SCOUTER: Slot Attention-based Classifier for Explainable Image Recognition (Supplementary Material)

Liangzhi Li, Bowen Wang, Manisha Verma, Yuta Nakashima,
Ryo Kawasaki, Hajime Nagahara
Osaka University, Japan

1. Computational Costs

Table 1 gives the cost comparison of SCOUTER and FC classifier. We can see that, compared with the FC classifier, SCOUTER requires a similar computational cost (slightly higher) and a similar number of parameters (slightly lower). The increase in the computational cost (flops) is because the xSlot module has some small FC layers (*i.e.*, Q and K), GRU, and some matrix calculations. However, as shown in the lower part of Fig. 1, this is not very significant.

On the other hand, as shown in the upper part of Fig. 1, SCOUTER has more parameters than the FC classifier when n is roughly in $[0, 90]$. This is because the FC layers and GRU, which are shared among all slots, have a certain number of parameters. For $n > 90$, SCOUTER uses fewer parameters than the FC classifier because there are only c' (64 in our implementation) learnable parameters for each category. This is much less than the parameter size of the FC classifier, which usually needs much more parameters per class ($2,048$ parameters for ResNet 50).

Comparing to the differences in the computation costs and the numbers of parameters of different backbone models, the additional cost of SCOUTER is almost negligible.

Table 1. Cost comparison of SCOUTER and FC classifier ($n = 100$ and input images are with the size of 260×260).

Models	Params (M)		Flops (G)	
	FC	SCOUTER	FC	SCOUTER
ResNet 26 [4]	14.1511	**14.1298**	**3.4238**	3.4565
ResNet 50 [4]	23.7129	**23.6916**	**5.9830**	6.0171
ResNeSt 26 [17]	15.2253	**15.2041**	**5.1803**	5.2130
ResNeSt 50 [17]	25.6391	**25.6179**	**7.7430**	7.7762
DenseNet 121 [7]	7.0564	**7.0719**	**3.7536**	3.7805
DenseNet 169 [7]	12.6510	**12.6435**	**4.4396**	4.4683
MobileNet 75 [5]	1.1194	**0.6537**	**0.0563**	0.0812
MobileNet 100 [5]	4.3301	**3.0859**	**0.3154**	0.3421
SeResNet 18 [6]	11.3169	**11.3509**	**2.6473**	2.6726
SeResNet 50 [6]	26.2439	**26.2226**	**5.6758**	5.7098
EfficientNet B2 [14]	7.8419	**7.8437**	**1.0250**	1.0564
EfficientNet B5 [14]	28.5457	**28.5244**	**3.6391**	3.6721

Figure 1. Flops and parameter sizes of SCOUTER and FC classifier with ResNet 50.

2. Components of xSlot Attention Module

In SCOUTER, we adopt a variant of the slot attention [9]. We make some essential modifications to several components in order to enable explainable classification, while other components, *i.e.* the gated recurrent unit (GRU) and position embedding (PE), remain unchanged, whose effects on the classification as well as the explainability are unexplored. To test the performance of the SCOUTER with and without these components, we consider two variants of SCOUTER. The first one is the SCOUTER without GRU, in which we replace the GRU component, which is used to update slot weights, with an average operation. The second variant is the SCOUTER without PE, where flattened input features are directly used without adding position information.

In Table 2, we show the performances of SCOUTER$_+$ and SCOUTER$_-$ as well as their variants in several performance metrics including computation costs, classification accuracy, and explainability. We can see that SCOUTER with all the components gets better results in most metrics than the variants, except for computation costs. The absence of GRU or PE not only causes a decrease of the classification accuracy, but also some deterioration on all explainability metrics, which proves their necessity.

Table 2. Performance comparison of SCOUTER and its variants on a subset ($n = 100$) of the ImageNet dataset. λ is set to 10 during training and ResNeSt 26 is adopted as the backbone. The explanation performance is measured on the GT category for the positive explanation and on the least similar class (LSC) for the negative explanation.

Explanation Type	Variants	Computational Costs		Classification	Explainability		
		Params (M)	Flops (G)	Accuracy	Precision	IAUC	DAUC
Positive	SCOUTER$_+$	15.2041	5.2130	**0.7991**	**0.9257**	**0.7647**	**0.2713**
	w.o. GRU	**15.1791**	**5.1901**	0.7961	0.9219	0.7456	0.2866
	w.o. PE	15.2041	5.2130	0.7974	0.8973	0.7557	0.3002
Negative	SCOUTER$_-$	15.2041	5.2130	**0.7946**	**0.8101**	**0.6730**	**0.7333**
	w.o. GRU	**15.1791**	**5.1901**	0.7910	0.7904	0.5959	0.7529
	w.o. PE	15.2041	5.2130	0.7903	0.8067	0.6141	0.7661

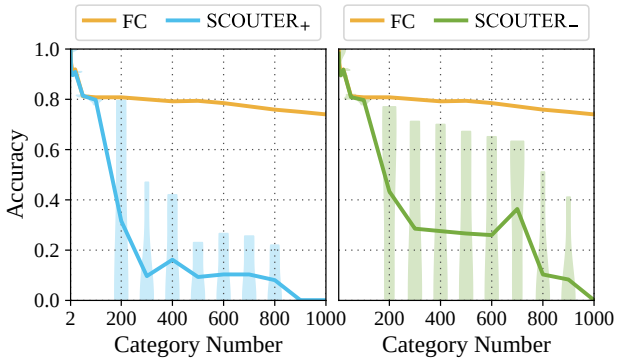


Figure 2. The classification performance of FC classifier, SCOUTER$_+$, and SCOUTER$_-$ when $2 \leq n \leq 1000$. We show the violin plots as well as the average value for SCOUTER$_+$ and SCOUTER$_-$, while the FC classifier is only with the average value.

