
Reproducibility Study - SCOUTER: Slot Attention-based Classifier for Explainable Image Recognition

Anonymous Author(s)

Affiliation

Address

email

Reproducibility Summary

1

2 **Scope of Reproducibility**

3 We aim to replicate the main findings of the paper *SCOUTER: Slot Attention-based Classifier for Explainable Image*
4 *Recognition* by Li et al. in order to verify the main claims they make: 1) The explanations generated by SCOUTER
5 outperform those by other explanation methods in several explanation evaluation metrics. 2) SCOUTER achieves
6 similar classification accuracy as a fully connected model. 3) SCOUTER achieves higher confusion matrix metrics than
7 a fully connected model on a binary classification problem.

8 **Methodology**

9 The authors provided code for training the models. We implemented the explanation evaluation metrics and confusion
10 matrix metrics ourselves. We used the same hyperparameters as the original work, in case the hyperparameter was
11 reported. We trained all models from scratch on various datasets and evaluated the explanations generated by these
12 models with all reported metrics. We compared the accuracy scores between different models on several datasets.
13 Finally, we calculated an assortment of confusion matrix metrics on models trained on a binary dataset.

14 **Results**

15 We were only able to reproduce 22.2% of the explanation evaluation metrics and could thus not find conclusive support
16 for claim 1. We could only verify claim 2 for one of the datasets and in total could reproduce 55.5% of the original
17 scores. We could reproduce all scores regarding claim 3, but the claim is still not justified, as the scores between the
18 fully connected and SCOUTER models lie very close to one another.

19 **What was easy**

20 The paper was well written, so understanding the SCOUTER architecture was straightforward. The code for training
21 a model was available and together with the examples the authors provide, this was achievable with relative ease. A
22 checkpoint system is implemented, so training a model can be split into multiple runs. All used datasets are available
23 and straightforward to obtain.

24 **What was difficult**

25 The original code did not contain any documentation, which made it difficult to navigate. No code for calculating
26 the metrics was provided and this had to be implemented from scratch. During the training of the models, memory
27 allocation issues occurred. Training and evaluating on a large dataset took a considerable amount of time.

28 **Communication with original authors**

29 We sent the authors an e-mail to request either the missing code or more details on how the metrics were implemented,
30 but unfortunately we did not receive a reply.

31 **1 Introduction**

32 Explainable Artificial Intelligence (XAI) is growing in popularity and becomes increasingly important as more and more
33 AI applications are used in daily life. It is important to visualize both positive and negative patterns in the explanation
34 of a model [2], but this discernment has not gained much attention yet. In [8], Li et al. introduce a model architecture
35 that is capable of generating both positive and negative explanations based on an explainable slot attention module.

36 **2 Scope of reproducibility**

37 The authors sought to tackle the problem of deep neural networks being unintelligible. For this purpose they developed
38 SCOUTER (Slot-based COnfigURable and Transparent classifier) [8]. The unique aspect of SCOUTER is that every
39 category has its corresponding positive or negative explanation as to why a particular image does or does not belong to
40 a certain category. This offers a more in-depth look into what a model bases its predictions on and thus increases its
41 explainability.

42 The main claim of the original paper is aptly summarised in the last sentence of its conclusion: "*Experimental results*
43 *prove that SCOUTER can give accurate explanations while keeping good classification performance*". This is what we
44 will be trying to reproduce. While this claim in itself is vague, the authors compare the score SCOUTER achieves on
45 certain datasets to other methods such as GradCAM [6], RISE [11], I-GOS [12] and IBA [13] and show that SCOUTER
46 achieves a similar or higher score in most metrics. Furthermore, they also train a model where the slot attention is
47 replaced with a fully connected layer as an (unexplainable) baseline to compare SCOUTER to. This can be dissected
48 into the three following claims that we will attempt to verify by reproducing the experiments of the authors:

- 49 1. SCOUTER will achieve the highest score on the following explanation evaluation metrics: area size, precision,
50 insertion area under curve, deletion area under curve, infidelity and sensitivity on the ImageNet dataset [3]
51 compared to other explanation methods.
- 52 2. SCOUTER will achieve similar classification accuracy as the FC model trained and validated on the ImageNet,
53 Con-text [7], and CUB-200-2011 [14] datasets
- 54 3. SCOUTER will achieve higher ROC-AUC, Accuracy, Precision, Recall, F1-Score and Cohen's Kappa scores
55 than the FC model on the ACRIMA dataset [4].

