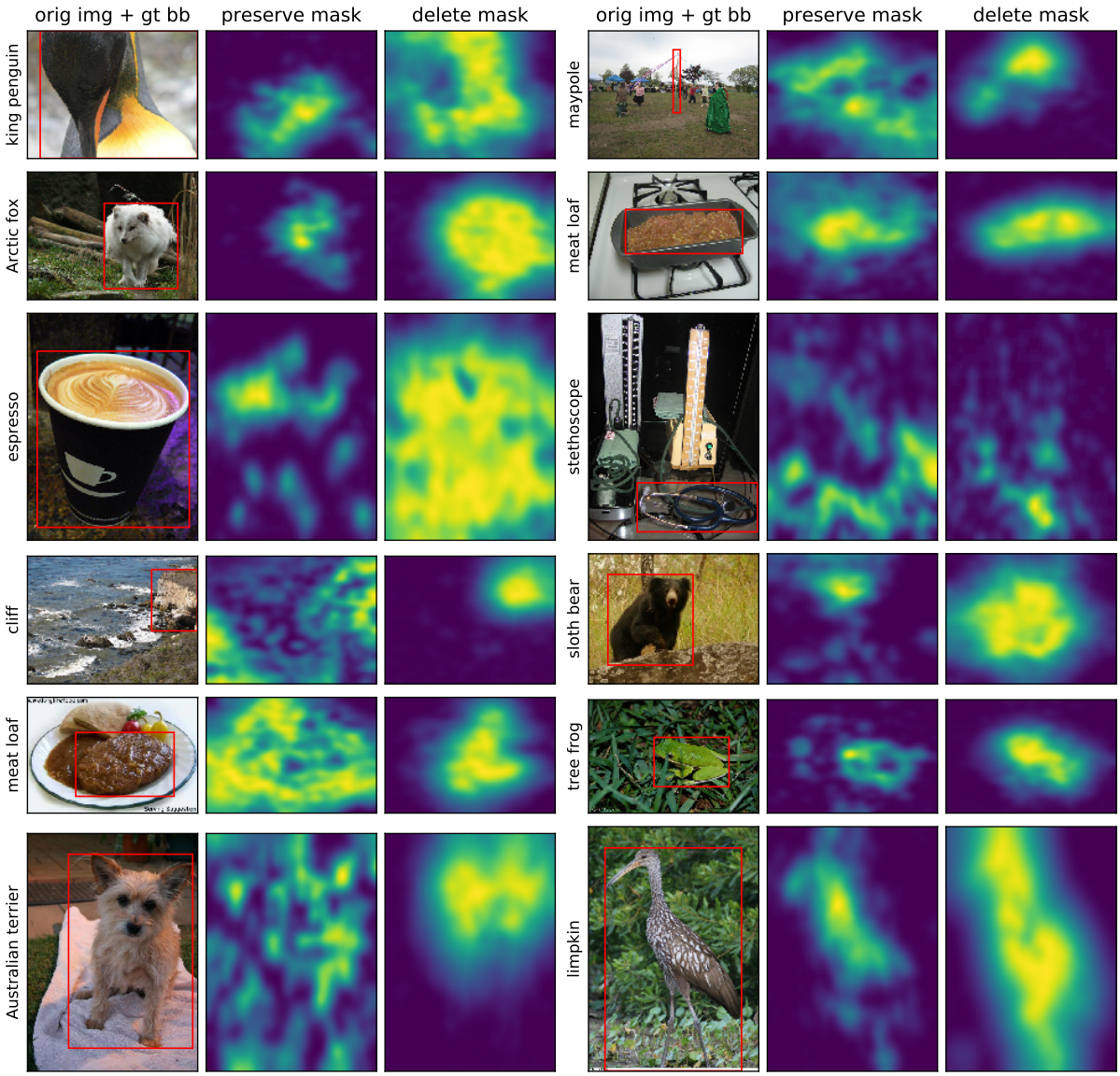

Figure 9. Comparison between using the preservation loss, where the target class score is maximized, versus the deletion loss, where the target class score is minimized (the deletion heat maps are subtracted from 1 to visually match the preservation ones).