3. Classification Performance when $n > 100$

Training of SCOUTER becomes unstable when the category number n of the ImageNet [2] subsets is larger than 100. One possible reason is that it is difficult to find consistent and discriminative supports when there are many categories. Fig. 2 shows the classification performance when $n > 100$. The number of independent runs of training is increased to 5 as the training process becomes unstable and often results in failures (low classification accuracy) when $n > 100$. λ is set to 10. ResNeSt 26 [17] is adopted as the backbone, with batch size of 70 and training epoch number of 20 (both are same as the settings of the experiments in the main paper). We can see that, although sometimes SCOUTER$_+$ and SCOUTER$_-$ can achieve similar performance with the FC classifier when $n < 400$, they become significantly unstable with the increase of category number n. As stated in the main paper, SCOUTER can only be used in small-or medium-sized datasets due to this issue.

4. Inter-and Intra-Category Explanation

To better understand (i) what supports SCOUTER uses as the basis for its decision making, (ii) how these supports can be differentiated among different categories, and (iii) whether they are being consistent for images in the same category, we give some additional visualization on the MNIST dataset [8] in Figs. 3 and 4 for SCOUTER$_+$ and SCOUTER$_-$, respectively. MNIST is adopted here as similarities and dissimilarities among categories (digits) are obvious and are easier to understand than ImageNet. In these two figures, (a) is for the inter-category visualization, which shows what the supports for the "Predicted" category look like given the image of the "Actual" category. Whereas, (b) is for intra-category visualization, which shows the support for different images of the same category. For the latter, we use the digit `6` as an example and the first ten samples of category `6` in the test set of MNIST are used.

In the inter-category visualization in Fig. 3, we can see that SCOUTER$_+$ successfully finds supports for the images of ground-truth (GT) categories. Notably, it also finds weaker supports for some categories with similar appearances, *e.g.*, the supports for the prediction of "why `5` is `6`" (as the lower half of this hand-wrote `5` digit is a little confusing and is very close to the lower part of `6`), as well as the prediction of "why `0` is `9`" and "why `8` is `9`" (both `0` and `8` have a circle like the one in `9`).

Similarly, in Fig. 4, we can see that SCOUTER$_-$ finds no supports for the images of the GT categories, while it finds strong supports for the non-GT categories. As digit recognition is an easy task, SCOUTER$_-$ can use some very simple supports to deny most non-GT categories. For example, in the prediction of "why `1` is not [non-GT categories]", all the slots of SCOUTER$_-$ find that the top end of the vertical stroke is `1`'s unique pattern, thus, they can deny all other categories with this support. Among some visually similar categories, the negative explanations are more informative. For example, in the visualization of "why `9` is not `1`" and "why `9` is not `7`", SCOUTER$_-$ precisely highlights the discriminative regions, without which `9` will look like the other two digits.

Also, in intra-category visualization, both SCOUTER$_+$ and SCOUTER$_-$ show consistent supports for the images of the same category. When predicting "why `6` is `6`", SCOUTER$_+$ always looks at the region close to the cross-

ing point of the bottom circle and vertical stroke. For explanation "why 6 is not 2", SCOUTER$_-$ always recognizes the presence of vertical stroke, which does not exist in the digit 2, as well as the missing of the bottom horizon stroke, which is essential for 2.

5. Some More Visualizations

In this section, we show more visualization results for ImageNet using SCOUTER and competing methods, including I-GOS [11], IBA [12], CAM [18], GradCAM [13], GradCAM++ [1], S-GradCAM++ [10], Score-CAM [16], SS-CAM [15], and Extremal Perturbation [3].

Subsets with $n = 100$ categories are used for training and visualization. Besides the first n categories (as used in the main paper), we also use several other subsets (with the same category number) in the ImageNet dataset, in order to provide visualizations with more diversity. Figs. 5 and 6 give examples of the positive explanation, while Fig. 7 gives examples of the negative explanation.

Among the positive explanations, we can see that SCOUTER$_+$ can find reasonable and precise supports. Especially for the image of "parallel bars", SCOUTER$_+$ can provide an explanatory region along the horizon bar. In addition, SCOUTER$_-$ with the least similar class (LSC) also finds supports on the foreground objects, which can be used to deny the LSC categories but are not enough for admitting the GT category, which conforms the quantitative results in the main paper.

Moreover, as shown in Fig. 7, SCOUTER$_-$ can give very detailed explanations when different categories with high visual similarities, *e.g.*, the differences in the eyes and ears between "Labrador retriever" and "golden retriever", and the differences of the horn between "water ox" and "ox".

