

RISE: Randomized Input Sampling for Explanation of Black-box Models

Vitali Petsiuk
 vpetsiuk@bu.edu

Boston University
 Boston, USA

Abir Das
 dasabir@bu.edu

Kate Saenko
 saenko@bu.edu

Abstract

Deep neural networks are being used increasingly to automate data analysis and decision making, yet their decision-making process is largely unclear and is difficult to explain to the end users. In this paper, we address the problem of Explainable AI for deep neural networks that take images as input and output a class probability. We propose an approach called RISE that generates an importance map indicating how salient each pixel is for the model’s prediction. In contrast to white-box approaches that estimate pixel importance using gradients or other internal network state, RISE works on black-box models. It estimates importance empirically by probing the model with randomly masked versions of the input image and obtaining the corresponding outputs. We compare our approach to state-of-the-art importance extraction methods using both an automatic deletion/insertion metric and a pointing metric based on human-annotated object segments. Extensive experiments on several benchmark datasets show that our approach matches or exceeds the performance of other methods, including white-box approaches.

1 Introduction

Recent success of deep neural networks has led to a remarkable growth in Artificial Intelligence (AI) research. In spite of the success, it remains largely unclear how a particular neural network comes to a decision, how certain it is about the decision, if and when it can be trusted, or when it has to be corrected. In domains where a decision can have serious consequences (*e.g.*, medical diagnosis, autonomous driving, criminal justice *etc.*), it is especially important that the decision-making models are transparent. There is extensive evidence for the importance of explanation towards understanding and building trust in cognitive psychology [32], philosophy [46] and machine learning [8, 12, 21] research. In this paper, we address the problem of *Explainable AI*, *i.e.*, providing explanations for the artificially intelligent model’s decision. Specifically, we are interested in explaining classification decisions made by deep neural networks on natural images.

Consider the prediction of a popular image classification model (ResNet50 obtained from [32]) on the image depicting several sheep shown in Fig. 1(a). We might wonder, why is the model predicting the presence of a cow in this photo? Does it see all sheep as

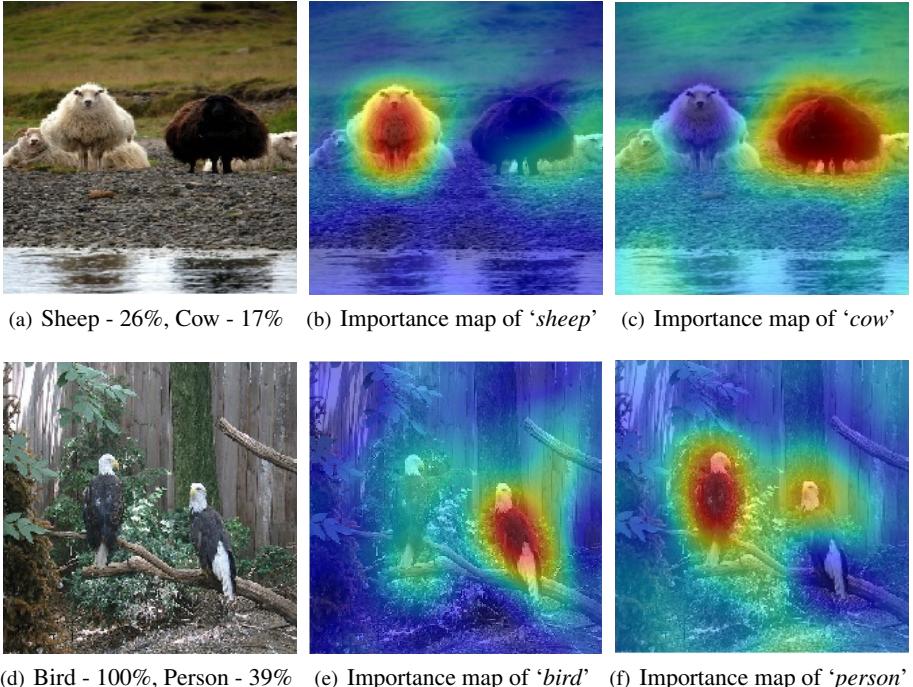


Figure 1: Our proposed RISE approach can explain why a black-box model (here, ResNet50) makes classification decisions by generating a pixel importance map for each decision (redder is more important). For the top image, it reveals that the model only recognizes the white sheep and confuses the black one with a cow; for the bottom image it confuses parts of birds with a person. (Images taken from the PASCAL VOC dataset.)

equally sheep-like? An explainable AI approach can provide answers to these questions, which in turn can help fix such mistakes. In this paper, we take a popular approach of generating a *saliency* or *importance* map that shows how important each image pixel is for the network’s prediction. In this case, our approach reveals that the ResNet model confuses the black sheep for a cow (Fig. 1(c)), potentially due to the scarcity of black colored sheep in its training data. A similar observation is made for the photo of two birds (Fig 1(d)) where the same ResNet model predicts the presence of a bird and a person. Our generated explanation reveals that the left bird provides most of the visual evidence for the ‘person’ class.

Existing methods [8, 10, 12, 15, 31, 32, 33] compute importance for a given base model (the one being explained) and an output category. However, they require access to the internals of the base model, such as the gradients of the output with respect to the input, intermediate feature maps, or the network’s weights. Many methods are also limited to certain network architectures and/or layer types [33]. In this paper, we advocate for a more general approach that can produce a saliency map for an arbitrary network without requiring access to its internals and does not require re-implementation for each network architecture. LIME [21] offers such a black-box approach by drawing random samples around the instance to be explained and fitting an approximate linear decision model. However, its saliency is based on superpixels, which may not capture correct regions (see Fig. 2(d)).