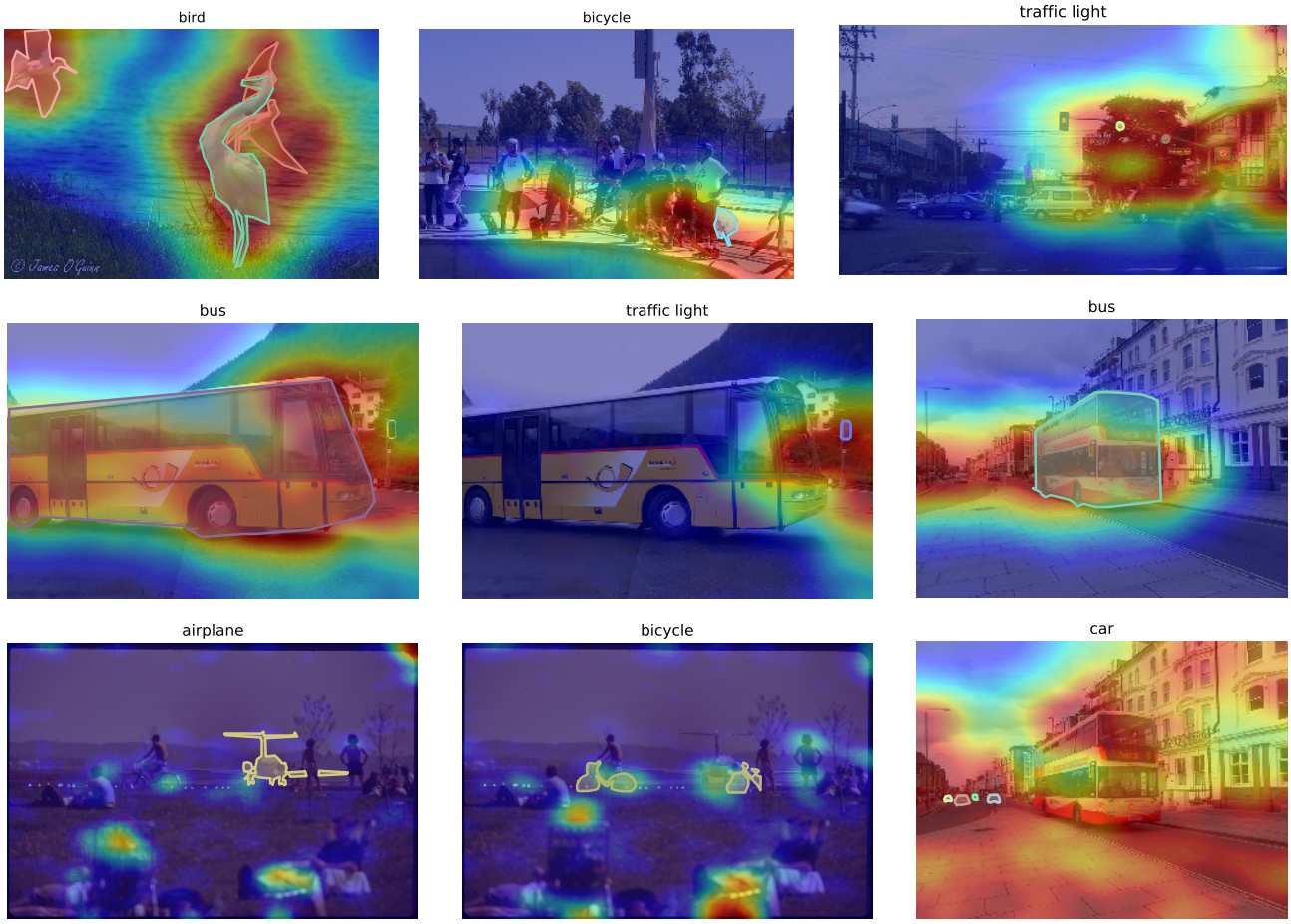

Figure 10. Examples on COCO using default parameters, with clear failure cases in the last row and pairs of images with different objects in the last two rows. Ground truth segmentations are highlighted in different color blocks per object instance while the mask is overlaid using the jet color scheme.