Figs. 8 and 9 show some more examples of two medical applications (glaucoma diagnosis and artery hardening diagnosis). We can see that SCOUTER$_+$ and SCOUTER$_-$ perform well in both tasks.

References

[1] Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian. Grad-CAM++: Generalized gradient-based visual explanations for deep convolutional networks. In *IEEE WACV*, pages 839–847, 2018. 3

[2] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *IEEE CVPR*, pages 248–255, 2009. 2

[3] Ruth Fong, Mandela Patrick, and Andrea Vedaldi. Understanding deep networks via extremal perturbations and smooth masks. In *IEEE ICCV*, pages 2950–2958, 2019. 3

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE CVPR*, pages 770–778, 2016. 1, 4, 5

[5] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. MobileNets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. 1

[6] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *IEEE CVPR*, pages 7132–7141, 2018. 1

[7] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *IEEE CVPR*, pages 4700–4708, 2017. 1

[8] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 2, 4, 5

[9] Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. *arXiv preprint arXiv:2006.15055*, 2020. 1

[10] Daniel Omeiza, Skyler Speakman, Celia Cintas, and Komminist Weldermariam. Smooth Grad-CAM++: An enhanced inference level visualization technique for deep convolutional neural network models. *arXiv preprint arXiv:1908.01224*, 2019. 3

[11] Zhongang Qi, Saeed Khorram, and Li Fuxin. Visualizing deep networks by optimizing with integrated gradients. In *AAAI*, volume 34, pages 11890–11898, 2020. 3

[12] Karl Schulz, Leon Sixt, Federico Tombari, and Tim Landgraf. Restricting the flow: Information bottlenecks for attribution. In *ICLR*, 2020. 3

[13] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-CAM: Visual explanations from deep networks via gradient-based localization. In *IEEE ICCV*, pages 618–626, 2017. 3

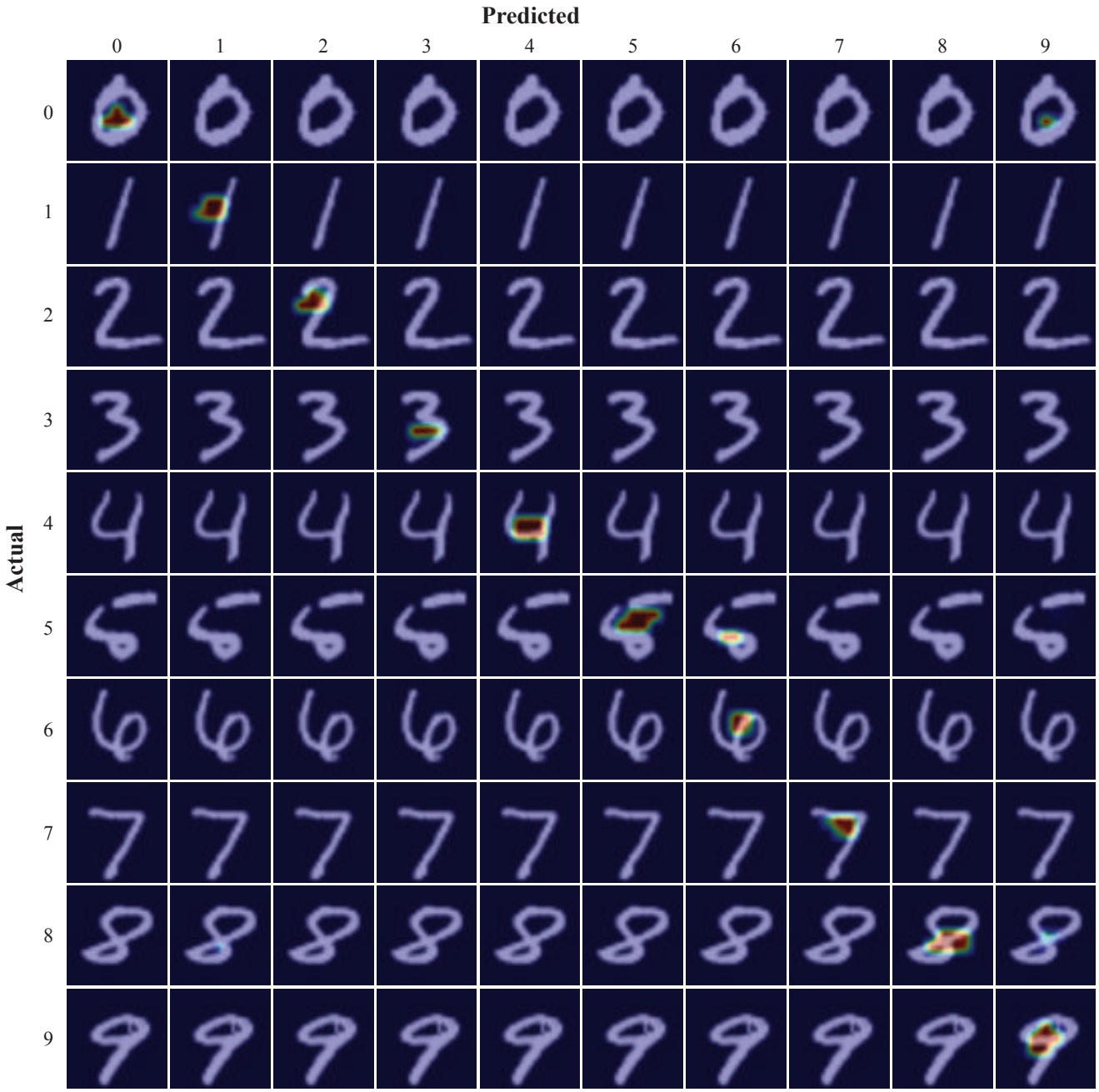
[14] Mingxing Tan and Quoc V Le. Efficientnet: Rethinking model scaling for convolutional neural networks. *arXiv preprint arXiv:1905.11946*, 2019. 1