We propose a new black-box approach for estimating pixel saliency called *Randomized Input Sampling for Explanation (RISE)*. Our approach is general and applies to any off-the-

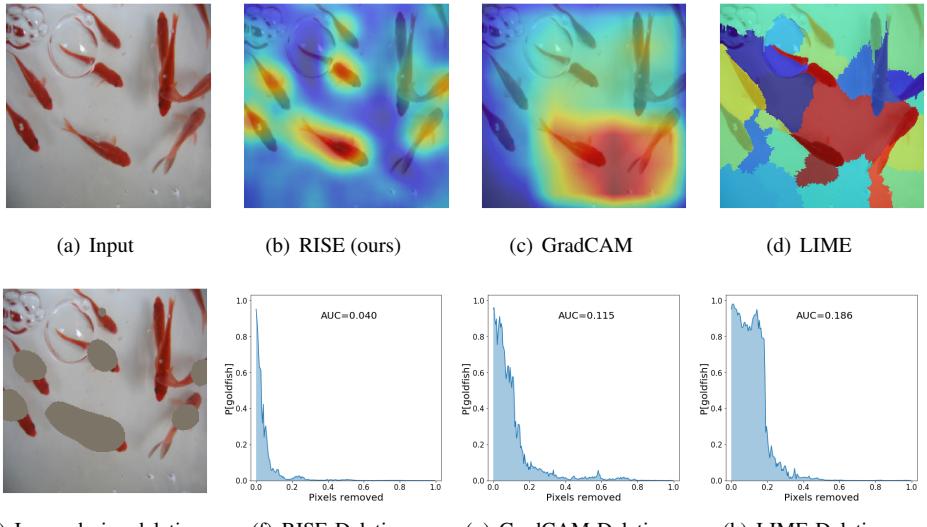


Figure 2: Estimation of importance of each pixel by RISE and other state-of-the-art methods for a base model’s prediction along with ‘deletion’ scores (AUC). The top row shows an input image (from ImageNet) and saliency maps produced by RISE, Grad-CAM [2] and LIME [21] with ResNet50 as the base network (redder values indicate higher importance). The bottom row illustrates the deletion metric: salient pixels are gradually masked from the image (2(e)) in order of decreasing importance, and the probability of the ‘goldfish’ class predicted by the network is plotted vs. the fraction of removed pixels. In this example, RISE provides more accurate saliency and achieves the lowest AUC.

shelf image network, treating it as a complete black box and not assuming access to its parameters, features or gradients. The key idea is to probe the base model by sub-sampling the input image via random masks and recording its response to each of the masked images. The final importance map is generated as a linear combination of the random binary masks where the combination weights come from the output probabilities predicted by the base model on the masked images (See Fig. 3). This seemingly simple yet surprisingly powerful approach allows us to peek inside an arbitrary network without accessing any of its internal structure. Thus, RISE is a true black-box explanation approach which is conceptually different from mainstream white-box saliency approaches such as GradCAM [23] and, in principle, is generalizable to base models of any architecture.

Another key contribution of our work is to propose causal metrics to evaluate the produced explanations. Most explanation approaches are evaluated in a human-centered way, where the generated saliency map is compared to the “ground truth” regions or bounding boxes drawn by humans in localization datasets [23, 53]. Some approaches also measure human trust or reliability on the explanations [21, 23]. Such evaluations not only require a lot of human effort but, importantly, are unfit for evaluating whether the explanation is the *true cause* of the model’s decision. They only capture how well the explanations imitate the human-annotated importance of the image regions. But an AI system could behave differently from a human and learn to use cues from the background (e.g., using grass to detect cows) or other cues that are non-intuitive to humans. Thus, a human-dependent metric cannot evaluate the correctness of an explanation that aims to extract the underlying decision process from the network.

Motivated by [8], we propose two automatic evaluation metrics: *deletion* and *insertion*.

The deletion metric measures the drop in the probability of a class as important pixels (given by the saliency map) are gradually removed from the image. A sharp drop, and thus a small area under the probability curve, are indicative of a good explanation. Fig. 2 shows plots produced by different explanation techniques for an image containing ‘goldfish’, where the total Area Under Curve (AUC) value is the smallest for our RISE model, indicating a more causal explanation. The insertion metric, on the other hand, captures the importance of the pixels in terms of their ability to *synthesize* an image and is measured by the rise in the probability of the class of interest as pixels are added according to the generated importance map. We argue that these two metrics not only alleviate the need for large-scale human evaluation or annotation effort, but are also better at assessing causal explanations by being human agnostic. For the sake of completeness, we also compare the performance of our method to state-of-the-art explanation models in terms of a human-centric evaluation metric.

2 Related work

The importance of producing explanations has been extensively studied in multiple fields, within and outside machine learning. Historically, representing knowledge using rules or decision trees [26, 27] has been found to be interpretable by humans. Another line of research focused on approximating the less interpretable models (*e.g.*, neural network, non-linear SVMs *etc.*) with simple, interpretable models such as decision rules or sparse linear models [6, 28]. In a recent work Ribeiro *et. al.* [29], fits a more interpretable approximate linear decision model (LIME) in the vicinity of a particular input. Though the approximation is fairly good locally, for a sufficiently complex model, a linear approximation may not lead to a faithful representation of the non-linear model. The LIME model can be applied to black-box networks like our approach, but its reliance on superpixels leads to inferior importance maps as shown in our experiments.

To explain classification decisions in images, previous works either visually ground image regions that strongly support the decision [18, 23] or generate a textual description of why the decision was made [30]. The visual grounding is generally expressed as a saliency or importance map which shows the importance of each pixel towards the model’s decision. Existing approaches to deep neural network explanation either design ‘interpretable’ network architectures or attempt to explain or ‘justify’ decisions made by an existing model.

Within the class of interpretable architectures, Xu *et. al.* [9], proposed an interpretable image captioning system by incorporating an attention network which learns where to look next in an image before producing each word of the caption. A neural module network is employed in [1, 2] to produce the answers to visual question-answering problems in an interpretable manner by learning to divide the problem into subproblems. However, these approaches achieve interpretability by incorporating changes to a white-box base model and are constrained to use specific network architectures.

Neural justification approaches attempt to justify the decision of a base model. *Third-person* models [10, 18] train additional models from human annotated ‘ground truth’ reasoning in the form of saliency maps or textual justifications. The success of such methods depends on the availability of tediously labeled ground-truth explanations, and they do not produce high-fidelity explanations. Alternatively, *first-person* models [1, 8, 23, 25] aim to generate explanations providing evidence for the model’s underlying decision process without using an additional model. In our work, we focus on producing a first-person justification.