	N	Mean Avg Feet:Eyes Ratio±SE
Small Mam.	3	4.03±0.25
Dog	8	3.70±0.50
Medium Mam.	14	3.49±0.32
Bird	21	3.03±0.04
Large Mam.	23	2.53±0.17
Amph/Rept	7	1.96±0.23

Table 2. Quantification of how much eyes are more salient than feet for images of different animal categories.

strates that for all classes, eyes were on average more salient than feet (recall that a lower intensity in a deletion mask denotes a more salient region whose deletion suppresses the target class' softmax score). Table 2 shows the mean average feet-to-eyes ratio for groups of animals.

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 3, 7

[2] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 7

[3] A. Mahendran and A. Vedaldi. Salient deconvolutional networks. In *European Conference on Computer Vision*, pages 120–135. Springer International Publishing, 2016. 4

[4] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. 3, 6

[5] R. R. Selvaraju, A. Das, R. Vedantam, M. Cogswell, D. Parikh, and D. Batra. Grad-cam: Why did you say that? visual explanations from deep networks via gradient-based localization. *arXiv preprint arXiv:1610.02391*, 2016. 2, 4

[6] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *Proc. ICLR*, 2014. 1, 2, 4

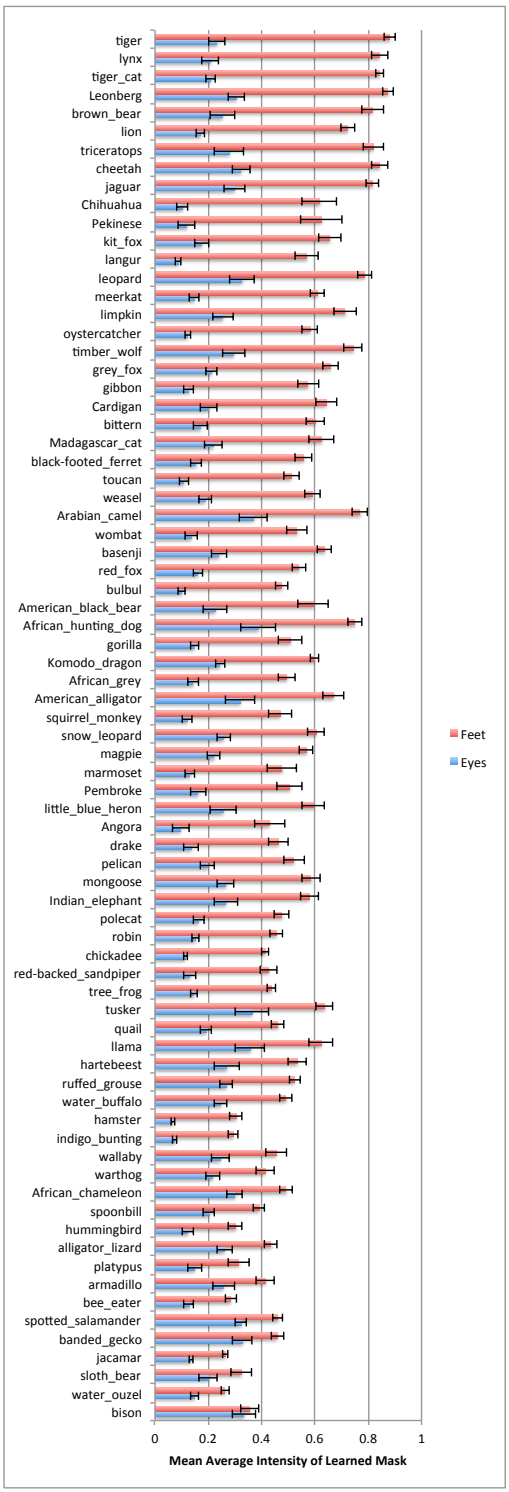

Figure 11. Class-specific differences between average eyes and feet intensity in learned masks (shown with standard error bars). Recall that a lower intensity denotes a more salient region for our "deletion" masks.

[7] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 3, 7

[8] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014. 1, 2, 4

[9] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015. 2, 3, 6, 7

[10] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014. 2, 4

[11] J. Zhang, Z. Lin, J. Brandt, X. Shen, and S. Sclaroff. Top-down neural attention by excitation backprop. In *European Conference on Computer Vision*, pages 543–559. Springer, 2016. 2, 4

[12] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2921–2929, 2016. 2