[15] Haofan Wang, Rakshit Naidu, Joy Michael, and Soumya Snigdha Kundu. SS-CAM: Smoothed Score-CAM for sharper visual feature localization. *arXiv preprint arXiv:2006.14255*, 2020. 3

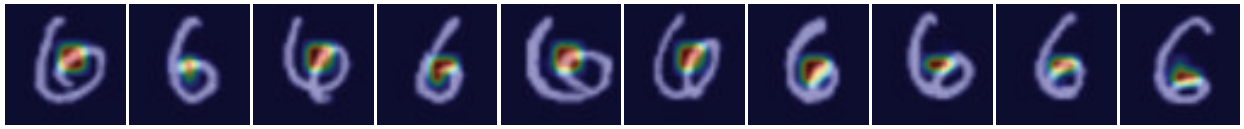
[16] Haofan Wang, Zifan Wang, Mengnan Du, Fan Yang, Zijian Zhang, Sirui Ding, Piotr Mardziel, and Xia Hu. Score-CAM: Score-weighted visual explanations for convolutional neural networks. In *IEEE CVPR Workshops*, pages 24–25, 2020. 3

[17] Hang Zhang, Chongruo Wu, Zhongyue Zhang, Yi Zhu, Zhi Zhang, Haibin Lin, Yue Sun, Tong He, Jonas Muller, R. Manmatha, Mu Li, and Alexander Smola. ResNeSt: Split-attention networks. *arXiv preprint arXiv:2004.08955*, 2020. 1, 2

[18] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *IEEE CVPR*, pages 2921–2929, 2016. 3

