SCOUTER: Slot Attention-based Classifier for Explainable Image Recognition

Liangzhi Li[1], Bowen Wang[2], Manisha Verma[1], Yuta Nakashima[1],
Ryo Kawasaki[3], Hajime Nagahara[1]
Osaka University, Japan
[1]{li, mverma, n-yuta, nagahara}@ids.osaka-u.ac.jp
[2]bowen.wang@is.ids.osaka-u.ac.jp [3]ryo.kawasaki@ophthal.med.osaka-u.ac.jp

Figure 1. Positive and negative explanations. The images from top to down are from the test sets of MNIST [22], Con-text [20], and CUB-200 [42] datasets. The models trained with positive (+) and negative (−) SCOUTER losses can respectively highlight the positive and negative supports, based on which one can reason why or why not the images are classified into the corresponding categories.

Abstract

Explainable artificial intelligence has been gaining attention in the past few years. However, most existing methods are based on gradients or intermediate features, which are not directly involved in the decision-making process of the classifier. In this paper, we propose a slot attention-based classifier called SCOUTER for transparent yet accurate classification. Two major differences from other attention-based methods include: (a) SCOUTER's explanation is involved in the final confidence for each category, offering more intuitive interpretation, and (b) all the categories have their corresponding positive or negative explanation, which tells "why the image is of a certain category" or "why the image is not of a certain category." We design a new loss tailored for SCOUTER that controls the model's behavior to switch between positive and negative explanations, as well as the size of explanatory regions. Experimental results show that SCOUTER can give better visual explanations while keeping good accuracy on small and medium-sized datasets. Code is available[4].

1. Introduction

It is of great significance to know how deep learning models make predictions, especially for the fields like medical diagnosis, where potential risks exist when black-box models are adopted. Therefore, explainable artificial intelli-

[4]https://github.com/wbw520/scouter

gence (XAI), which can give a close look into the inference process of models, has gained lots of attention.

The most popular paradigm in XAI is *attributive explanation*, which involves the contribution levels of pixels or regions to the final prediction [32, 6]. Therefore, it can be used to answer the question "why image x belongs to category t", or in short, "why x is t". Explanation is typically provided by showing a heat map that highlights the regions that support the prediction. Such visualization is of great help for understanding a model's behavior.

One natural question that arises here is *how these regions contribute to the decision*. That is, attributive explanation highlights some visual components that serve as a *positive support* for the decision; meanwhile, they may also serve as a *negative support*. The second last column of the top row in Fig. 1 shows an illustrative example of such negative supports. The presence of the horizontal line of digit "7" is a negative support when predicting the image as digit "1". Although negative supports are extremely important for various applications including computer-assisted diagnosis over medical imaging, most post-hoc approaches, such as back-propagation-based ones (*e.g.* [53, 32]) and perturbation-based ones [28, 10], basically are ignorance of the meanings of supports since their basis lies in inferring the sensitivity or relevance between the features and the decision confidences. The attention-based approach (*e.g.* [21, 43]) cannot distinguish positive/negative supports because black-box classification layers after the explanatory layer apply nonlinear transformation to the features.

In this paper, we propose a new XAI method, coined SCOUTER (Slot-based COnfigURable and Transparent classifiER). Supposing that, for each category l, there exists a *support set* $\mathcal{S}_l = \{s_{l1}, s_{l2}, \dots\}$, in which the elements are in support of the decision towards/against category l for the input image. SCOUTER is designed to find a subset $\mathcal{S}'_l \in \mathcal{S}_l$ that includes one or several supports from $\mathcal{S}_l$. The decisions by SCOUTER are solely based on the presence of $\mathcal{S} = \{\mathcal{S}'_l | l = 1, 2, \dots\}$ found in the image, without using a black-box classifier that makes $\mathcal{S}$ less interpretable. This transparency enables SCOUTER configurable to find either positive ($\mathcal{S}_+$) or negative ($\mathcal{S}_-$) supports, of which visualization can serve as positive or negative explanations.

With this new paradigm of *explainable classifiers*, smaller support regions may be preferable to facilitate the semantic interpretation of each support. That is, finding a combination of eyes, noses, lips, *etc*. may offer more explainability than directly finding a face. We thus introduce a new term in the loss function to constrain the size of support sizes. Such a constraint over the support sizes may deteriorate the classification performance by itself, but combinations of multiple supports can compensate for missing clues.

SCOUTER is built on top of the recently-emerged slot attention [23], which offers an object-centric approach for image representation. Based on this approach, we propose an explainable slot attention (xSlot) module. The output from the xSlot module is directly used as the confidence values for each category, which correspond to $\mathcal{S}$, and thus commonly used fully-connected (FC) layer-based classifiers are no longer necessary. Some visualization examples of the support regions in the xSlot module are shown in Fig. 1, with configurations of finding positive or negative supports. These examples successfully demonstrate that SCOUTER learned to find necessary supports.

The main contributions of our work include:

- We propose a transparent classifier that gives precise and meaningful explanations, which are directly involved in the decision-making process.
- We design a loss to adjust the exploratory regions for different tasks/datasets, which can better meet the requirements of various applications.
- We introduce novel concepts of positive and negative explanations, of which the latter one is a new type of explanation that can be extremely helpful when machine decisions are against users' expectations.

2. Related Work

2.1. Image Classification

For various computer vision tasks, including object detection, semantic segmentation, *etc*., most deep neural network-based models share a common structure: as the first step, a backbone network B is used to extract features F from the image x. Then a downstream network is tailored for a different task, produces a desired output from F, *e.g.* the region proposal network and pyramid pooling module. Among these tasks, image classification is one of the fundamental tasks and is used for pre-training the backbone. Some popular backbones include VGG [34], Inception [35], ResNet [14] and its variants (ResNeXt [47], ResNeSt [50]), DenseNet [19], MobileNet [17], squeeze-and-excitation network (SENet) [18], EfficientNet [36], as well as the recently introduced SpineNet [9].

Nevertheless of the importance, researchers still prefer using a simple classifier for image classification tasks, consisting of one or two FC layers and softmax. One major reason is that, despite its black-box nature, the FC classifier is the most general and expressive choice. This paper explores the possibility to use an explainable classifier in order for a transparent decision-making process while maintaining the classification performance.

2.2. Explainable AI

There are mainly three kinds of XAI methods [46], *i.e.* visualization, distillation, and intrinsic methods. **Visualization methods** usually provide explanation in the form

of heat maps, which represent the importance of the input or intermediate features. The most popular visualization methods are based on back-propagation, including CAM [53], GradCAM [32], DeepLIFT [33] and some following works [2, 26, 39, 38, 6]; or perturbation-based, including Occlusion [49], RISE [28], meaningful perturbations [11], real-time saliency [4], extremal perturbations [10], I-GOS [29], IBA [31], *etc*. One of the biggest advantages of these methods is that they produce visual explanation (*i.e*., heat maps) for each class, which is called *attributive explanation*. *Counterfactual explanation* [13] gives explanation on "how to change image x (belongs to t_a) to make it look like images in t_b."

Recently, a new kind of explanation called *discriminant explanation* [41] is proposed to answer the question "why image x belongs to t_p rather than t_i," which is similar to our proposed negative explanation. The major difference is that the discriminant explanation actually tries to identify the facts that differentiate t_i from one single class t_p, while the proposed negative explanation is designed to spot the facts to deny the target category t_i itself.

Distillation methods are built upon model distillation [16]. The basic idea is to use an inherently transparent model to mimic the output and behaviors of a trained black-box deep neural network [52, 30]. **Intrinsic methods** involve explanations as a part of their models, for example, as attention maps [21, 43, 25, 45]. Thus, this kind of XAI methods may be more desirable as they do not need to generate explanations after the decision is made [46].

2.3. Self-attention in Computer Vision

Self-attention is first introduced in the Transformers [37], in which self-attention layers scan through the input elements one by one and update them using the aggregation over the whole input. Initially, self-attention is used in place of recurrent neural networks for sequential data, *e.g*., natural language processing [7]. Recently, self-attention is adopted to the computer vision field, *e.g*., Image Transformer [27], Axial-DeepLab [40], DEtection TRansformer (DETR) [1], Image Generative Pre-trained Transformer (Image GPT) [3], *etc*. Slot attention [23] is also based on this mechanism to extract object-centric features from images (there are some other works [15, 12] using the concept of *slot*); however, the original slot attention is tested only on some synthetic image datasets. SCOUTER is based on slot attention but is designed to be an explainable classifier applicable to natural images.

3. SCOUTER

Given an image x, the objective of a classification model is to find its most probable category l in category set $\Omega = \{\omega_1, \omega_2, \ldots, \omega_n\}$. As mentioned in Section 2.1, this can be done by first extracting features $F = B(x) \in \mathbb{R}^{c \times h \times w}$ using a backbone network B. F is then mapped into a score vector $o \in \mathbb{R}^n$, representing the confidence values, using FC layers and softmax as the classifier. However, such FC classifiers can learn an arbitrary (nonlinear) transformation and thus can be black-box by themselves.