Several approaches generate importance maps by isolating contributions of image re-

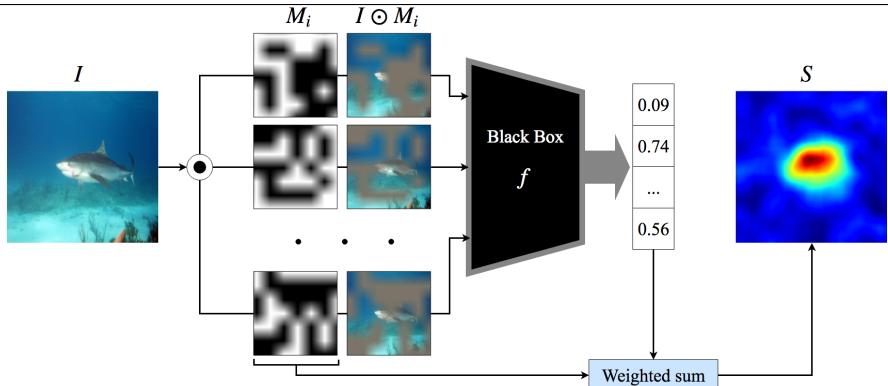


Figure 3: Overview of RISE: Input image I is element-wise multiplied with random masks M_i and the masked images are fed to the base model. The saliency map is a linear combination of the masks where the weights come from the score of the target class corresponding to the respective masked inputs.

gions to the prediction. In one of the early works [31], Zeiler *et al.* visualize the internal representation learned by CNNs using deconvolutional networks. Other approaches [10, 15, 30] have tried to synthesize an input (an image) that highly activates a neuron. The Class Activation Mapping (CAM) approach [32] achieves class-specific importance of each location of an image by computing a weighted sum of the feature activation values at that location across all channels. However, the approach can only be applied to a particular kind of CNN architecture where a global average pooling is performed over convolutional feature map channels immediately prior to the classification layer. Grad-CAM [33] extends CAM by weighing the feature activation values at every location with the average gradient of the class score (w.r.t. the feature activation values) for every feature map channel. Zhang *et al.* [37] introduce a probabilistic winner-take-all strategy to compute top-down importance of neurons towards model predictions. Fong *et al.* [8] and Cao *et al.* [10] learn a perturbation mask that maximally affects the model’s output by backpropagating the error signals through the model. However, all of the above methods [10, 8, 15, 30, 25, 30, 32, 33] assume access to the internals of the base model to obtain feature activation values, gradients or weights. RISE is a more general framework as the importance map is obtained with access to only the input and output of the base model.

3 Randomized Input Sampling for Explanation (RISE)

One way to measure the importance of an image region is to obscure or ‘perturb’ it and observe how much this affects the black box decision. For example, this can be done by setting pixel intensities to zero [8, 21, 31], blurring the region [8] or by adding noise. In this work we estimate the importance of pixels by dimming them in random combinations, reducing their intensities down to zero. We model this by multiplying an image with a $[0, 1]$ valued mask. The mask generation process is described in detail in section 3.2.

3.1 Random Masking

Let $f : \mathcal{I} \rightarrow \mathbb{R}$ be a black-box model, that for a given input from \mathcal{I} produces scalar confidence score. In our case, \mathcal{I} is the space of color images $\mathcal{I} = \{I \mid I : \Lambda \rightarrow \mathbb{R}^3\}$ of size $H \times W$

$(\Lambda = \{1, \dots, H\} \times \{1, \dots, W\})$, where every image I is a mapping from coordinates to three color values. For example, f may be a classifier that produces the probability that object of some class is present in the image, or a captioning model that outputs the probability of the next word given a partial sentence.

Let $M : \Lambda \rightarrow \{0, 1\}$ be a random binary mask with distribution \mathcal{D} . Consider the random variable $f(I \odot M)$, where \odot denotes element-wise multiplication. First, the image is masked by preserving only a subset of pixels. Then, the confidence score for the masked image is computed by the black box. We define importance of pixel $\lambda \in \Lambda$ as the expected score over all possible masks M conditioned on the event that pixel λ is observed, *i.e.*, $M(\lambda) = 1$:

$$S_{I,f}(\lambda) = \mathbb{E}_M [f(I \odot M) | M(\lambda) = 1]. \quad (1)$$

The intuition behind this is that $f(I \odot M)$ is high when pixels preserved by mask M are important.

Eq. (1) can be rewritten as a summation over mask $m : \Lambda \rightarrow \{0, 1\}$:

$$\begin{aligned} S_{I,f}(\lambda) &= \sum_m f(I \odot m) P[M = m | M(\lambda) = 1] \\ &= \frac{1}{P[M(\lambda) = 1]} \sum_m f(I \odot m) P[M = m, M(\lambda) = 1]. \end{aligned} \quad (2)$$

Now,

$$\begin{aligned} P[M = m, M(\lambda) = 1] &= \begin{cases} 0, & \text{if } m(\lambda) = 0, \\ P[M = m], & \text{if } m(\lambda) = 1. \end{cases} \\ &= m(\lambda) P[M = m]. \end{aligned} \quad (3)$$

Substituting $P[M = m, M(\lambda) = 1]$ from (3) in (2),

$$S_{I,f}(\lambda) = \frac{1}{P[M(\lambda) = 1]} \sum_m f(I \odot m) \cdot m(\lambda) \cdot P[M = m]. \quad (4)$$

It can be written in matrix notation, combined with the fact that $P[M(\lambda) = 1] = \mathbb{E}[M(\lambda)]$:

$$S_{I,f}(\lambda) = \frac{1}{\mathbb{E}[M]} \sum_m f(I \odot m) \cdot m \cdot P[M = m]. \quad (5)$$

Thus, saliency map can be computed as a weighted sum of random masks, where weights are the probability scores, that masks produce, adjusted for the distribution of the random masks.

We propose to generate importance maps by empirically estimating the sum in equation (5) using Monte Carlo sampling. To produce an importance map, explaining the decision of model f on image I , we sample set of masks $\{M_1, \dots, M_N\}$ according to \mathcal{D} and probe the model by running it on masked images $I \odot M_i$, $i = 1, \dots, N$. Then, we take the weighted average of the masks where the weights are the confidence scores $f(I \odot M_i)$ and normalize it by the expectation of M :

$$S_{I,f}(\lambda) \stackrel{\text{MC}}{\approx} \frac{1}{\mathbb{E}[M] \cdot N} \sum_{i=1}^N f(I \odot M_i) \cdot M_i(\lambda). \quad (6)$$

Note that our method does not use any information from inside the model and thus, is suitable for explaining black-box models.