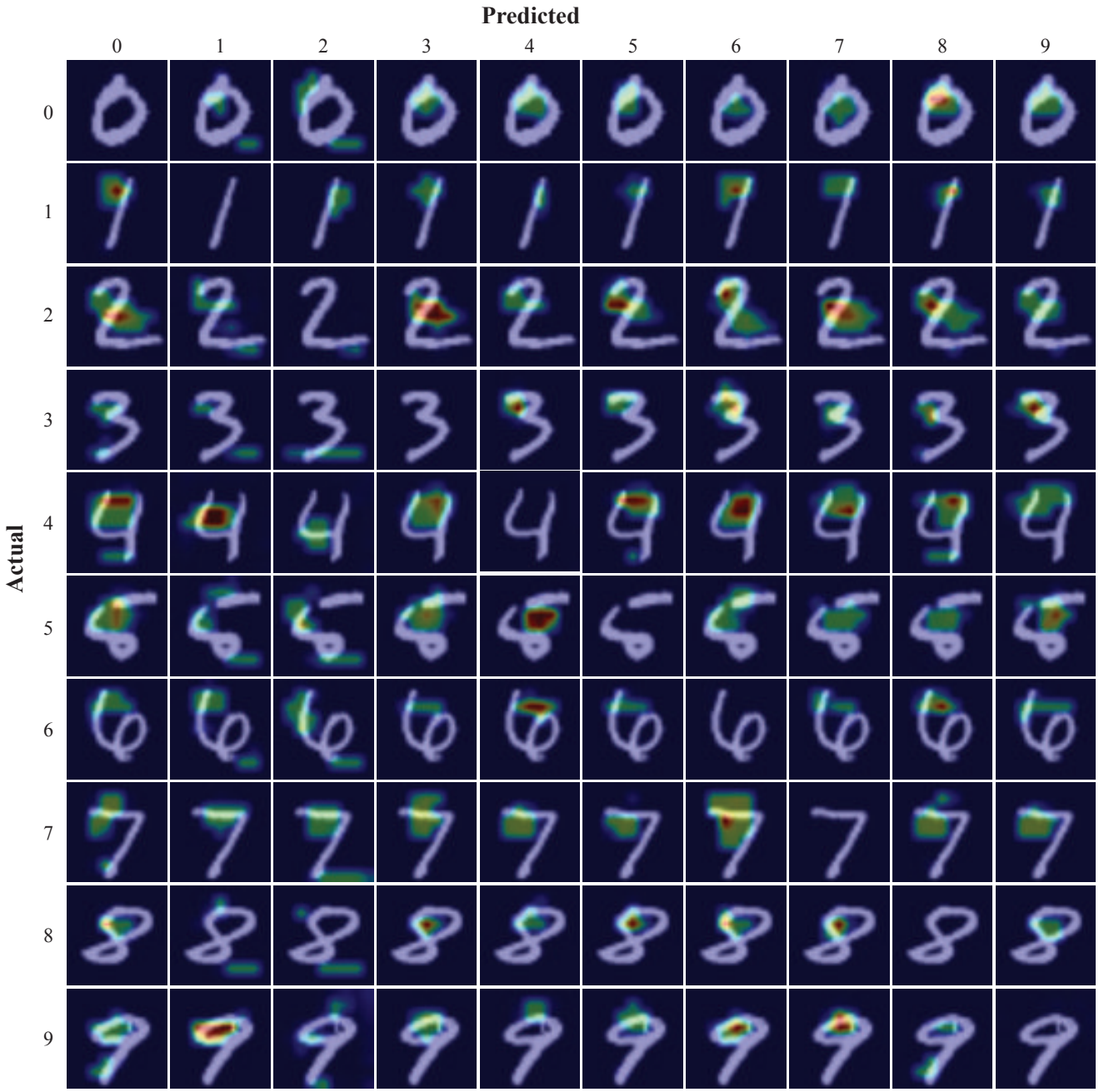


(a) **Explanation Confusion Matrix**: why SCOUTER$_+$ predicts the images of [*Actual Category*] are [*Predicted Category*]



(b) **Explanation Consistency**: why SCOUTER$_+$ predicts the images of a same category ("6") are "6"

Figure 3. Visualized positive explanations using SCOUTER$_+$ (with ResNet 18 [4] and $\lambda = 1$) on the MNIST dataset [8].



(a) **Explanation Confusion Matrix**: why SCOUTER_ not predicts the images of [*Actual Category*] are [*Predicted Category*]

(b) **Explanation Consistency**: why SCOUTER_ predicts the images of a same category ("6") are not "2"

Figure 4. Visualized negative explanations using SCOUTER_ (with ResNet 18 [4] and $\lambda = 1$) on the MNIST dataset [8].