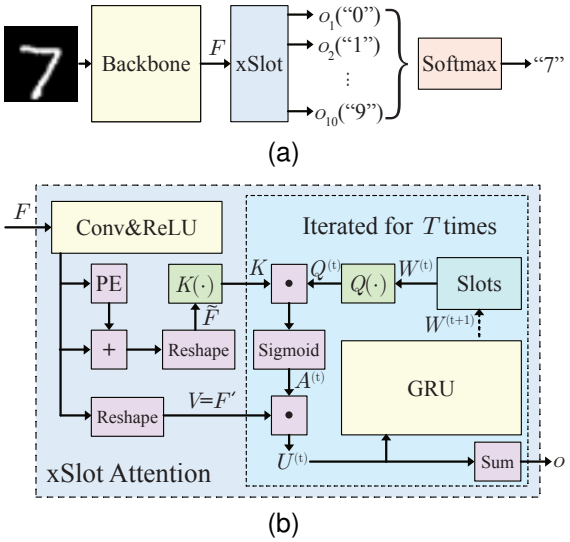


Figure 2. Classification pipeline. (a) Overview of the classification model. (b) The xSlot Attention module in SCOUTER.

We replace such an FC classifier with our SCOUTER (Fig. 2(a)), consisting of the xSlot attention module, which produces the confidence for each category given features F. The whole network, including the backbone, is trained with the *SCOUTER loss*, which provides control over the size of explanatory regions and switching between positive and negative explanations.

3.1. xSlot Attention

In the original slot attention mechanism [23], a *slot* is a representation of a local region aggregated based on the attention over the feature maps. A single slot attention module with L slots is attached on top of the backbone network B, which produces L different features as output. This configuration is handy when there are multiple objects of interest. This idea can be easily transferred to spot the supports $\mathcal{S}$ in the input image that leads to a certain decision.

The main building block of SCOUTER is the xSlot attention module, which is a variant of the slot attention module tailored for SCOUTER. Each slot of the xSlot attention module is associated with a category and gives the confidence that the input image falls into the category. With the slot attention mechanism, the slot for category l is required to find support $\mathcal{S}_l$ in the image that directly correlates to l.

Given feature F, the xSlot attention module updates slot $w_l^{(t)}$ for T times, where $w_l^{(t)}$ represents the slot after t updates and $l \in \Omega$ is the category associated to this slot. The

slot is initialized with random weights, *i.e.*,

$$w_l^{(0)} \sim \mathcal{N}(\mu, \mathrm{diag}(\sigma)) \in \mathbb{R}^{1\times c'}, \tag{1}$$

where μ and σ are the mean and variance of a Gaussian, and c' is the size of the weight vector. We denote the slots for all categories by $W^{(t)} \in \mathbb{R}^{n\times c'}$.

The slot $W^{(t+1)}$ is updated using $W^{(t)}$ and feature F. Firstly, F goes through a 1×1 convolutional layer to reduce the number of channels and the ReLU nonlinearity as $F' = \mathrm{ReLU}(\mathrm{Conv}(F)) \in \mathbb{R}_+^{c'\times d}$, with F's spatial dimensions being flattened ($d = hw$). F' is augmented by adding the position embedding to take the spatial information into account, following [37, 1], *i.e.* $\tilde{F} = F' + \mathrm{PE}$, where PE is the position embedding. We then use two multilayer perceptrons (MLPs) Q and K, each of which has three FC layers and the ReLU nonlinearity between them. This design is for giving more flexibility in the computation of *query* and *key* in the self-attention mechanism. Using

$$Q(W^{(t)}) \in \mathbb{R}^{n\times c'}, \quad K(\tilde{F}) \in \mathbb{R}^{c'\times d}, \tag{2}$$

we obtain the dot-product attention $A^{(t)}$ using sigmoid σ as

$$A^{(t)} = \sigma(Q(W^{(t)})K(\tilde{F})) \quad \in (0,1)^{n\times d}. \tag{3}$$

The attention is used to compute the weighted sum of features in the spatial dimensions by

$$U^{(t)} = A^{(t)}{F'}^{\top} \quad \in \mathbb{R}^{n\times c'}, \tag{4}$$

and slot $W^{(t)}$ is eventually updated through a gated recurrent unit (GRU) as

$$W^{(t+1)} = \mathrm{GRU}(U^{(t)}, W^{(t)}), \tag{5}$$

taking $U^{(t)}$ and $W^{(t)}$ as input and hidden state, respectively. Following the original slot attention module, we update the slot for $T = 3$ times.

The output of the xSlot attention module is the sum of all elements for category l in $U^{(T)}$, which is a function of F. Formally, the output of xSlot Attention module is:

$$\mathrm{xSlot}(F) = U^{(T)}\mathbf{1}_{c'} \quad \in \mathbb{R}_+^n, \tag{6}$$

where $\mathbf{1}$ is the column vector with all c' elements being 1.

Note that in the original slot attention module, a linear transformation is applied to the features, *i.e.*, $V(\tilde{F})$, which is then weighted using Eq. (4). However, the xSlot attention module omits this transformation as it already has a sufficient number of learnable parameters in Q, K, GRU, *etc*., and thus the flexibility. Also, the confidences, given by Eq. (6), are typically computed by an FC layer, while SCOUTER just sums up the output of xSlot attention module, which is actually the presence of learned supports for each category. This simplicity is essential for a transparent classifier as discussed in Section 3.3.

3.2. SCOUTER Loss

The whole model, including the backbone network, can be trained by simply applying softmax to $\mathrm{xSlot}(F)$ and minimizing cross-entropy loss L_{CE}. However, there is a phenomenon that, in some cases, the model prefers attending to a broad area (*e.g.*, a slot covers a combination of several supports that occupy large areas in the image) depending on the content of the image. As argued in Section 1, it can be beneficial to have control over the area of support regions to constrain the coverage of a single slot.

Therefore, we design the SCOUTER loss to limit the area of support regions. The SCOUTER loss is defined by

$$L_{\mathrm{SCOUTER}} = L_{\mathrm{CE}} + \lambda L_{\mathrm{Area}}, \tag{7}$$

where L_{Area} is the area loss, λ is a hyper-parameter to adjust the importance of the area loss. The area loss is defined by

$$L_{\mathrm{Area}} = \mathbf{1}_n^{\top} A^{(T)} \mathbf{1}_d, \tag{8}$$

which simply sums up all the elements in $A^{(T)}$. With larger λ, SCOUTER attends smaller regions by selecting fewer and smaller supports. On the contrary, it prefers a larger area with smaller λ.

3.3. Switching Positive and Negative Explanation

The model with the SCOUTER loss in Eq. (7) can only provide positive explanation since larger elements in $A^{(T)}$ means the prediction is made based on the corresponding features. We introduced a hyper-parameter $e \in \{+1, -1\}$ in Eq. (6), *i.e.*,

$$o = \mathrm{xSlot}_e(F) = e \cdot U^{(T)}\mathbf{1}_{c'} \quad \in \mathbb{R}_+^n, \tag{9}$$

where $o = \{o_1, \ldots, o_n\}$. This hyper-parameter configures the xSlot attention module to learn to find either positive or negative supports.

With the softmax cross-entropy loss, the model learns to give the largest confidence o_l corresponding to ground-truth (GT) category l and a smaller value $o_{l'}$ to wrong category $l' \neq l$. For $e = +1$, all elements given by xSlot is non-negative since both $A^{(T)}$ and F' are non-negative and thus $U^{(T)}$ is. For arbitrary non-negative F', thanks to simple reduction in Eq. (6), larger o_l can be produced only when some elements in $a_l^{(T)}$, the row vector in $A^{(T)}$ corresponding to l, is close to 1, whereas a smaller $o_{l'}$ is given when all elements in $a_l^{(T)}$ are close to 0. Therefore, by setting e to $+1$, the model learns to find the positive supports $\mathcal{S}_+$ among the images of the GT category. The visualization of $a_l^{(T)}$ thus serves as positive explanation, as in Fig. 1 (left).

On the contrary, for $e = -1$, all elements in o are negative and thus the prediction by Eq. (9) gives o_l close to 0 for correct category l and smaller $o_{l'}$ for non-GT category l'. To make o_l close to 0, all elements in $a_l^{(T)}$ must be close to

0, and a smaller $o_{l'}$ is given when $a_{l'}^{(T)}$ has some elements close to 1. For this, the model learns to find the negative supports $\mathcal{S}_-$ that do not appear in the images of the GT category. As a result, $a_{l'}^{(T)}$ can be used as negative explanation, as shown in Fig. 1 (right).

4. Experiments

4.1. Experimental Setup

We chose to use the ImageNet dataset [5] for a detailed evaluation of SCOUTER, because of the following three reasons: (i) It is commonly used in the evaluation of classification models. (ii) There are many classes with similar semantics and appearances, and the relationships among them are available in the synsets of the WordNet, which can be used to evaluate positive and negative explanations. (iii) Bounding boxes are available for foreground objects, which helps measure the accuracy of visual explanation. In experiments, we use subsets of ImageNet by extracting the first n ($0 < n \leq 1,000$) categories in the ascending order of the category IDs. Also, we present classification performance on Con-text [20] and CUB-200 [42] datasets and illustrate glaucoma diagnosis using quantitative and qualitative results on ACRIMA [8] dataset.