3.2 Mask generation

Masking pixels independently may cause adversarial effects: a slight change in pixel values may cause significant variation in the model’s confidence scores. Moreover, generating masks by independently setting their elements to zeros and ones will result in mask space of size $2^{H \times W}$. A larger space size requires more samples for a good estimation in equation (6).

To address these issues we first sample smaller binary masks and then upsample them to larger resolution using bilinear interpolation. Bilinear upsampling does not introduce sharp edges in $I \odot M_i$ as well as results in a smooth importance map S . After interpolation, masks M_i are no longer binary, but have values in $[0, 1]$. Finally, to allow more flexible masking, we shift all masks by a random number of pixels in both spatial directions.

Formally, mask generation can be summarized as:

1. Sample N binary masks of size $h \times w$ (smaller than image size $H \times W$) by setting each element independently to 1 with probability p and to 0 with the remaining probability.
2. Upsample all masks to size $(h+1)C_H \times (w+1)C_W$ using bilinear interpolation, where $C_H \times C_W = \lfloor H/h \rfloor \times \lfloor W/w \rfloor$ is the size of the cell in the upsampled mask.
3. Crop areas $H \times W$ with uniformly random indents from $(0, 0)$ up to (C_H, C_W) .

4 Experiments

Datasets and Base Models: We evaluated RISE on 3 publicly available object classification datasets, namely, PASCAL VOC07 [4], MSCOCO2014 [13] and ImageNet [22]. Given a base model, we test importance maps generated by different explanation methods for a target object category present in images from the VOC and MSCOCO datasets. For the ImageNet dataset, we test the explanation generated for the top probable class of the image. We chose the particular versions of the VOC and MSCOCO datasets to compare fairly with the state-of-the-art reporting on the same datasets and same base models. For these two datasets, we used ResNet50 [8] and VGG16 [24] networks trained by [52] as base models. For ImageNet, the same base models were downloaded from the PyTorch model zoo ¹.

4.1 Evaluation Metrics

Despite a growing body of research focusing on explainable machine learning, there is still no consensus about how to measure the explainability of a machine learning model [19]. As a result, human evaluation has been the predominant way to assess model explanation by measuring it from the perspective of transparency, user trust or human comprehension of the decisions made by the model [10]. Existing justification methods [23, 52] have evaluated saliency maps by their ability to localize objects. However, localization is merely a proxy for human explanation and may not correctly capture what *causes* the base model to make a decision irrespective of whether the decision is right or wrong as far as the proxy task is concerned. As a typical example, let us consider an image of a car driving on a road. Evaluating an explanation against the localization bounding box of the car does not give credit for (in fact discredits) correctly capturing ‘road’ as a possible cause behind the base model’s decision of classifying the image as that of a car. We argue that keeping humans out of the loop for evaluation makes it more fair and true to the classifier’s own *view* on the problem rather than representing a human’s view. Such a metric is not only objective (free from human bias) in nature but also saves time and resources.

¹<https://github.com/pytorch/vision>

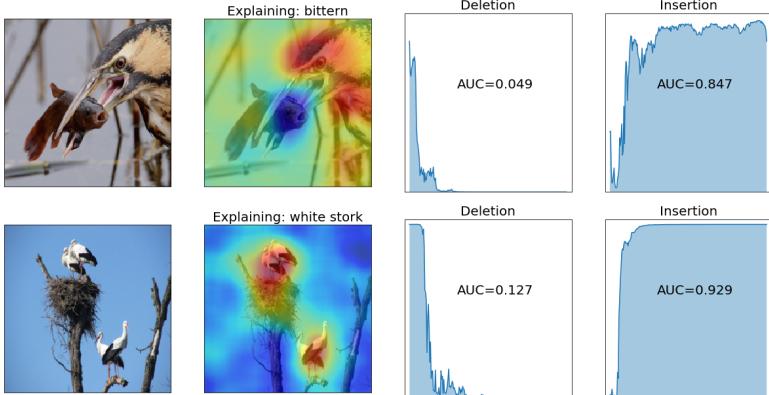


Figure 4: RISE-generated importance maps (second column) for two representative images (first column) with deletion (third column) and insertion (fourth column) curves.

Causal metrics for explanations: To address these issues, we propose two automatic evaluation metrics: *deletion* and *insertion*, motivated by [8]. The intuition behind the *deletion metric* is that the removal of the ‘cause’ will force the base model to change its decision. Specifically, this metric measures a decrease in the probability of the predicted class as more and more important pixels are removed, where the importance is obtained from the importance map. A sharp drop and thus a low area under the probability curve (as a function of the fraction of removed pixels) means a good explanation. The *insertion metric*, on the other hand, takes a complementary approach. It measures the increase in probability as more and more pixels are introduced, with higher AUC indicative of a better explanation.

There are several ways of removing pixels from an image [9], e.g., setting the pixel values to zero or any other constant gray value, blurring the pixels or even cropping out a tight bounding box. The same is true when pixels are introduced, e.g., they can be introduced to a constant canvas or by starting with a highly blurred image and gradually unblurring regions. All of these approaches have different pros and cons. A common issue is the introduction of spurious evidence which can fool the classifier. For example, if pixels are introduced to a constant canvas and if the introduced region happens to be oval in shape, the classifier may classify the image as a ‘balloon’ (possibly a printed balloon) with high probability. This issue is less severe if pixels are introduced to an initially blurred canvas as blurring takes away most of the finer details of an image without exposing it to sharp edges as image regions are introduced. This strategy gives higher scores for all methods, so we adopt it for insertion. For deletion, the aim is to fool the classifier as quickly as possible and blurring small regions instead of setting them to a constant gray level does not help. This is because a good classifier is usually able to fill in the missing details quite remarkably from the surrounding regions and from the small amount of low-frequency information left after blurring a tiny region. As a result, we set the image regions to constant values when removing them for the deletion metric evaluation. We used the same strategies for all the existing approaches with which we compared our method in terms of these two metrics.