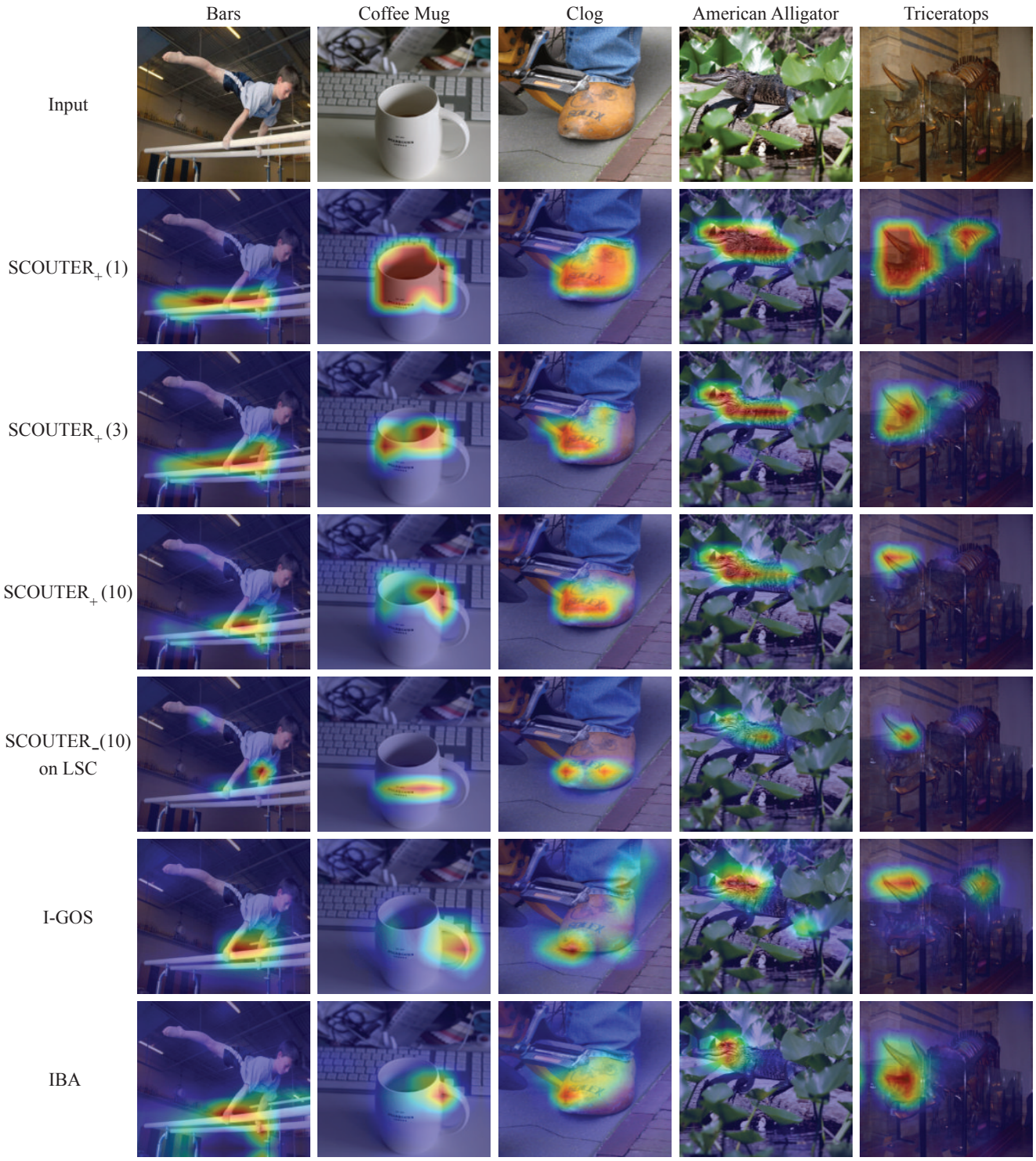


Figure 5. More examples of visualized positive explanations (Part 1). The number in parentheses represents the λ value used in the SCOUTER training.

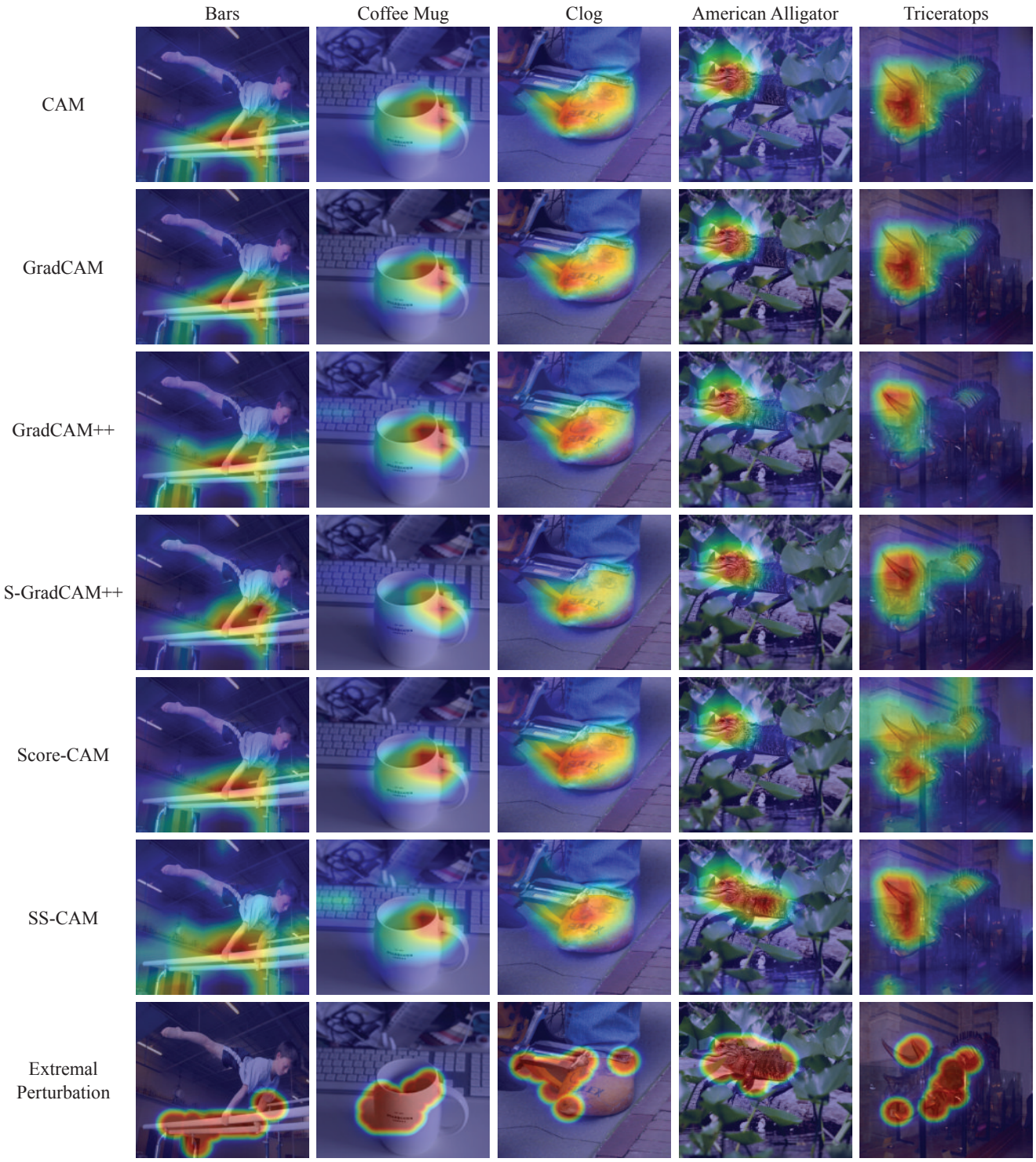


Figure 6. More examples of visualized positive explanations (Part 2). The number in parentheses represents the λ value used in the SCOUTER training.