The size of images is 260×260. The models were trained on the training set for 20 epochs and the performance scores are computed on the validation set with the trained models after the last epoch. All the quantitative results are obtained by averaging the scores from three independent runs. Our models are implemented with PyTorch. AdamW [24] is adopted as the optimizer with the initial learning rate of 10^{-4}. The feature F extracted by the backbone network is mapped into a new feature F' with the channel number $c' = 64$. All the experiments are conducted on three local GPU servers equipped with two Intel Xeon Gold 5122 (@3.60GHz) CPUs, four NVIDIA Tesla V100-SXM2 (32GB) GPUs, and 384GB memory. Every run of training occupied one single GPU.

4.2. Explainability

Explainability is usually evaluated qualitatively or with some simple quantitative tests from machine teaching experiments [13], which are subjective and may not be so convincing [41]. Recently, a new evaluation metric using the intersection over union (IoU) between explanations and semantic masks is proposed [41]. However, this is designed for the counterfactual explanation, in which two categories are involved in one explanation. Also, our objective is to generate small yet accurate explanation, rather than having the same shape and size as the objects of interest. Therefore, the intersection between our explanation and the mask can be always small and thus IoUs are biased toward 0.

We instead evaluate the accuracy of our visual explanation by the precision that measures how much regions spotted by our explanation are covered by the objects of interest. We use bounding boxes provided in ImageNet as a proxy of the object regions and compute the percentage of the pixels located inside the bounding box over the total pixel numbers in the whole explanation. Specifically, for set I of all pixels in the input image and set D of all pixels in the bounding box, our precision is defined as $\text{Precision}_l = \frac{\sum_{i \in D} a_i^l}{\sum_{i \in I} a_i^l}$, where category $l \in \Omega$ and a_i^l is the value of pixel i in $\bar{A}_l$, which is attention map A resized to the same size as the input image by bilinear interpolation. This metric is associated with category l since SCOUTER produces different support regions for different categories. For the positive explanation, the GT category's precision must be close to 1, while the other categories' precision is not necessarily high. We compute this metric on the visualization results of the GT category for positive explanations and on the least similar class (LSC) (using Eq. 10) for SCOUTER$_-$, as LSC images usually show strong and consistent negative explanations. This precision metric actually is a generalization of the pointing game [51], which counts one *hit* when the point with the largest value on the heat map locates in the bounding box and the final score is calculated as $\frac{\#\text{Hits}}{\#\text{Hits} + \#\text{Misses}}$.

We also adopt several other metrics, *i.e.*, (i) insertion area under curve (IAUC) [28], which measures the accuracy gain of a model when gradually adding pixels according to their importance given in the explanation (heat map) to a synthesized input image; (ii) deletion area under curve (DAUC) [28], which measures the performance drop when gradually removing important pixels from the input image; (iii) infidelity [48], which measures the degree to which the explanation captures how the prediction changes in response to input perturbations; and (iv) sensitivity [48], which measures the degree to which the explanation is affected by the input perturbations. In addition, we calculate the (v) overall size of the explanation areas by $\text{Area}_l = \sum_{i \in I} a_i^l$, where a smaller value is better to pinpoint the supports to differentiate one class from the others.

We conduct the explainability experiments with the ImageNet subset with the first 100 classes. We train seven models with (1) an FC classifier, (2)–(4) SCOUTER$_+$ ($\lambda = 1, 3, 10$), and (5)–(7) SCOUTER$_-$ ($\lambda = 1, 3, 10$) using ResNeSt 26 [50] as the backbone. The results of competing methods are obtained from the FC classifier-based model.

The numerical results are shown in Table 1. We can see that SCOUTER can keep a small area size while achieving good scores in all metrics. These results demonstrate that the visualization by SCOUTER is preferable in terms of area size and precision, insensitive to noises (sensitivity), and with good explainability (infidelity, IAUC, and DAUC). In addition, SCOUTER$_-$ on LSC does not perform well on IAUC and DAUC. This is because these two metrics are

Table 1. Evaluation of the explanations on the GT class. SCOUTER$_-$ is with the least similar class (LSC).

Methods	Year	Type	Area Size	Precision	IAUC	DAUC	Infidelity	Sensitivity
CAM [53]	2016	Back-Prop	0.0835	0.7751	0.7185	0.5014	0.1037	0.1123
GradCAM [32]	2017	Back-Prop	0.0838	0.7758	0.7187	0.5015	0.1038	0.0739
DeepLIFT [33]	2017	Back-Prop	0.0874	0.7504	0.7207	0.4699	0.1042	0.0800
GradCAM++ [2]	2018	Back-Prop	0.0836	0.7861	0.7306	0.4779	0.1036	0.0807
S-GradCAM++ [26]	2019	Back-Prop	0.0868	0.7983	0.6991	0.4896	0.1548	0.0812
Score-CAM [39]	2020	Back-Prop	0.0818	0.7714	0.7213	0.5247	0.1035	0.0900
SS-CAM [38]	2020	Back-Prop	0.1062	0.7902	0.7143	0.4570	0.1109	0.1183
RISE [28]	2018	Perturbation	0.3346	0.5566	0.6913	0.4903	0.1199	0.1548
Extremal Perturbation [10]	2019	Perturbation	0.1458	0.8944	0.7121	0.5213	0.1042	0.2097
I-GOS [29]	2019	Perturbation	0.0505	0.8471	0.6838	0.3019	0.1106	0.6099
IBA [31]	2020	Perturbation	0.0609	0.8019	0.6688	0.5044	0.1039	0.0894
SCOUTER$_+$ ($\lambda = 1$)		Intrinsic	0.1561	0.8493	0.7512	0.1753	0.0799	0.0796
SCOUTER$_+$ ($\lambda = 3$)		Intrinsic	0.0723	0.8488	**0.7650**	**0.1423**	0.0949	0.0608
SCOUTER$_+$ ($\lambda = 10$)		Intrinsic	0.0476	**0.9257**	0.7647	0.2713	0.0840	0.1150
SCOUTER$_-$ (LSC, $\lambda = 1$)		Intrinsic	0.0643	0.8238	0.7343	0.1969	0.0046	**0.0567**
SCOUTER$_-$ (LSC, $\lambda = 3$)		Intrinsic	0.0545	0.8937	0.6958	0.4286	0.0196	0.1497
SCOUTER$_-$ (LSC, $\lambda = 10$)		Intrinsic	**0.0217**	0.8101	0.6730	0.7333	**0.0014**	0.1895

Table 2. Area sizes of the explanations ($\lambda = 10$).

Methods	Target Classes			
	GT	Highly-similar	Similar	Dissimilar
SCOUTER$_+$	0.0476	0.0259	0.0093	0.0039
SCOUTER$_-$	0.0097	0.0141	0.0185	0.0204

measured on GT, and the supports to deny LSC do not necessarily have the essential information to admit GT.

Among the competing methods, extremal perturbation [10], I-GOS [29], and IBA [31] also take the size of support regions into account, and thus some of them give smaller explanatory regions. Extremal perturbation's explanatory regions cover some parts of foreground objects. This leads to a high precision score, but the performance over other metrics are not satisfactory. I-GOS and IBA give small explanation areas. I-GOS results in low IAUC and sensitivity scores. IBA gets relatively low scores of IAUC and DAUC, which means its explanations cannot give correct attention to the pixels and thus have no enough explainability.

To further explore the explanation for non-GT categories, we define the semantic similarity between categories based on [44], which uses WordNet, as

$$\text{Similarity} = 2\frac{d(\text{LCS}(l, l'))}{d(l) + d(l')}, \tag{10}$$

where $d(\cdot)$ gives the depth of category l in WordNet, and $\text{LCS}(l, l')$ is to find the least common subsumer of two arbitrary categories l and l'. We define the highly-similar categories as the category pairs with a similarity score no less than 0.9, similar categories as with a score in $[0.7, 0.9)$, and the remaining categories are regarded as dissimilar categories. Table 2 summarizes the area sizes of the explanatory regions for GT, highly-similar, similar, and dissimilar categories. We see a clear trend: SCOUTER$_+$ decreases the area size when the inter-category similarity becomes lower, while SCOUTER$_-$ gives larger explanatory regions for the dissimilar categories.

Some visualization results are given in Figs. 3 and 4. It can be seen that SCOUTER gives reasonable and accurate explanations. Comparing SCOUTER$_+$'s explanation with SS-CAM [39], and IBA [31], we find that SCOUTER$_+$ can give explanations with more flexible shapes which fit the target objects better. For example, in the first row of Fig. 3, SCOUTER$_+$ gives more accurate attention around the neck. In the second row, it accurately finds the individual entities rather than covering all of them with connected regions. In addition, in the last row, SCOUTER$_+$ spots the dorsal fin, which is also an important clue to recognize sharks.

Compared with SS-CAM, IBA shows smaller explanatory regions, especially on the shark image. However, IBA is less precise and less reasonable, as is in the ostrich and chicken images, which conforms to the numerical results in Table 1. In Fig. 4, SCOUTER$_-$ can also find the negative supports, *e.g.*, the wattle of the hen, and the hammerhead and the fin of the shark. More visualization results can be found in the supplementary material.