Pointing game: We also evaluate explanations in terms of a human evaluation metric, the *pointing game* introduced in [5]. If the highest saliency point lies inside the human-annotated bounding box of an object, it is counted as a hit. The pointing game accuracy is given by $\frac{\#Hits}{\#Hits + \#Misses}$, averaged over all target categories in the dataset. For a classification

Table 1: Comparative evaluation in terms of deletion (lower is better) and insertion (higher is better) scores on ImageNet dataset. Except for Grad-CAM, the rest are black-box explanation models.

Method	ResNet50		VGG16	
	Deletion	Insertion	Deletion	Insertion
Grad-CAM [20]	0.1232	0.6766	0.1087	0.6149
Sliding window [31]	0.1421	0.6618	0.1158	0.5917
LIME [21]	0.1217	0.6940	0.1014	0.6167
RISE (ours)	0.1076 ± 0.0005	0.7267 ± 0.0006	0.0980 ± 0.0025	0.6663 ± 0.0014

model that learns to rely on objects, this metric should be high for a good explanation.

4.2 Experimental Results

Experimental Settings: The binary random masks are generated with equal probabilities for 0’s and 1’s. For different CNN classifiers, we empirically select different numbers of masks, in particular, we used 4000 masks for the VGG16 network and 8000 for ResNet50. We have used $h = w = 7$ and $H = W = 224$ throughout. All the results used for comparison were either taken from published works or by running the publicly available code on datasets for which reported results could not be obtained.

Deletion and Insertion scores: Table 1 shows a comparative evaluation of RISE with other state-of-the-art approaches in terms of both deletion and insertion metrics on val split of ImageNet. RISE reports an average value with associated standard deviations for 3 independent runs. The sliding window approach [31] systematically occludes fixed size image regions and probes the model with the perturbed image to measure the importance of the occluded region. We used a sliding window of size 64×64 with stride 8. For LIME [21], the number of samples was set to 1000 (taken from the code). For this experiment, we used the ImageNet classification dataset where no ground truth segmentation or localization mask is provided and thus explainability performance can only be measured via automatic metrics like deletion and insertion. For both the base models and according to both the metrics, RISE provides better performance, outperforming even the white-box Grad-CAM method. The values are better for ResNet50 which is intuitive as it is a better classification model than VGG16. However, due to increased number of forward passes, RISE is heavy in computation. This can potentially be addressed by intelligently sampling fewer number of random masks which is kept as a future work. RISE, sometimes, provides noisy importance maps due to sampling approximation especially in presence of objects with varying sizes. Fig. 4 shows examples of RISE-generated importance maps along with the deletion and insertion curves. The appendix contains additional visual examples including a few noisy importance maps.

Pointing game accuracy: The performance in terms of pointing game accuracy is shown in Table 2 for the test split of PASCAL VOC07 and val split of MSCOCO2014 datasets. In this table, RISE is the only black-box method. The base models are obtained from [20] and thus we list the pointing game accuracies reported in the paper. RISE reports an average value of 3 independent runs; low standard deviation values indicate the robustness of the proposed approach against the randomness of the masks. For VGG16, RISE performs consistently better than all of the white-box methods with a significantly improved performance for the VOC dataset. For the deeper ResNet50 network with residual connections, RISE does not have the highest pointing accuracy but comes close. We stress again that good pointing accuracy may not correlate with actual causal processes in a network, however, RISE is competitive despite being black-box and more general than methods like CAM, which is only applicable to architectures without fully-connected layers.

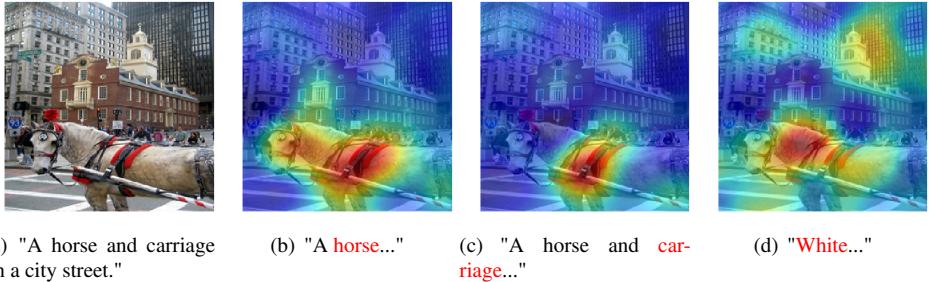


Figure 5: Explanations of image captioning models. (a) is the image with the caption generated by [■]. (b) and (c) show the importance map generated by RISE for two words ‘horse’ and ‘carriage’ respectively from the generated caption. (d) shows the importance map for an arbitrary word ‘white’.

Table 2: Mean accuracy (%) in the pointing game. Except for RISE, the rest require white-box model.

Base model	Dataset	AM [■]	Deconv [■]	CAM [■]	MWP [■]	c-MWP [■]	RISE
VGG16	VOC	76.00	75.50	-	76.90	80.00	87.33 ± 0.49
	MSCOCO	37.10	38.60	-	39.50	49.60	50.71 ± 0.10
Resnet50	VOC	65.80	73.00	90.60	80.90	89.20	88.94 ± 0.61
	MSCOCO	30.40	38.2	58.4	46.8	57.4	55.58 ± 0.51

4.3 RISE for Captioning

RISE can easily be extended to explain captions for any image description system. Some existing works use a separate attention network [■] or assume access to feature activations [■] and/or gradient values [■] to ground words in an image caption. The most similar to our work is Ramanishka *et al.* [■] where the base model is probed with conv features from small patches of the input image to estimate its importance for each word in the caption. However, our approach is not constrained to a single fixed size patch and is thus less sensitive to object sizes as well as better at capturing additional context that may be present in the image. We provide a small example of RISE being applied for explaining image caption.

We take a base captioning model [■] that models the probability of the next word w_k given a partial sentence $s = (w_1, \dots, w_{k-1})$ and an input image I :

$$f(I, s, w_k) = P[w_k | I, w_1, \dots, w_{k-1}] \quad (7)$$

We probe the base model by running it on a set of N randomly masked inputs $f(I \odot M_i, s, w_k)$ and computing saliency as $\frac{1}{N \cdot \mathbb{E}[M]} \sum_{i=1}^N f(I \odot M_i, s, w_k) \cdot M_i$ for each word in s . Input sentence s can be any arbitrary sentence including the caption generated by the base model itself. Three such explanation instances for MSCOCO image are shown in Fig. 5.