56 **3 Methodology**

57 The original paper provides a link to the Github repository¹ with the code and instructions necessary to train the models
58 which were reported in the paper. However, the code used to evaluate the explanations of the trained models was not
59 included. We therefore had to implement these ourselves. The area size metric was partially implemented in the
60 authors code, where the area size for a single image was calculated. We extended this code to calculate the average
61 area size over the entire validation set. We implemented the following explanation evaluation metrics from various
62 papers ourselves: precision [8], Insertion Area Under Curve (IAUC) [11], Deletion Area Under Curve (DAUC) [11],
63 infidelity [16] and sensitivity [16]. Interesting to note is that the precision metric is defined by the authors themselves.
64 The papers the authors referenced for IAUC and DAUC², and infidelity and sensitivity³ provided code for the metrics
65 implementation. We used these and adapted them slightly to integrate it with the code for SCOUTER to deviate as little
66 as possible from the original experiments.

67 Furthermore, there was no code available for working with the ACRIMA dataset, so we implemented this ourselves as
68 well.

69 Using the code of the authors composited with our own code, we conducted our experiments on the GPU nodes of the
70 LISA cluster on SurfSara⁴ which uses an Nvidia GeForce 1080Ti, 11GB GDDR5X GPU.

71

¹<https://github.com/wbw520/scouter>

²<https://github.com/eclique/RISE>

³https://github.com/chihkuanyeh/saliency_evaluation

⁴<https://userinfo.surfsara.nl/systems/lisa/description>

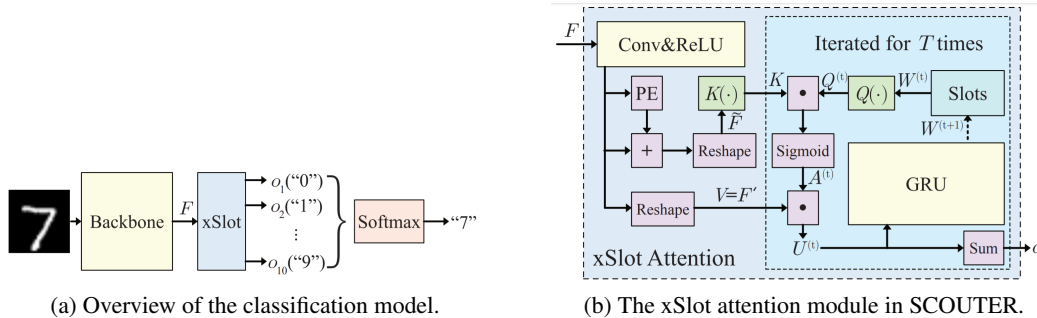


Figure 1: Overview of the SCOUTER model. Taken from Figure 2 in [8].

72 3.1 Model description

73 Typically a classification model consists of the following: feature extraction using a backbone network, which is then
 74 mapped onto a score vector representing the confidence for each class. When using fully connected layers to map such
 75 a feature onto a score vector it results in a model that is a black-box, which does not give much information about how
 76 or why a certain class attains a higher confidence score.

77 Such a fully connected classifier is replaced instead by an explainable xSlot module, which is based on an object-centric
 78 slot attention module [9]. This creates the SCOUTER model as seen in Figure 1 [8].

79 In order to reproduce the experiments we will train a multitude of such SCOUTER models. All models use ResNeSt-26
 80 [17] as their backbone, since that was also used in the experiments we aim to reproduce. All models use the same
 81 SCOUTER loss as defined by the original authors. Since there are no pre-trained models made available we will train
 82 all models from scratch. For all models the amount of parameters is just above 15,000,000. The full table for parameter
 83 counts can be found in Table 1.