4.3. Classification Performance

We compare SCOUTER and FC classifiers with several commonly used backbone networks with respect to the classification accuracy. The results are summarized in Fig. 5. With the increase of the category number, both the FC classifier and SCOUTER show a performance drop. They show similar trends with respect to the category number.

The relationship between λ, which controls the size of explanatory regions, and the classification accuracy is

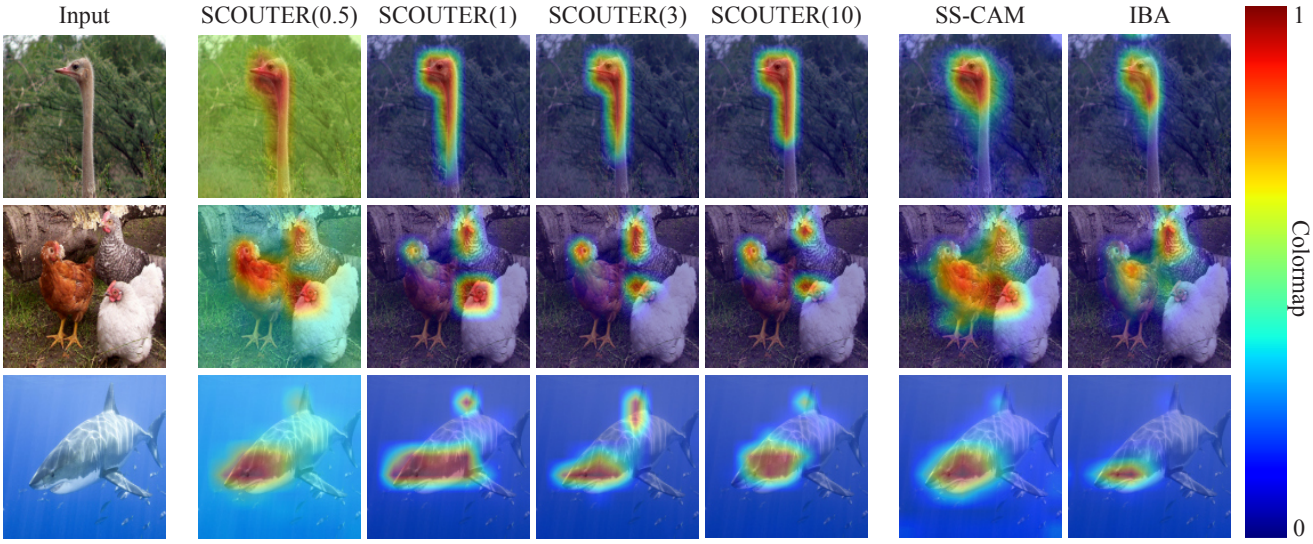
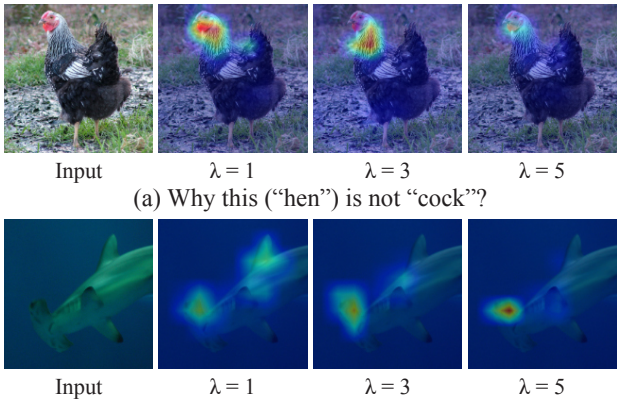


Figure 3. Visualized positive explanations using SCOUTER$_+$ and existing methods. The numbers in the parentheses are the λ values used in the SCOUTER training.



(a) Why this ("hen") is not "cock"?

(b) Why this ("hammerhead shark") is not "tiger shark"?

Figure 4. Visualized negative explanations using SCOUTER$_-$.

shown in Fig. 6 for ResNeSt 26 model with $n = 100$. A clear pattern is that the area sizes of both SCOUTER$_+$ and SCOUTER$_-$ drop quickly with the increase of λ. However, there is no significant trend in the classification accuracy, which should be because the cross-entropy loss term works well regardless of λ.

Also, according to the visualization results in Figs. 3 and 4, a larger λ does not simply decrease the explanatory regions' sizes. Instead, SCOUTER shifts its focus from some larger supports to fewer, smaller yet also decisive supports. For example, in the first row of Fig. 4, when λ is small, SCOUTER$_-$ can easily make a decision that the input image is not a cock because of its heads and unique feathers on the neck. With a larger λ, SCOUTER finds smaller combinations of supports (*i.e.*, its neck) and thus the explanation changes from the (larger) head region to the (smaller) neck region, and ultimately, to the (much smaller) wattle region.

We also summarize the classification performance of the FC classifier, SCOUTER$_+$ ($\lambda = 10$), and SCOUTER$_-$ ($\lambda = 10$) over ImageNet [5], Con-text [20], and CUB-200 [42] datasets in Table 3. The subsets with $n = 100$ are adopted for ImageNet and CUB-200, while all 30 categories are used for the Con-text. The results show that SCOUTER can be generalized to different domains and has a similar performance with the FC classifier over all datasets.

One drawback of SCOUTER is that its training is unstable when there are more than 100 categories. This is possibly because of the increasing difficulty in finding effective supports that consistently appear in all images of the same category but are not shared by other categories. This drawback limits the application of SCOUTER to small- or medium-sized datasets.

4.4. Case Study

SCOUTER uses the area loss, which constrains the size of support (or explanatory regions). This constraint can benefit some applications, including the classification of medical images, since small explanatory regions can better show the symptoms and are more informative in some cases. A typical example is the automatic glaucoma diagnosis, where doctors are eager to know the precise regions in the optic disc that lead to the machine diagnosis. We tested SCOUTER with $\lambda = 10$ over a publicly available glaucoma diagnosis dataset, *i.e.*, ACRIMA [8], which has two categories (normal and glaucoma). The dataset size is 705, in which 309 samples are normal and 396 samples are glaucoma. We split the dataset into train and test sets with a ratio of 7:3. ResNeSt 26 is used as backbone. The results are shown in Table 4. We can see that both SCOUTER$_+$ and SCOUTER$_-$ get better performances than the FC classifier.

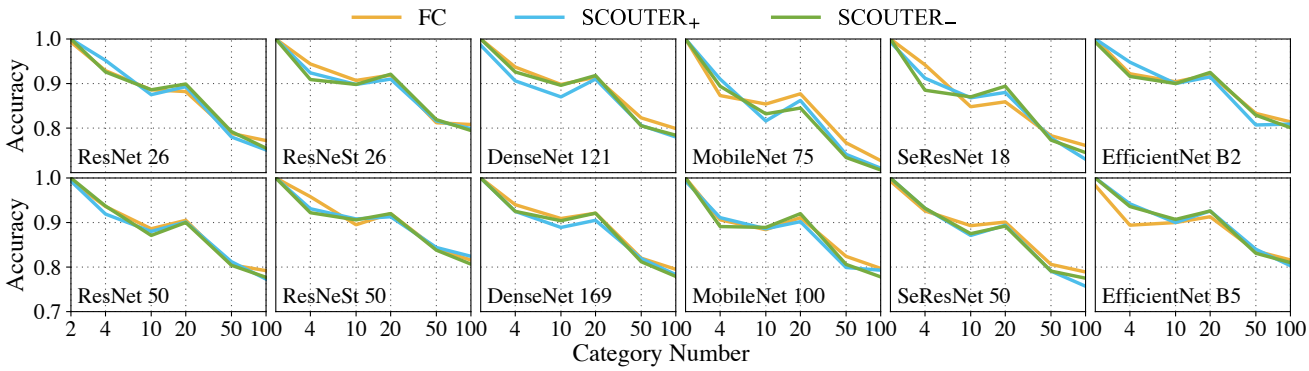


Figure 5. Classification performance of different models with FC classifier, SCOUTER$_+$ ($\lambda = 10$), and SCOUTER$_-$ ($\lambda = 10$). The horizontal axis is the category number n (in the logarithmic scale), which is used to generate the training and test set with the first n categories of ImageNet dataset; the vertical axis is the accuracy of the model.

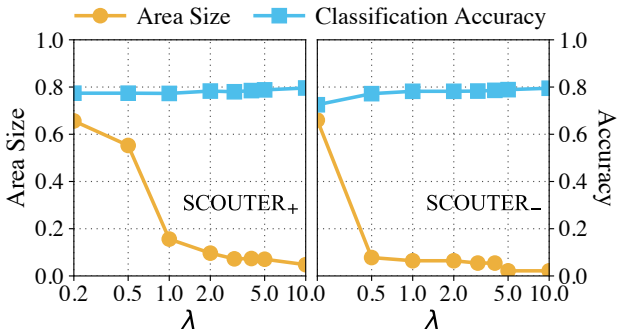


Figure 6. Relationships between λ and explanation area sizes and between λ and classification accuracy for the GT (SCOUTER$_+$, left) and LSC (SCOUTER$_-$, right) when $n = 100$. The horizontal axis is in the logarithmic scale.