5 Conclusion

This paper presented RISE, an approach for explaining black-box models by estimating the importance of input image regions for the model’s prediction. Despite its simplicity and generality, the method outperforms existing explanation approaches in terms of automatic causal metrics and performs competitively in terms of the human-centric pointing metric. Future work will be to exploit the generality of the approach for explaining decisions made by complex networks in video and other domains.

Acknowledgement: This work was partially supported by the DARPA XAI program.

References

- [1] Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. Learning to Compose Neural Networks for Question Answering. In *The Conference of the North American Chapter of the Association for Computational Linguistics*, pages 1545–1554, 2016.
- [2] Chunshui Cao, Xianming Liu, Yi Yang, Yinan Yu, Jiang Wang, Zilei Wang, Yongzhen Huang, Liang Wang, Chang Huang, Wei Xu, et al. Look and think twice: Capturing top-down visual attention with feedback convolutional neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2956–2964, 2015.
- [3] Mark W Craven and Jude W Shavlik. *Extracting Comprehensible Models from Trained Neural Networks*. PhD thesis, University of Wisconsin, Madison, 1996.
- [4] Piotr Dabkowski and Yarin Gal. Real Time Image Saliency for Black Box Classifiers. In *Neural Information Processing Systems*, pages 6970–6979, 2017.
- [5] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2625–2634, 2015.
- [6] Mary T Dzindolet, Scott A Peterson, Regina A Pomranky, Linda G Pierce, and Hall P Beck. The Role of Trust in Automation Reliance. *International Journal of Human-Computer Studies*, 58(6):697–718, 2003.
- [7] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The Pascal Visual Object Classes (VOC) Challenge. *International journal of computer vision*, 88(2):303–338, 2010.
- [8] Ruth C Fong and Andrea Vedaldi. Interpretable Explanations of Black Boxes by Meaningful Perturbation. In *IEEE International Conference on Computer Vision*, Oct 2017.
- [9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [10] Lisa Anne Hendricks, Zeynep Akata, Marcus Rohrbach, Jeff Donahue, Bernt Schiele, and Trevor Darrell. Generating Visual Explanations. In *European Conference on Computer Vision*, pages 3–19, 2016.
- [11] Bernease Herman. The Promise and Peril of Human Evaluation for Model Interpretability. In *Interpretable ML Symposium, Neural Information Processing Systems*, Dec 2017.
- [12] Ronghang Hu, Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Kate Saenko. Learning to Reason: End-To-End Module Networks for Visual Question Answering. In *IEEE International Conference on Computer Vision*, Oct 2017.
- [13] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755, 2014.

- [14] Zachary C Lipton. The Mythos of Model Interpretability. *arXiv preprint arXiv:1606.03490*, 2016.
- [15] Tania Lombrozo. The Structure and Function of Explanations. *Trends in Cognitive Sciences*, 10(10):464–470, 2006.
- [16] Tania Lombrozo. The Instrumental Value of Explanations. *Philosophy Compass*, 6(8): 539–551, 2011.
- [17] Anh Nguyen, Alexey Dosovitskiy, Jason Yosinski, Thomas Brox, and Jeff Clune. Synthesizing the Preferred Inputs for Neurons in Neural Networks via Deep Generator Networks. In *Neural Information Processing Systems*, pages 3387–3395, 2016.
- [18] Dong Huk Park, Lisa Anne Hendricks, Zeynep Akata, Anna Rohrbach, Bernt Schiele, Trevor Darrell, and Marcus Rohrbach. Multimodal Explanations: Justifying Decisions and Pointing to the Evidence. In *IEEE Computer Vision and Pattern Recognition*, Jun 2018.
- [19] Forough Poursabzi-Sangdeh, Daniel G Goldstein, Jake M Hofman, Jennifer W Vaughan, and Hanna Wallach. Manipulating and Measuring Model Interpretability. *arXiv preprint arXiv:1802.07810*, 2018.
- [20] Vasili Ramanishka, Abir Das, Jianming Zhang, and Kate Saenko. Top-Down Visual Saliency Guided by Captions. In *IEEE Conference on Computer Vision and Pattern Recognition*, July 2017.
- [21] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Why Should I Trust You?: Explaining the Predictions of any Classifier. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144. ACM, 2016.
- [22] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.
- [23] 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 International Conference on Computer Vision*, Oct 2017.
- [24] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *International Conference on Learning Representations*, May 2015.
- [25] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. *arXiv preprint arXiv:1312.6034*, 2013.
- [26] William R Swartout. Producing Explanations and Justifications of Expert Consulting Programs. 1981.
- [27] William R Swartout and Johanna D Moore. Explanation in Second Generation Expert Systems. In *Second Generation Expert Systems*, pages 543–585. Springer, 1993.

- [28] Sebastian Thrun. Extracting Rules from Artificial Neural Networks with Distributed Representations. In *Advances in Neural Information Processing Systems*, pages 505–512, 1995.
- [29] Kelvin Xu, Jimmy Lei Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. In *International Conference on Machine Learning*, pages 2048–2057, 2015.
- [30] Jason Yosinski, Jeff Clune, Thomas Fuchs, and Hod Lipson. Understanding Neural Networks Through Deep Visualization. In *International Conference on Machine Learning Workshop on Deep Learning*, 2014.
- [31] Matthew D Zeiler and Rob Fergus. Visualizing and Understanding Convolutional Networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.
- [32] Jianming Zhang, Sarah Adel Bargal, Zhe Lin, Xiaohui Shen Jonathan Brandt, and Stan Sclaroff. Top-down Neural Attention by Excitation Backprop. *International Journal of Computer Vision*, Dec 2017.
- [33] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning Deep Features for Discriminative Localization. In *IEEE Computer Vision and Pattern Recognition*, pages 2921–2929, 2016.

Appendix A Algorithms to compute causal metrics

Algorithm to compute deletion score.