Dataset	Fully Connected	SCOUTER
ImageNet	15,225,348	15,199,584
Con-text	15,081,918	15,195,104
CUB-200-2011	15,081,918	15,199,584
ACRIMA	15,024,546	15,193,312

Table 1: The number of parameters for various models.

84 3.2 Datasets

85 We used various datasets during our experiments. For reproducing the original experiments we used the ImageNet,
 86 Con-text, CUB-200-2011 and ACRIMA datasets. In line with the original experiments, we use the train part of the
 87 dataset for the training and the validation part for calculating the metrics.

88 **ImageNet.** The ImageNet Large Scale Visual Recognition Challenge (ILSVRC) dataset⁵ [3] is widely used for
 89 classification models. The categories consist of and are organized according to nouns in the WordNet hierarchy [5]. It
 90 contains 1,000 categories, 1,281,167 images for training, 50,000 images for validation and 100,000 images for testing.
 91 We preprocessed the structure of the directories of the validation set to be in line with the author’s code.

92 **Con-text.** The Con-text dataset⁶ [7] is focused on the use of fine-grained classification of buildings into their sub-classes
 93 such as cafe, tavern, diner, etc. by detecting scene text in images. The dataset consists of 28 categories with 24,255
 94 images in total. Splitting the dataset was done by the authors using a seed, as there is no inherent split.

95 **CUB-200-2011.** The Caltech-UCSD Birds 200-2011 (CUB-200-2011) dataset⁷ [14] consists of images with photos
 96 of 200 bird species (mostly North American). It consists of 200 categories with 11,788 images in total. The train set
 97 contains 5,994 images and the test set contains 5,794 images. Note that while the original authors cite the CUB-200

⁵Download link: <https://image-net.org/download.php>

⁶Download link: <https://staff.fnwi.uva.nl/s.karaoglu/datasetWeb/Dataset.html>

⁷Download link: <http://www.vision.caltech.edu/visipedia/CUB-200-2011.html>

98 dataset [15], everything in the available code points towards the authors using the CUB-200-2011 dataset. For example:
 99 the code to load the "CUB-200" data is only functional when using the CUB-200-2011 dataset. As such, we made the
 100 decision to use CUB-200-2011 for our experiments.

101 **ACRIMA.** The ACRIMA dataset⁸ [4] can be used for automatic glaucoma assessment using fundus images. It contains
 102 2 categories and 705 images. It is composed of 396 glaucomatous images and 309 normal images. There is no inherent
 103 split for the data, so we made our own with 80% of the images in the train set and 20% in the validation set using a seed.

104 3.3 Hyperparameters

105 Many of the hyper-parameters were set in accordance with the original paper, as these were documented and reported.
 106 However, not all hyperparameter settings were documented. There is a specific lack of the "slots per class" hyperpa-
 107 rameter. We tested both the positive and negative SCOUTER model with four different slots per class hyperparameter
 108 settings, namely: 1, 2, 3, and 5. We tested this with λ values of both 1 and 10. We found there to be no significant
 109 difference in performance in classification accuracy or evaluation metrics between the different slots per class values.
 110 As such, all the models that we report on were trained with 1 slot per class since this value was set by the authors in the
 111 examples they provided with their code. The full hyperparameter settings can found in Table 2.

Hyperparameter	Value	Hyperparameter	Value
Epochs	20	Lambda Value	{1, 3, 10}
Batch Size	70	Slots per Class	1
Number of Classes	$\min(N_{classes}, 100)$	Power of Slot Loss	2
Learning Rate	0.0001	Image Size	260
Learning Rate Drop	70	Channel	2048
Hidden Dimensions	64	Number of Freeze Layers	0
Hidden Layers	3	Number of Workers	4
Weight Decay	0.0001	World Size	1

Table 2: Hyperparameter settings used for the experiments.

112 3.4 Experimental setup and code

113 Our code is available at: https://anonymous.4open.science/r/reproduce_SCOUTER-030C.

114 We trained the models using the hyperparameter setup as described above. In order to reproduce the original results we
 115 trained six different models for the explanation evaluation: both SCOUTER₊ and SCOUTER₋ models with λ values
 116 of 1, 3 and 10.