Table 3. Classification accuracy on various datasets.

Models	ImageNet	Con-text	CUB-200
ResNeSt 26 (FC)	**0.8080**	0.6732	**0.7538**
ResNeSt 26 (SCOUTER$_+$)	0.7991	**0.6870**	0.7362
ResNeSt 26 (SCOUTER$_-$)	0.7946	0.6866	0.7490
ResNeSt 50 (FC)	0.8158	0.6918	**0.7739**
ResNeSt 50 (SCOUTER$_+$)	**0.8242**	**0.6943**	0.7397
ResNeSt 50 (SCOUTER$_-$)	0.8066	0.6922	0.7600
ResNeSt 101 (FC)	0.8255	0.7038	**0.7804**
ResNeSt 101 (SCOUTER$_+$)	0.8251	**0.7131**	0.7428
ResNeSt 101 (SCOUTER$_-$)	**0.8267**	0.7062	0.7643

Besides, in the visualization results in Fig. 7, SCOUTER shows much more precise and reasonable explanations that locate on some vessels in the optic disc and show clinical meanings (vessel shape change due to the enlarged optic cup), which are verified by doctors. Comparing with the other XAI methods, although IBA gives small regions, they cover some unrelated or uninformative locations.

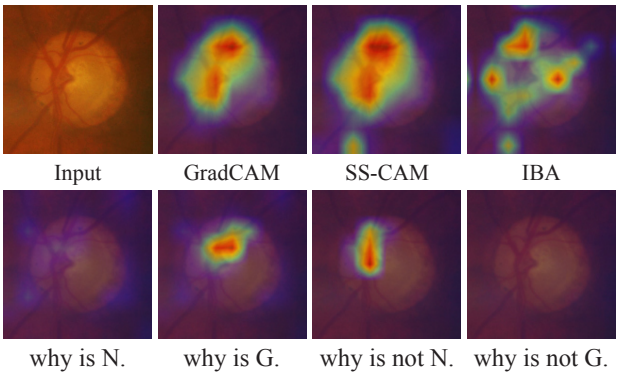


Figure 7. Explanations for a positive sample in the glaucoma diagnosis dataset. Top row: input image and explanations by existing methods. Bottom row: explanations by SCOUTER$_+$ (the first and second columns) and SCOUTER$_-$ (the third and fourth columns) for normal (N.) and glaucoma (G.).

Table 4. Classification Performance on ACRIMA Dataset [8].

Methods	AUC	Acc.	Prec.	Rec.	F1	Kappa
FC	0.9997	0.9857	0.9915	0.9831	0.9872	0.9710
SCOUTER$_+$	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
SCOUTER$_-$	0.9999	0.9952	**1.0000**	0.9915	0.9957	0.9903

5. Conclusion

A new explainable classifier is proposed in this paper. There are two variants, *i.e.*, SCOUTER$_+$ and SCOUTER$_-$, which can respectively give positive or negative explanation of the classification process. SCOUTER adopts an explainable variant of the slot attention module, namely, xSlot attention, which is also based on the self-attention. Moreover, a loss is designed to control the area size of explanatory regions. Experimental results prove that SCOUTER can give accurate explanations while keeping good classification performance.

6. Acknowledgements

This work was supported by Council for Science, Technology and Innovation (CSTI), cross-ministerial Strategic Innovation Promotion Program (SIP), "Innovative AI Hospital System" (Funding Agency: National Institute of Biomedical Innovation, Health and Nutrition (NIBIOHN)). This work was also supported by JSPS KAKENHI Grant Number 19K10662 and 20K23343.

References

[1] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. *arXiv preprint arXiv:2005.12872*, 2020. 3, 4

[2] 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, 6

[3] Mark Chen, Alec Radford, Rewon Child, Jeff Wu, Heewoo Jun, Prafulla Dhariwal, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *ICML*, 2020. 3

[4] Piotr Dabkowski and Yarin Gal. Real time image saliency for black box classifiers. In *NeurIPS*, pages 6967–6976, 2017. 3

[5] 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. 5, 7, 2

[6] Saurabh Desai and Harish G. Ramaswamy. Ablation-CAM: Visual explanations for deep convolutional network via gradient-free localization. In *IEEE WACV*, pages 972–980, 2020. 2, 3

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 3

[8] Andres Diaz-Pinto, Sandra Morales, Valery Naranjo, Thomas Köhler, Jose M Mossi, and Amparo Navea. CNNs for automatic glaucoma assessment using fundus images: an extensive validation. *Biomedical Engineering Online*, 18(1):29, 2019. 5, 7, 8

[9] Xianzhi Du, Tsung-Yi Lin, Pengchong Jin, Golnaz Ghiasi, Mingxing Tan, Yin Cui, Quoc V Le, and Xiaodan Song. SpineNet: Learning scale-permuted backbone for recognition and localization. In *IEEE CVPR*, pages 11592–11601, 2020. 2

[10] Ruth Fong, Mandela Patrick, and Andrea Vedaldi. Understanding deep networks via extremal perturbations and smooth masks. In *IEEE ICCV*, pages 2950–2958, 2019. 2, 3, 6

[11] Ruth C Fong and Andrea Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. In *IEEE ICCV*, pages 3429–3437, 2017. 3

[12] Anirudh Goyal, Alex Lamb, Phanideep Gampa, Philippe Beaudoin, Sergey Levine, Charles Blundell, Yoshua Bengio, and Michael Mozer. Object files and schemata: Factorizing declarative and procedural knowledge in dynamical systems. *arXiv preprint arXiv:2006.16225*, 2020. 3

[13] Yash Goyal, Ziyan Wu, Jan Ernst, Dhruv Batra, Devi Parikh, and Stefan Lee. Counterfactual visual explanations. In *ICML*, 2019. 3, 5

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE CVPR*, pages 770–778, 2016. 2, 1, 4, 5

[15] Mikael Henaff, Jason Weston, Arthur Szlam, Antoine Bordes, and Yann LeCun. Tracking the world state with recurrent entity networks. In *ICLR*, 2017. 3

[16] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 3

[17] 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. 2, 1

[18] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *IEEE CVPR*, pages 7132–7141, 2018. 2, 1

[19] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *IEEE CVPR*, pages 4700–4708, 2017. 2, 1

[20] Sezer Karaoglu, Ran Tao, Jan van Gemert, and Theo Gevers. Con-Text: Text detection for fine-grained object classification. *IEEE TIP*, 26(8):3965–3980, 2017. 1, 5, 7

[21] Jinkyu Kim and John Canny. Interpretable learning for self-driving cars by visualizing causal attention. In *IEEE ICCV*, pages 2942–2950, 2017. 2, 3

[22] 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. 1, 2, 4, 5

[23] 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. 2, 3, 1

[24] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2018. 5

[25] David Mascharka, Philip Tran, Ryan Soklaski, and Arjun Majumdar. Transparency by design: Closing the gap between performance and interpretability in visual reasoning. In *IEEE CVPR*, pages 4942–4950, 2018. 3

[26] 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, 6

[27] Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. Image transformer. In *ICML*, 2018. 3

[28] Vitali Petsiuk, Abir Das, and Kate Saenko. RISE: Randomized input sampling for explanation of black-box models. In *BMVC*, 2018. 2, 3, 5, 6

[29] Zhongang Qi, Saeed Khorram, and Fuxin Li. Visualizing deep networks by optimizing with integrated gradients. In *CVPR Workshops*, 2019. 3, 6

[30] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Why should I trust you? Explaining the predictions of any classifier. In *KDD*, pages 1135–1144, 2016. 3

[31] Karl Schulz, Leon Sixt, Federico Tombari, and Tim Landgraf. Restricting the flow: Information bottlenecks for attribution. In *ICLR*, 2020. 3, 6

[32] 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. 2, 3, 6

[33] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *ICML*, page 3145–3153, 2017. 3, 6

[34] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 2

[35] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *IEEE CVPR*, pages 1–9, 2015. 2

[36] Mingxing Tan and Quoc V Le. Efficientnet: Rethinking model scaling for convolutional neural networks. *arXiv preprint arXiv:1905.11946*, 2019. 2, 1

[37] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, pages 5998–6008, 2017. 3, 4

[38] 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, 6

[39] 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, 6

[40] Huiyu Wang, Yukun Zhu, Bradley Green, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. Axial-deeplab: Stand-alone axial-attention for panoptic segmentation. *arXiv preprint arXiv:2003.07853*, 2020. 3

[41] Pei Wang and Nuno Vasconcelos. SCOUT: Self-aware discriminant counterfactual explanations. In *IEEE CVPR*, pages 8981–8990, 2020. 3, 5

[42] Peter Welinder, Steve Branson, Takeshi Mita, Catherine Wah, Florian Schroff, Serge Belongie, and Pietro Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010. 1, 5, 7

[43] Zbigniew Wojna, Alex Gorban, Dar-Shyang Lee, Kevin Murphy, Qian Yu, Yeqing Li, and Julian Ibarz. Attention-based extraction of structured information from street view imagery. In *ICDAR*, pages 844–850, 2017. 2, 3