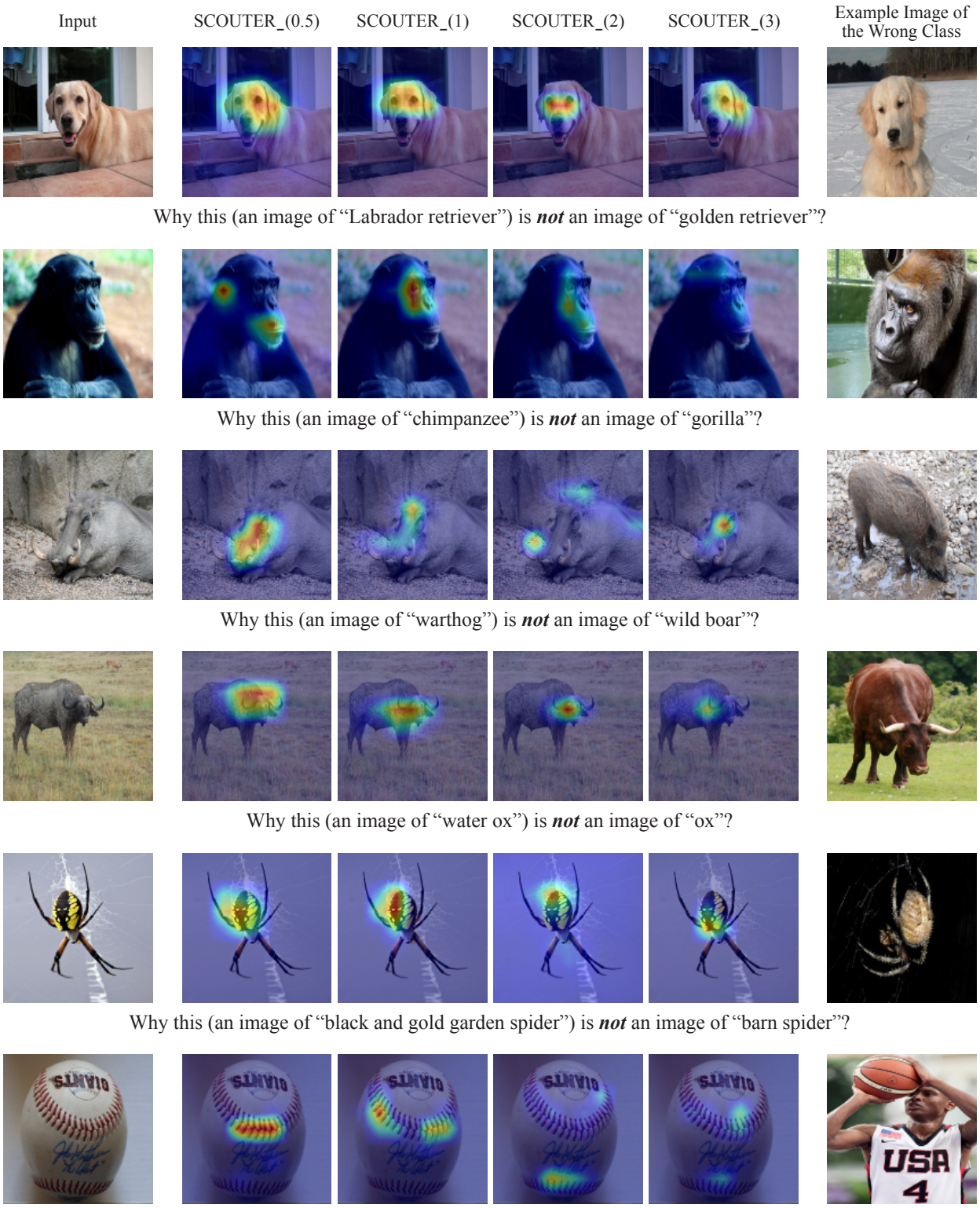


Figure 7. More examples of visualized negative explanations for similar categories. The number in parentheses represents the λ value used in the SCOUTER training.