117 Wherever classification accuracy is reported it is the accuracy of the model on the validation set after the final epoch, as
 118 was done by the original authors. For the evaluation of classification on ImageNet we reused the positive and negative
 119 SCOUTER models with a λ value of 10, since this model has the same hyperparameter settings. We trained separate
 120 positive and negative SCOUTER models with $\lambda = 10$ for the Con-text and CUB-200-2011 classification evaluation.
 121 For all datasets we also trained a fully connected classifier model (with ResNeSt-26 as backbone) to compare the
 122 SCOUTER models to.

123 To reproduce the confusion matrix metrics for ACRIMA we trained a positive and negative SCOUTER model on that
 124 dataset with $\lambda = 10$.

125 Results relating to Con-text, CUB-200-2011 and ACRIMA were obtained by averaging the scores from three independent
 126 runs. Due to restricted GPU hours and time constraints, we only trained a single model for each configuration on
 127 ImageNet.

128 3.4.1 Metrics

129 We used several metrics to evaluate the generated explanations. The following metrics were calculated on the ground
 130 truth class for SCOUTER₊ and on the least similar class for SCOUTER₋. The least similar class was determined via
 131 Wu-Palmer similarity of the WordNet synsets of the categories as implemented in NLTK [1]. This follows the same
 132 formula the original authors used to measure similarity.

133 **Area size** measures the average size of the generated explanations. This is calculated by summing all the pixel values in

⁸Download link: <https://figshare.com/s/c2d31f850af14c5b5232>

134 the attention map.
135 **Precision** measures the relative amount of pixels of the attention map that falls within the image’s bounding box. Some
136 images in the ImageNet dataset have multiple bounding boxes. We chose to calculate the precision as the max value of
137 each bounding box in the image.
138 **IAUC** measures the increase in accuracy under the gradual addition of pixels based on their importance in the
139 explanation. The starting state was the image after applying a Gaussian filter of size 11 and $\sigma = 5$.
140 **DAUC** measures the decrease in accuracy under the gradual removal of pixels based on their importance in the
141 explanation. The final state was an image consisting of only zeroes.
142 **Infidelity** measures how well the explanation captures the change in the model’s prediction under input perturbations.
143 The image was perturbed by adding noise sampled from a unit Gaussian. This metric was calculated over the first 50
144 images in the validation set.
145 **Sensitivity** measures how much the explanation is affected by input perturbations. We calculated the maximum
146 sensitivity, as was done in [16]. The image was perturbed by adding noise sampled from a uniform distribution ranging
147 from -0.2 to 0.2. This metric was calculated over the first 50 images in the validation set.
148 The authors do not give a complete description of how they implemented these metrics. We thus tried to stay as
149 close as possible to the implementations in [11] and [16]. All parameters were thus chosen in accordance with these
150 implementations.
151 Classification performance was mostly measured via accuracy. The performance of the models on the ACRIMA dataset
152 was evaluated more extensively with several confusion matrix metrics: ROC-AUC, accuracy, precision, recall, F1-score
153 and Cohen’s Kappa as implemented in Scikit-learn [10].

154 **3.5 Computational requirements**

155 All experiments were conducted on the the GPU nodes of the LISA cluster on SurfSara using a Nvidia GeForce 1080Ti,
156 11GB GDDR5X.
157 Computation time varied greatly between datasets. Training a model on the ImageNet dataset took up to 12 hours, for
158 CUB-200-2011 it took around 2 hours, for Con-text it was 1.5 hours and training on the ACRIMA dataset took less
159 than 5 minutes. The calculation of the explanation evaluation metrics was done on the ImageNet validation set and thus
160 took a long time as well, with IAUC and DAUC taking the longest at 1.5 hours per model.

161 **4 Results**

162 We chose to classify results that fall within ± 0.05 of the original results as reproducible. Regarding the explanation
163 evaluation metrics, we found that we could not reproduce most results reported in the original paper. The results we
164 acquired do not fully support claim 1. We were able to obtain similar classification accuracy scores on the ImageNet
165 dataset for all models, but we could not reproduce the scores for SCOUTER₊ and SCOUTER₋ on the Con-text and
166 CUB-200-2011 datasets. Therefore we cannot verify claim 2 with these results. Finally, we were able to reproduce all
167 scores from the confusion matrix metrics on the ACRIMA dataset. While we were able to reproduce the scores, we
168 cannot completely verify claim 3.