[44] Zhibiao Wu and Martha Palmer. Verbs semantics and lexical selection. In *ACL*, page 133–138, 1994. 6

[45] Ning Xie, Farley Lai, Derek Doran, and Asim Kadav. Visual entailment: A novel task for fine-grained image understanding. *arXiv preprint arXiv:1901.06706*, 2019. 3

[46] Ning Xie, Gabrielle Ras, Marcel van Gerven, and Derek Doran. Explainable deep learning: A field guide for the uninitiated. *arXiv preprint arXiv:2004.14545*, 2020. 2, 3

[47] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *IEEE CVPR*, pages 1492–1500, 2017. 2

[48] Chih-Kuan Yeh, Cheng-Yu Hsieh, Arun Suggala, David I Inouye, and Pradeep K Ravikumar. On the (in)fidelity and sensitivity of explanations. In *NeurIPS*, pages 10967–10978, 2019. 5

[49] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *ECCV*, pages 818–833, 2014. 3

[50] 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. 2, 5, 1

[51] Jianming Zhang, Sarah Adel Bargal, Zhe Lin, Jonathan Brandt, Xiaohui Shen, and Stan Sclaroff. Top-down neural attention by excitation backprop. *International Journal of Computer Vision*, 126(10):1084–1102, 2018. 5

[52] Quanshi Zhang, Yu Yang, Haotian Ma, and Ying Nian Wu. Interpreting CNNs via decision trees. In *IEEE CVPR*, pages 6261–6270, 2019. 3

[53] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *IEEE CVPR*, pages 2921–2929, 2016. 2, 3, 6

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

1. Computational Costs

The results in Table 1 show 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 increase is not very significant and it increases linearly.

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. While for $n > 90$, SCOUTER uses fewer parameters than the FC classifier because there are only c' ($c' = 64$ in our implementation) learnable parameters for each category. This is much less than the parameter size of the FC classifier, in which each category usually needs more than 512 parameters ($2,048$ parameters for ResNet 50).

Comparing to the differences in the computation costs and the numbers of parameters of different backbone models, the changes due to 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 [14]	14.1511	**14.1298**	**3.4238**	3.4565
ResNet 50 [14]	23.7129	**23.6916**	**5.9830**	6.0171
ResNeSt 26 [50]	15.2253	**15.2041**	**5.1803**	5.2130
ResNeSt 50 [50]	25.6391	**25.6179**	**7.7430**	7.7762
DenseNet 121 [19]	7.0564	**7.0719**	**3.7536**	3.7805
DenseNet 169 [19]	12.6510	**12.6435**	**4.4396**	4.4683
MobileNet 75 [17]	1.1194	**0.6537**	**0.0563**	0.0812
MobileNet 100 [17]	4.3301	**3.0859**	**0.3154**	0.3421
SeResNet 18 [18]	11.3169	**11.3509**	**2.6473**	2.6726
SeResNet 50 [18]	26.2439	**26.2226**	**5.6758**	5.7098
EfficientNet B2 [36]	7.8419	**7.8437**	**1.0250**	1.0564
EfficientNet B5 [36]	28.5457	**28.5244**	**3.6391**	3.6721

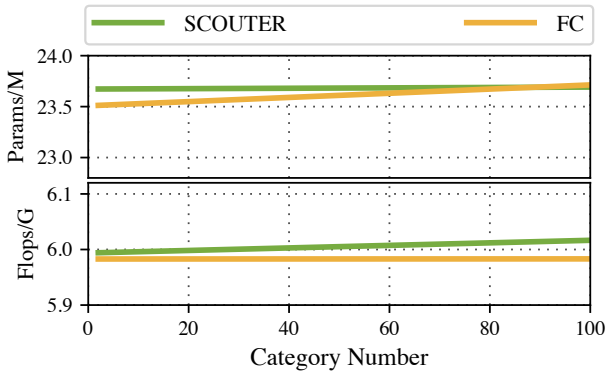


Figure 1. Flops and parameter sizes of SCOUTER and FC classifier with ResNet 50.

2. Components of xSlot Attention Module

In SCOUTER, we adopt a variant of the slot attention [23]. 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, that is used for updating slot weights, with an average operation. The second one is SCOUTER without PE, where flattened input features are directly used without adding any extra 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 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

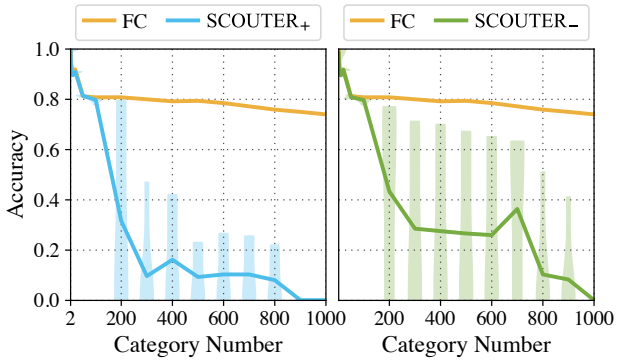


Figure 2. The classification performance of FC classifier, SCOUTER$_+$, and SCOUTER$_-$ when $2 \leq n \leq 1000$. We show the violin plots as well as the average value for SCOUTER$_+$ and SCOUTER$_-$, while the FC classifier is only with the average value.

3. Classification Performance when $n > 100$

Training of SCOUTER becomes unstable when the category number n of the ImageNet [5] 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 [50] 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, we admit that SCOUTER can only be used in small-or medium-sized datasets due to this issue.

4. Inter-and Intra-Category Explanation

To better understand what supports SCOUTER uses as the basis for its decision making, how these supports can be differentiated among different categories, and whether they are being consistent for images in the same category, we give some additional visualization on MNIST dataset [22] in Figs. 3 and 4 for SCOUTER$_+$ and SCOUTER$_-$, respectively. MNIST is adopted here as similarities and dissimilarities among categories (numerals) are widely known and easier for understanding than ImageNet classes. In these two figures, (a) is for the inter-categories 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 one, we use the number *6* as an example and the first ten samples with label *6* in the test set of MNIST are used.

In the inter-category visualization in Fig. 3, we can see that SCOUTER$_+$ successfully find 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* character 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 number recognition is a very 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 numbers.

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 crossing 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 character *2*, as well as the missing of the bottom horizon stroke, which is essential for *2*.

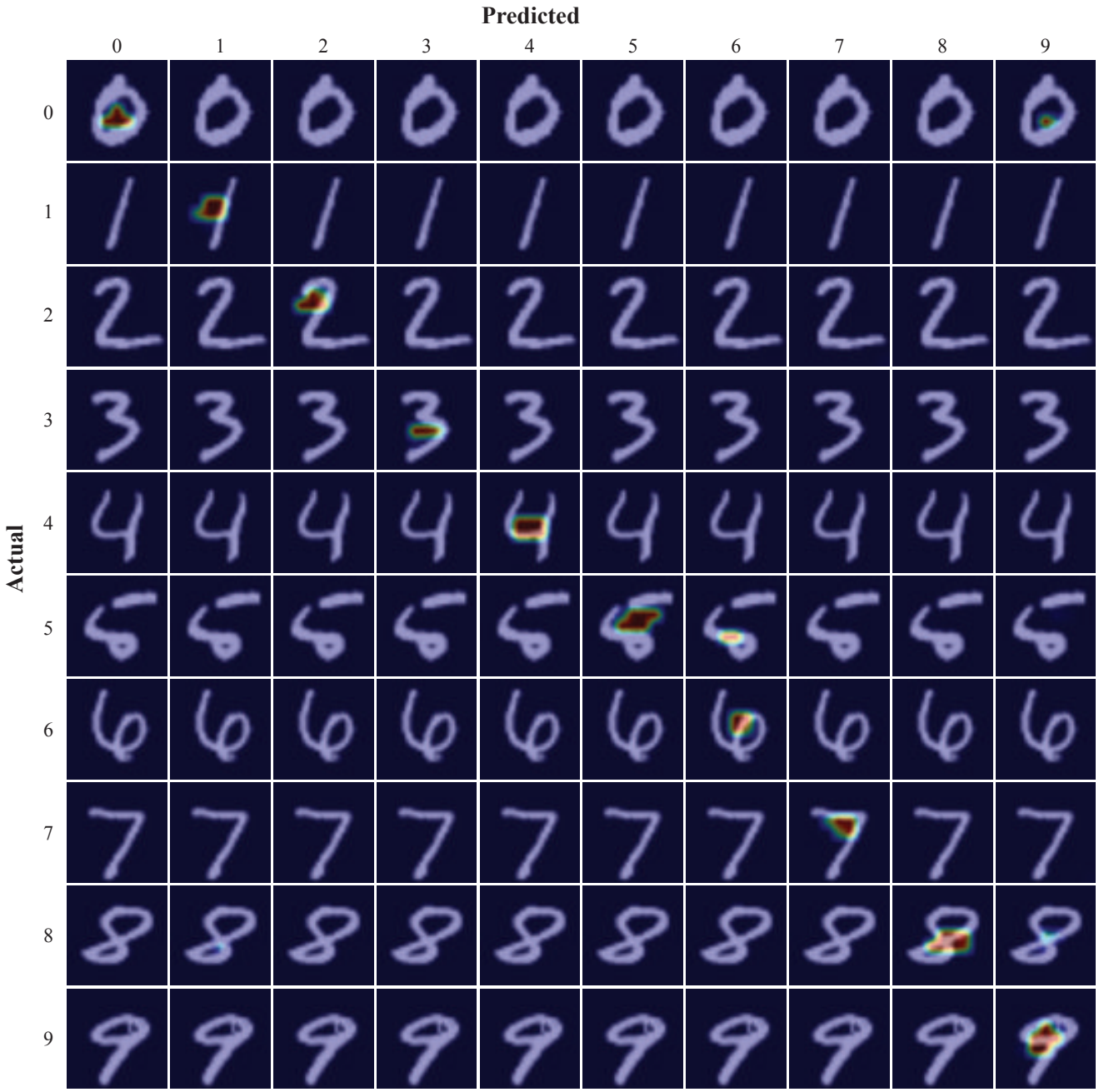
5. Some More Visualizations

In this section, we show more visualization results for ImageNet using SCOUTER and competing methods, including I-GOS [29], IBA [31], CAM [53], GradCAM [32], GradCAM++ [2], S-GradCAM++ [26], Score-CAM [39], SS-CAM [38], and Extremal Perturbation [10].