Algorithm 1

```

1: procedure DELETION
2:   Input: black box  $f$ , image  $I$ , importance map  $S$ , number of pixels  $N$  removed per step
3:   Output: deletion score  $d$ 
4:    $n \leftarrow 0$ 
5:    $h_n \leftarrow f(I)$ 
6:   while  $I$  has non-zero pixels do
7:     According to  $S$ , set next  $N$  pixels in  $I$  to 0
8:      $n \leftarrow n + 1$ 
9:      $h_n \leftarrow f(I)$ 
10:     $d \leftarrow \text{AreaUnderCurve}(h_i \text{ vs. } i/n, \quad \forall i = 0, \dots, n)$ 
11:   return  $d$ 
```

Algorithm to compute insertion score.

Algorithm 2

```

1: procedure INSERTION
2:   Input: black box  $f$ , image  $I$ , importance map  $S$ , number of pixels  $N$  removed per step
3:   Output: insertion score  $d$ 
4:    $n \leftarrow 0$ 
5:    $I' \leftarrow \text{Blur}(I)$ 
6:    $h_n \leftarrow f(I)$ 
7:   while  $I \neq I'$  do
8:     According to  $S$ , set next  $N$  pixels in  $I'$  to corresponding pixels in  $I$ 
9:      $n \leftarrow n + 1$ 
10:     $h_n \leftarrow f(I')$ 
11:     $d \leftarrow \text{AreaUnderCurve}(h_i \text{ vs. } i/n, \quad \forall i = 0, \dots, n)$ 
12:   return  $d$ 
```

Appendix B More saliency maps and their scores

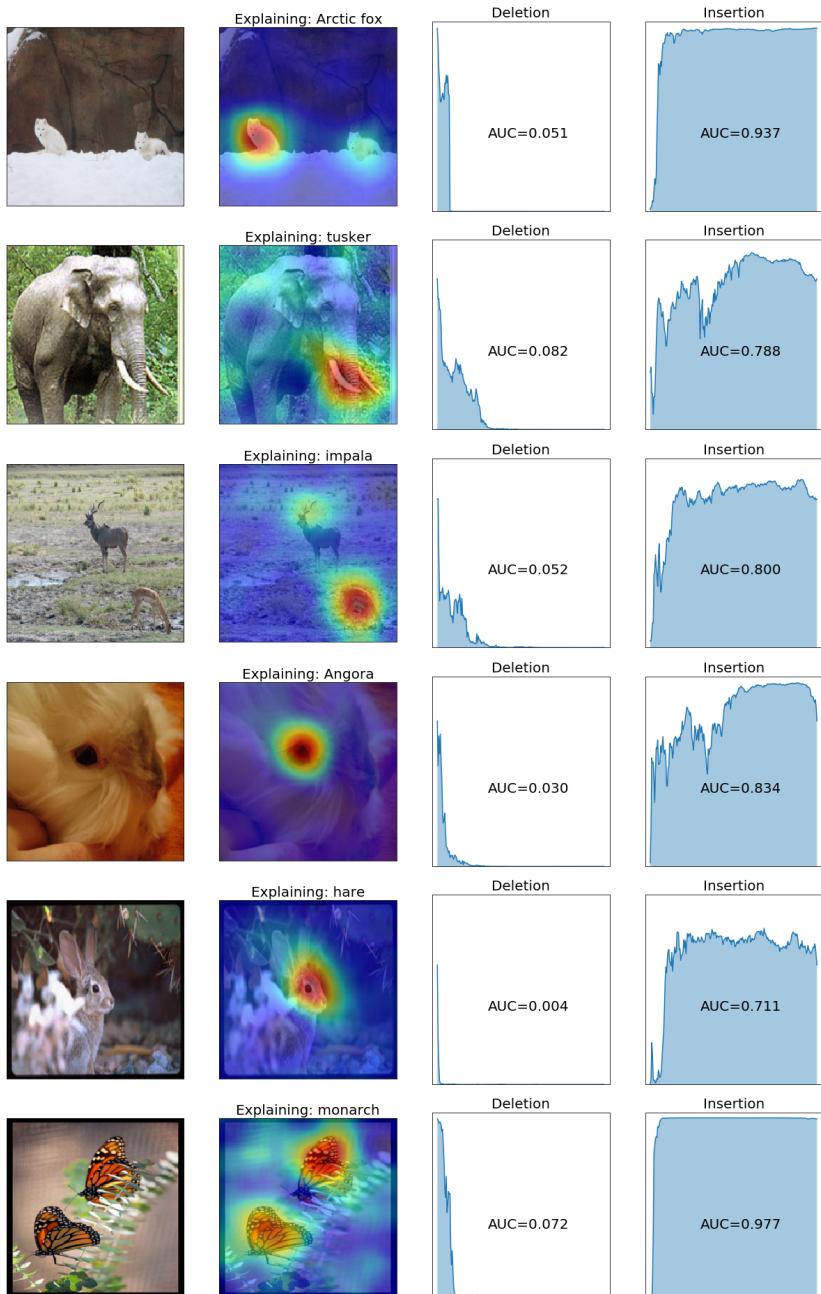


Figure 6: RISE-generated importance maps (second column) for representative images (first column) with deletion (third column) and insertion (fourth column) curves.