169 **4.1 Results reproducing original paper**

170 **4.1.1 Result 1: reproducing evaluation metric scores**

171 The results of our experiments regarding verifying claim 1 can be seen in Table 3. From this we can see that the area
172 size metric is largely reproducible, but the other metrics are not. Precision deviates not too much from the original
173 scores, but IAUC, DAUC, infidelity and sensitivity differ a lot. Compared to the original scores obtained for the other
174 explanation methods, SCOUTER does not outperform them with our acquired scores. Thus, we were not able to verify
175 claim 1 with our implementation.

176 **4.1.2 Result 2: Reproducing Classification Accuracy**

177 The results of our experiments regarding the verification of claim 2 can be seen in Table 4. As we can see, we were
178 able to reproduce all scores for the models trained on ImageNet. We could also recreate the accuracy scores for the FC
179 model on the other datasets. However, we did not obtain similar scores for any of the SCOUTER models on Con-text

Model	Area Size		Precision		IAUC		DAUC		Infidelity		Sensitivity	
	original	reproduced	original	reproduced	original	reproduced	original	reproduced	original	reproduced	original	reproduced
Scouter ₊ ($\lambda = 1$)	0.1561	0.1564	0.8493	0.7898	0.7512	0.3377	0.1753	0.4013	0.0799	0.0006	0.0796	1.9167
Scouter ₊ ($\lambda = 3$)	0.0723	0.1545	0.8488	0.7949	0.7650	0.3564	0.1423	0.4641	0.0949	0.0001	0.0608	1.5672
Scouter ₊ ($\lambda = 10$)	0.0476	0.1448	0.9257	0.7870	0.7647	0.3466	0.2713	0.4203	0.0840	0.0601	0.1150	2.2629
Scouter ₋ ($\lambda = 1$)	0.0643	0.0946	0.8238	0.8481	0.7343	0.2446	0.1969	0.4845	0.0046	0.0012	0.0567	2.2735
Scouter ₋ ($\lambda = 3$)	0.0545	0.0804	0.8937	0.6686	0.6958	0.3488	0.4286	0.3555	0.0196	0.0961	0.1497	2.9514
Scouter ₋ ($\lambda = 10$)	0.0217	0.0364	0.8101	0.8968	0.6730	0.2148	0.7333	0.4783	0.0014	0.0028	0.1895	3.0524

Table 3: Explanation evaluation metrics for all different SCOUTER models trained on ImageNet. The original scores are reported in Table 1 in [8]. Scores that diverge more than 0.05 from the original value are highlighted in orange.

180 and CUB-200-2011. Our trained SCOUTER models perform significantly worse on these datasets compared to what
181 the original paper reported and the scores we obtained for the FC models. Therefore, we did not find full support for claim 2, as we did not find SCOUTER to perform similar to the FC model on Con-text and CUB-200-2011.

Model	ImageNet		Con-text		CUB-200-2011	
	original	reproduced	original	reproduced	original	reproduced
FC	0.8080	0.8086	0.6732	0.6831 (± 0.0156)	0.7538	0.7824 (± 0.0274)
Scouter ₊	0.7991	0.7717	0.6870	0.5492 (± 0.0182)	0.7362	0.4718 (± 0.0212)
Scouter ₋	0.7946	0.7952	0.6866	0.6093 (± 0.0191)	0.7490	0.4143 (± 0.0235)

Table 4: Classification accuracy on various datasets. The original scores are reported in Table 3 in [8] with ResNeSt-26 as backbone. Scores that diverge more than 0.05 from the original value are highlighted in orange. Where applicable, standard deviation is reported in parentheses.

182

183 4.1.3 Result 3: reproducing ACRIMA confusion matrix evaluations

184 The results of our experiments regarding claim 3 can be seen in Table 5. We were able to reproduce all results reported
185 in the original paper. However, claim 3 states that SCOUTER achieves a higher score than the FC model in the reported
186 confusion matrix metrics. This was not the case with the results we found. There is a very slight difference between the
187 scores of SCOUTER and the FC model, where in some cases the FC model obtains a marginally higher score than (one
188 of) the SCOUTER models. In the original paper, SCOUTER also only slightly outperformed the FC model. Thus, we
189 have been able to reproduce the reported scores, but these results do not fully support claim 3.