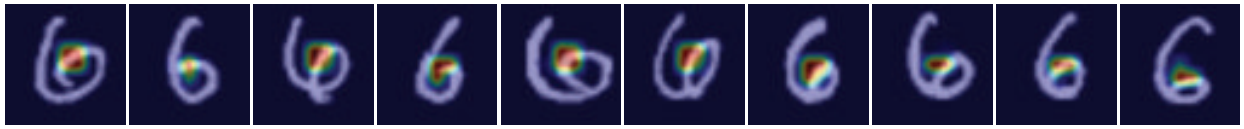
Subsets with $n = 100$ categories are used for training and visualization. Besides the first n categories (as used in the main paper), we also use several other subsets (with the same category number) in the ImageNet dataset, in order to provide visualization examples with more diversity. Figs. 5 and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

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. For example, the differences in the eyes and ears between "Labrador retriever" and "golden retriever", and the differences of the horn between "water ox" and "ox". This kind of information is very useful for certain applications.

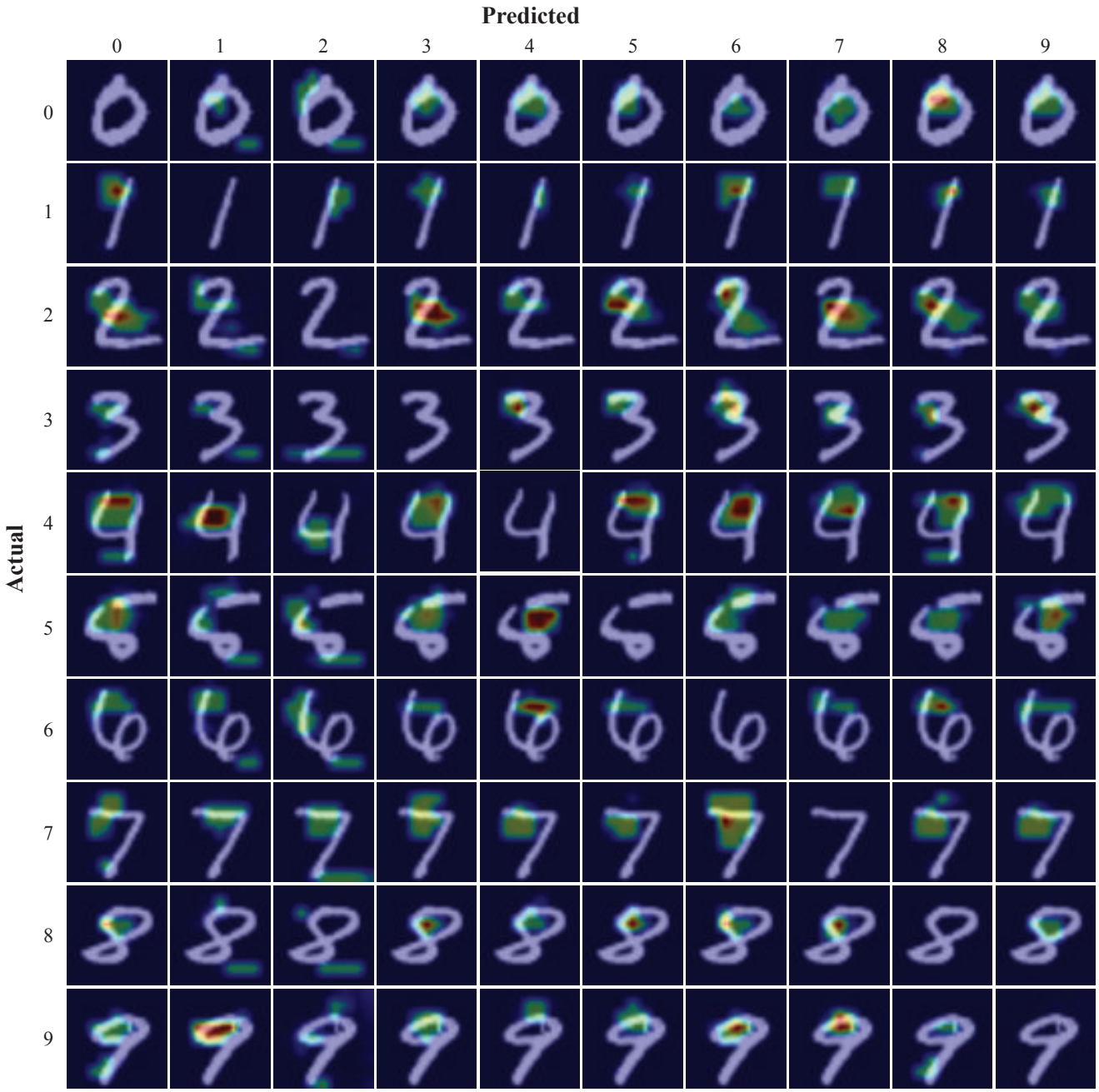


(a) **Explanation Confusion Matrix**: why SCOUTER$_+$ predicts the images of [*Actual Category*] are [*Predicted Category*]



(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 [14] and $\lambda = 1$) on the MNIST dataset [22].



(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 [14] and $\lambda = 1$) on the MNIST dataset [22].

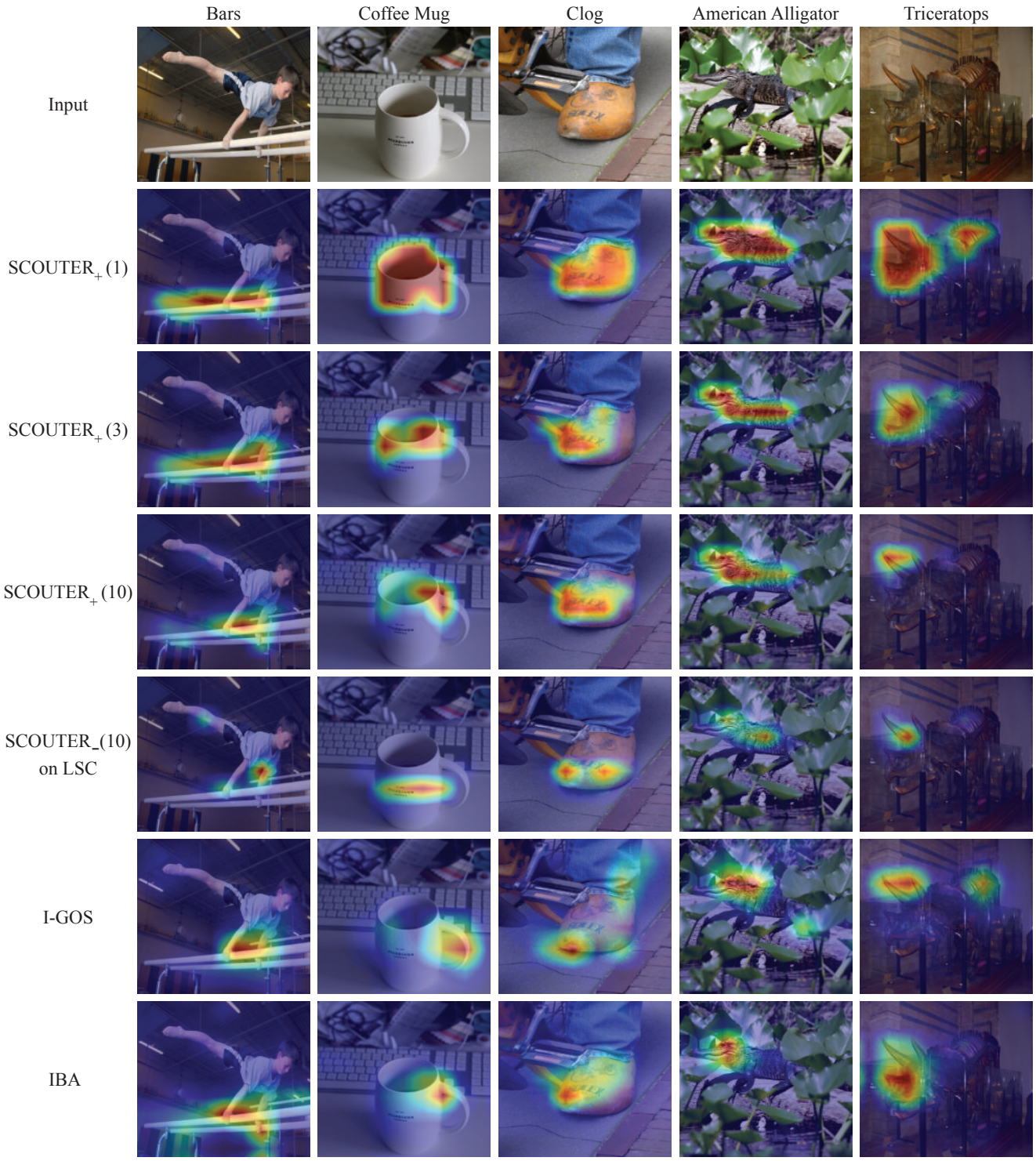


Figure 5. More examples of visualized positive explanations (Part 1). The number in parentheses represents the λ value used in the SCOUTER training.