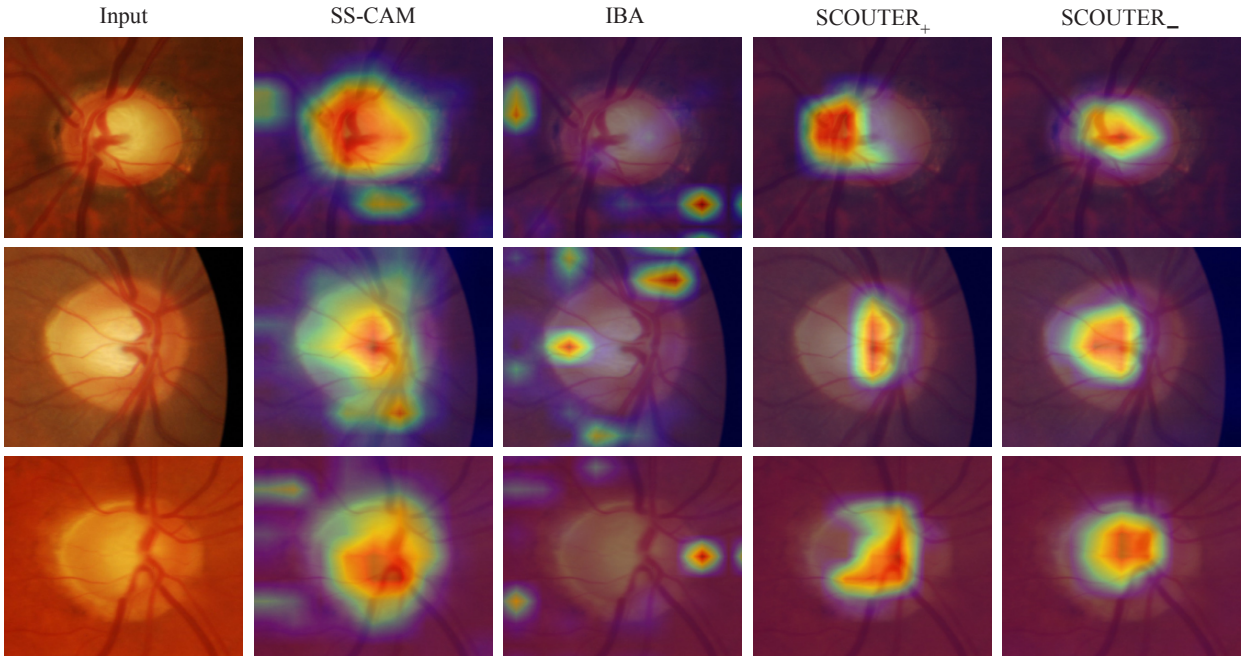


Figure 8. More examples of visualized explanations for glaucoma diagnosis on three positive samples using SS-CAM, IBA, SCOUTER$_+$, and SCOUTER$_-$. The first three methods are for "why this is `glaucoma`" while SCOUTER$_-$ is for "why this is not `normal`". SCOUTER shows explanations covering only related regions (vessel shape changes), which have been validated by two doctors.

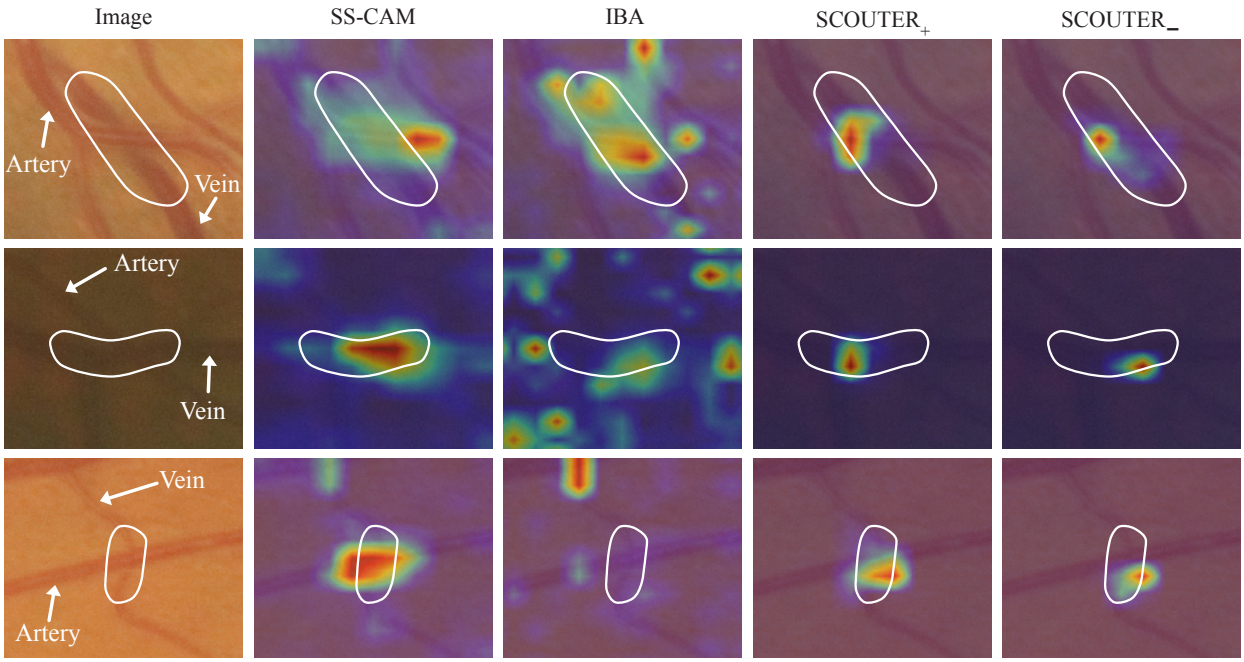


Figure 9. More examples of visualized explanations for artery hardening diagnosis on three "`moderate`" samples using SS-CAM, IBA, SCOUTER$_+$, and SCOUTER$_-$. The first three methods are for "why this is `moderate`" while SCOUTER$_-$ is for "why this is not `none`". The white circles give the approximate location of the symptoms (shape changes on the vessel wall of the vein, which are caused by the increased blood pressure in the artery). SCOUTER gives precise explanations which are mostly within the symptom region and precisely on the wall of the vein. The explanations of SS-CAM are off the target in the first row and on the wrong vessels (artery) in the first and the third rows while IBA fails to find the symptom in the second and the third rows.