Model	AUC		Accuracy		Precision		Recall		F1-Score		Kappa	
	original	reproduced	original	reproduced	original	reproduced	original	reproduced	original	reproduced	original	reproduced
FC	0.9997	0.9993 (± 0.0003)	0.9857	0.9843 (± 0.0141)	0.9915	0.9897 (± 0.0121)	0.9831	0.9811 (± 0.0103)	0.9872	0.9836 (± 0.0092)	0.9710	0.9561 (± 0.0227)
Scouter ₊	1.000	0.9953 (± 0.0034)	1.000	0.9831 (± 0.0129)	1.000	0.9718 (± 0.0212)	1.000	0.9919 (± 0.0096)	1.000	0.9854 (± 0.0103)	1.000	0.9566 (± 0.0329)
Scouter ₋	0.9999	0.9989 (± 0.0004)	0.9952	0.9856 (± 0.0092)	1.0000	0.9896 (± 0.0082)	0.9915	0.9768 (± 0.0213)	0.9957	0.9876 (± 0.106)	0.9903	0.9757 (± 0.0208)

Table 5: Confusion matrix metrics obtained by the models on the ACRIMA datasets. The original scores are reported in Table 4 in [8]. Scores that diverge more than 0.05 from the original value are highlighted in orange. Where applicable, standard deviation is reported in parentheses.

190 4.2 Results beyond original paper

191 Since the accuracy scores we found for the SCOUTER models trained on CUB-200-2011 and Con-text were so much
192 lower, we wanted to see if training for more epochs would be beneficial. The models did not seem to have converged
193 fully after only 20 epochs. In the examples the authors reported in their code repository, they state 150 epochs for
194 training models on these datasets, so we tested that amount. However, for SCOUTER₊ trained on CUB-200-2011 this
195 only resulted in an accuracy of 0.6443, which still deviates significantly from the original score of 0.7362.

196 5 Discussion

197 Given the results presented above, we did not verify all claims presented in Section 2.
198 Regarding claim 1, we believe this to be mostly due to the fact that we had to implement most of the metrics ourselves.
199 The authors do not report on how they implemented their metrics and what settings they have used. That means it is
200 highly likely that there exist discrepancies between our code and theirs. It could be the case that they have done some

201 additional calculations on the metrics, especially sensitivity, since that metric always lies between 0 and 1 in [8], but our
202 found scores do not. Furthermore, the sensitivity scores reported in [16] of which we have used the code also do not
203 necessarily lie in this range. Since our found scores are not in line with what was originally reported, we cannot verify
204 if SCOUTER outperforms other explanation methods.

205 The fact that we were not able to obtain similar classification scores for both SCOUTER models on Con-text and CUB-
206 200-2011 could be due to the fact that the authors used different hyperparameters than we did. Not all hyperparameters
207 were reported, so in some cases we had to make decisions ourselves. Unfortunately, we did not have the time to run an
208 extensive hyperparameter search. We could thus not verify claim 2.

209 While we did find similar scores for the confusion matrix metrics on ACRIMA, we could not find support for what
210 claim 3 states: SCOUTER outperforms the FC model on this dataset. The scores we found are very similar between the
211 models, but we would argue that this was the case in the original paper as well. The scores may not fully support the
212 claim that is being made.

213 Finally, our approach has some shortcomings that is mostly due to time constraints. We did not do three separate runs
214 for training models on ImageNet and thus our findings are based on a single run, which is not ideal. The results on
215 the ImageNet dataset should therefore not be interpreted as final. Furthermore, we were not able to experiment with
216 different backbones and only used ResNeSt-26, which was the main backbone that was used in the original paper.

217 5.1 What was easy

218 The original paper was well written, making it manageable to understand the SCOUTER model. Moreover, the code for
219 training the models was available. As such, training the models was done with relative ease. A checkpoint system for
220 the models was implemented, meaning training could be stopped and resumed later. The datasets the original authors
221 used are publicly available and straightforward to find and download.