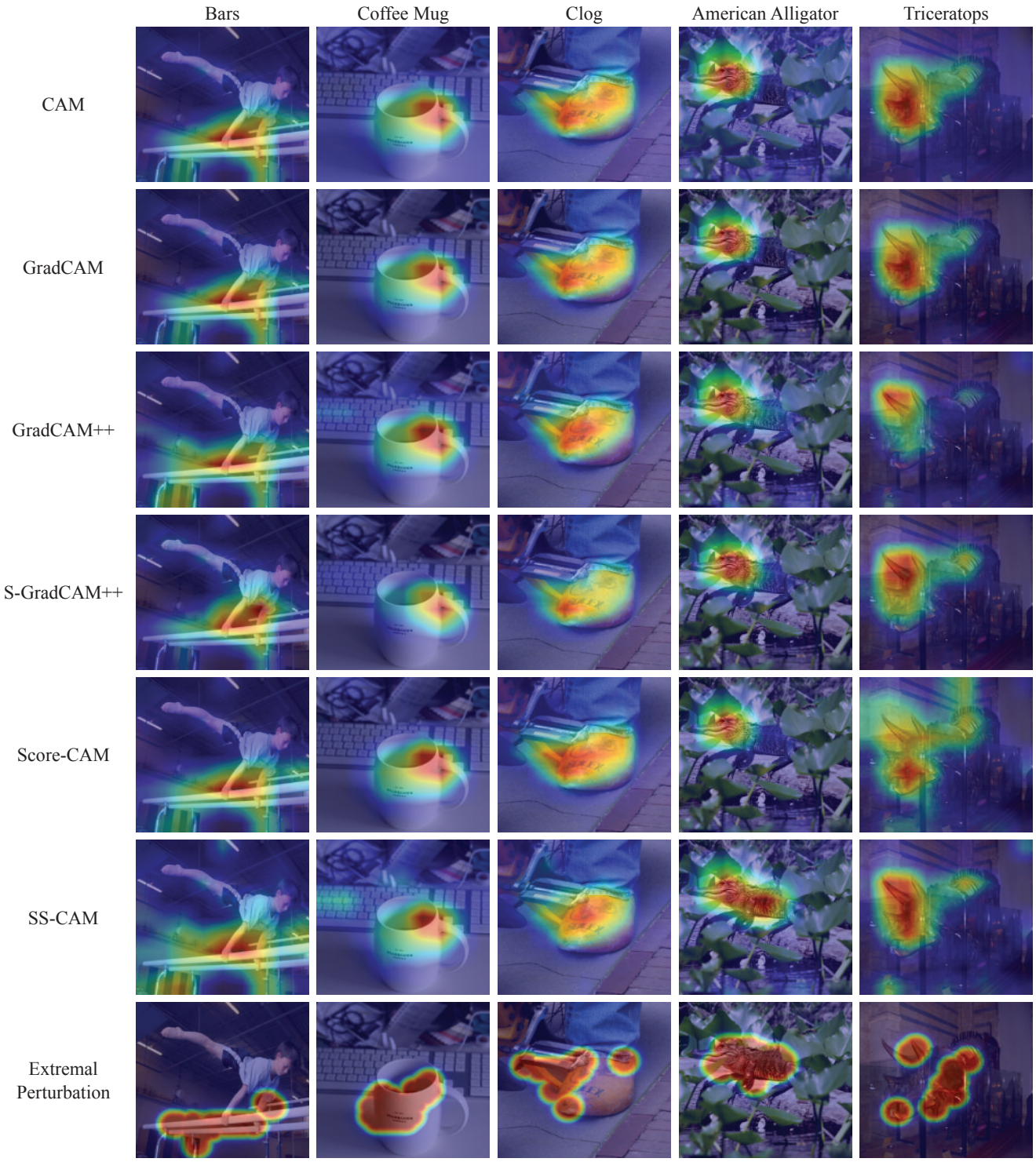


Figure 6. More examples of visualized positive explanations (Part 2). The number in parentheses represents the λ value used in the SCOUTER training.

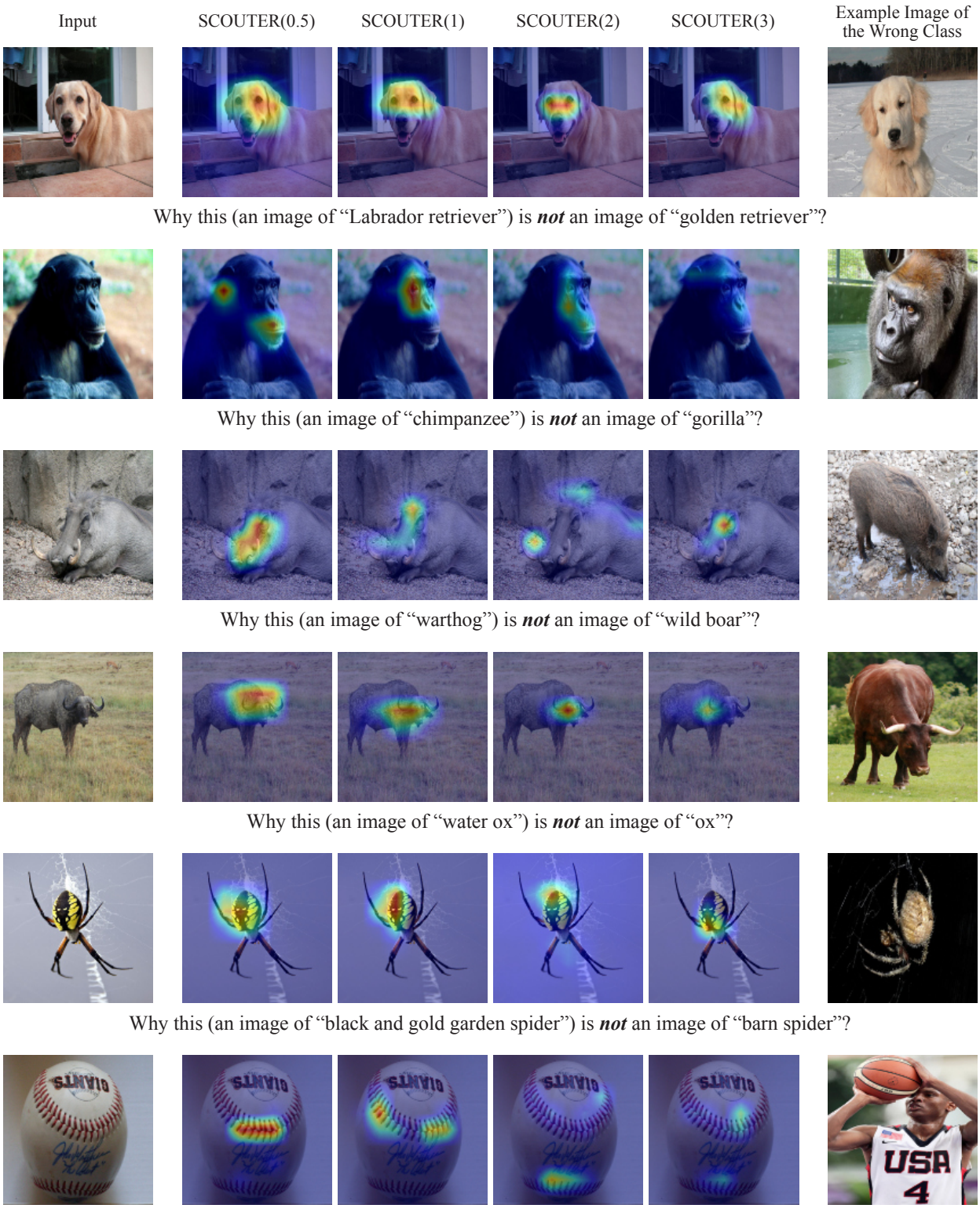


Figure 7. More examples of visualized negative explanations for similar categories. The number in parentheses represents the λ value used in the SCOUTER training.