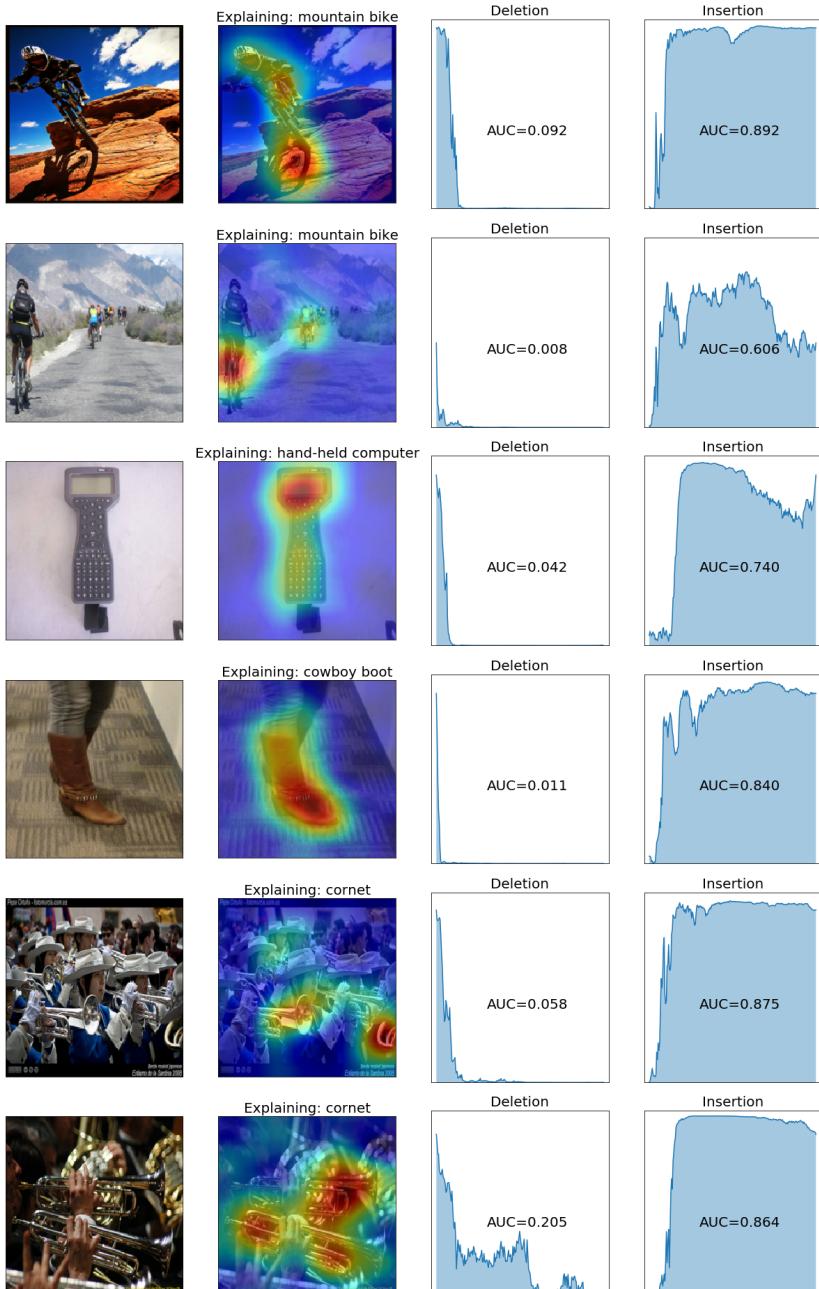


Figure 7: RISE-generated importance maps (second column) for representative images (first column) with deletion (third column) and insertion (fourth column) curves.

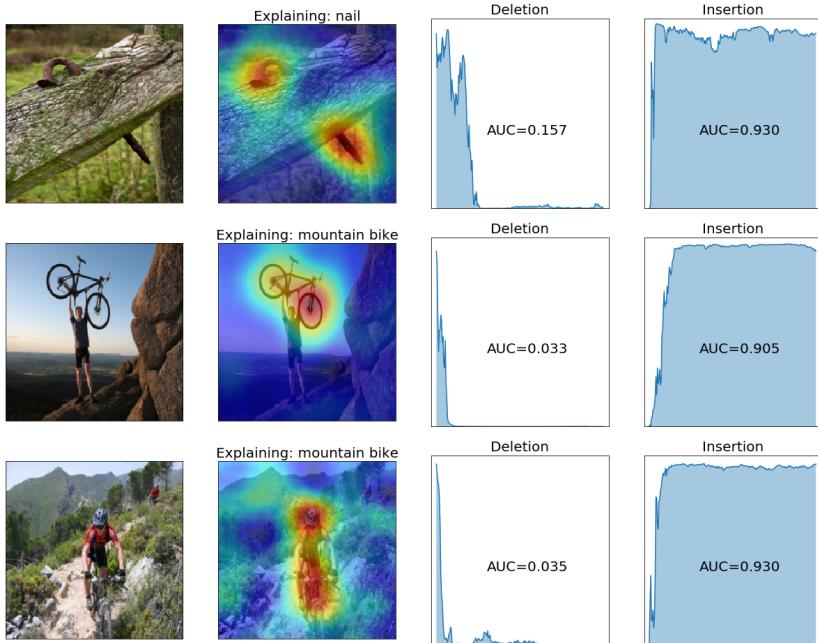


Figure 8: RISE-generated importance maps (second column) for representative images (first column) with deletion (third column) and insertion (fourth column) curves.

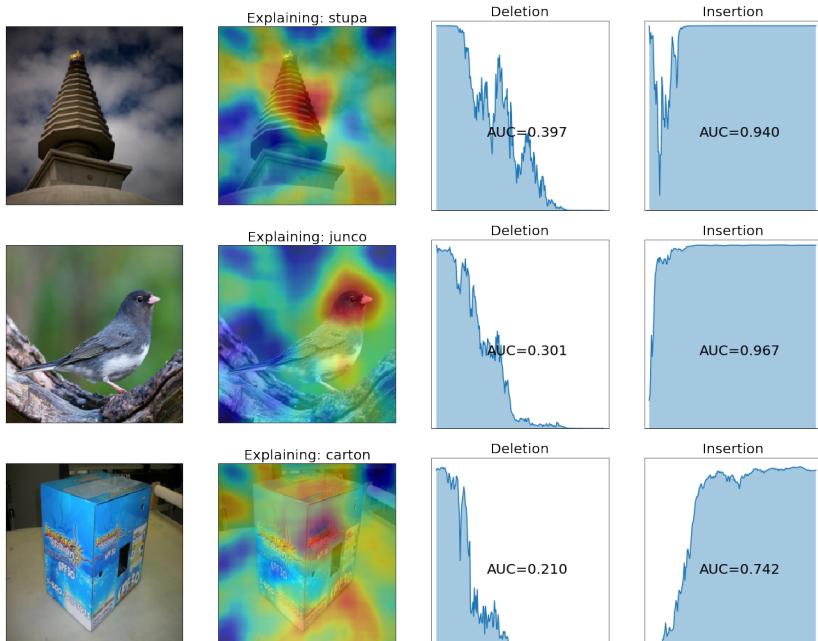


Figure 9: Failure cases. In some cases RISE does pick up more important features, but cannot get rid of the background noise (in part due to MC approximation with only a subset) like in rows 1 and 2.