222 5.2 What was difficult

223 The code of the original authors was devoid of documentation, making it difficult to navigate and pinpoint which
224 part performed what operation. Due to this, we spent a lot of time on any implementation we had to create or extent.
225 Furthermore, the generation of attention maps, arguably one of the most important parts, was hidden somewhere in the
226 code and not documented.

227 While the code for training the models was accessible, there was no code available for the evaluation metrics. During
228 training, we would encounter a memory allocation error every 8 to 13 epochs, meaning we had to resume from
229 checkpoints. The ImageNet dataset is very large and thus took a lot of time to train. Lastly, there was no code provided
230 for working with the ACRIMA dataset. We had to implement loading the dataset and evaluating the performance
231 ourselves.

232 5.3 Communication with original authors

233 We sent an e-mail to the authors enquiring about the missing code. However, we did not receive a reply.

234 References

- 235 [1] Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python: analyzing text with the*
236 *natural language toolkit*. " O'Reilly Media, Inc.", 2009.
- 237 [2] Jooyoung Chang, Jinho Lee, Ahnul Ha, Young Soo Han, Eunoo Bak, Seulggie Choi, Jae Moon Yun, Uk Kang,
238 Il Hyung Shin, Joo Young Shin, et al. Explaining the rationale of deep learning glaucoma decisions with adversarial
239 examples. *Ophthalmology*, 128(1):78–88, 2021.
- 240 [3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image
241 database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.
- 242 [4] Andres Diaz-Pinto, Sandra Morales, Valery Naranjo, Thomas Köhler, Jose M. Mossi, and Amparo Navea. Cnns
243 for automatic glaucoma assessment using fundus images: An extensive validation, Mar 2019.
- 244 [5] Christiane Fellbaum. *WordNet: An Electronic Lexical Database*. Bradford Books, 1998.

- 245 [6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In
246 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- 247 [7] Sezer Karaoglu, Ran Tao, Jan van Gemert, and Theo Gevers. Con-text: Text detection for fine-grained object
248 classification. In *IEEE Transactions on Image Processing (TIP)*, 2017.
- 249 [8] Liangzhi Li, Bowen Wang, Manisha Verma, Yuta Nakashima, Ryo Kawasaki, and Hajime Nagahara. SCOUTER:
250 slot attention-based classifier for explainable image recognition. *CoRR*, abs/2009.06138, 2020.
- 251 [9] Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob
252 Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. *Advances in Neural
253 Information Processing Systems*, 33:11525–11538, 2020.
- 254 [10] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss,
255 V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn:
256 Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- 257 [11] Vitali Petsiuk, Abir Das, and Kate Saenko. Rise: Randomized input sampling for explanation of black-box models.
258 In *Proceedings of the British Machine Vision Conference (BMVC)*, 2018.
- 259 [12] Zhongang Qi, Saeed Khorrani, and Fuxin Li. Visualizing deep networks by optimizing with integrated gradients.
260 In *CVPR Workshops*, volume 2, 2019.
- 261 [13] Karl Schulz, Leon Sixt, Federico Tombari, and Tim Landgraf. Restricting the flow: Information bottlenecks for
262 attribution. *arXiv preprint arXiv:2001.00396*, 2020.
- 263 [14] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset.
264 Technical report, 2011.
- 265 [15] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200.
266 Technical Report CNS-TR-2010-001, California Institute of Technology, 2010.
- 267 [16] Chih-Kuan Yeh, Cheng-Yu Hsieh, Arun Suggala, David I Inouye, and Pradeep K Ravikumar. On the (in) fidelity
268 and sensitivity of explanations. *Advances in Neural Information Processing Systems*, 32:10967–10978, 2019.
- 269 [17] Hang Zhang, Chongruo Wu, Zhongyue Zhang, Yi Zhu, Haibin Lin, Zhi Zhang, Yue Sun, Tong He, Jonas Mueller,
270 R. Manmatha, Mu Li, and Alexander Smola. Resnest: Split-attention networks, 2020.