RESCIENCE C

Replication / ML Reproducibility Challenge 2022

[Re] Reproducibility Study of "CartoonX: Cartoon Explanations of Image Classifiers"

Aditya Prakash Patra[1,2], Sina Taslimi[1,2,*], Luke Chin A Foeng[1,2] Pratik Kayal[1,2]
[1]University of Amsterdam – [2]Equal contributions

Edited by
(Editor)

Reviewed by
(Reviewer 1)
(Reviewer 2)

Received
03 February 2023

Published
–

DOI
–

Reproducibility Summary

Scope of Reproducibility – In this reproducibility study, we verify the claims and contributions in *Cartoon Explanations of Image Classifiers* by Kolek et al. [1]. These include (i) A proposed technique named CartoonX used to extract visual explanations for predictions via image classification networks, (ii) CartoonX being able to reveal piece-wise smooth regions of the image, unlike previous methods, which extract relevant pixel-sparse regions, and (iii) CartoonX achieving lower distortion values than these methods.

Methodology – The authors provide their substantial codebase via Git Hub, which played a vital role initially. However, it was discovered that several figures would require additional scripts to reproduce them. Additionally, the GPUs used consisted of several different CUDA-enabled GPU models, including a GTX 2060 Ti and an RTX 3060.

Results – We verified the main claims of the paper and offered extensions. Our qualitative CartoonX and PixelRDE visualization results were similar to the original paper's. From visualizing them, we saw that CartoonX could reveal piece-wise information in the image relevant to the classifier. Our quantitative distortion plots followed trends similar to the original plots, allowing us to verify their claims, but only after adjustments to the unclear 'λ' and 'number of steps' hyperparameters the paper provides.

What was easy – The initial implementation of the provided script was simple as the code provided included instructions on dependency installation procedures and how to run the script for an initial qualitative assessment.

What was difficult – While the qualitative result reproduction was simple, quantitative reproduction remained difficult. The main issue was the experiment specifications needed to create the quantitative results. Lastly, CartoonX's predictions depend on the hyperparameter choices, mainly 'λ' and the number of 'iteration steps'.

Communication with original authors – We reached out to the authors with several questions regarding the quantitative results. While they could answer several questions posed in time, some clarifications needed to be included. New questions arose after this interaction but were left unanswered due to the limited timeline of this reproduction.

Copyright © 2023 A. P. Patra, S. Taslimi, L. Chin A Foeng, P. Kayal , released under a Creative Commons Attribution 4.0 International license.
Correspondence should be addressed to Sina Taslimi (taslimisina@gmail.com)
The authors have declared that no competing interests exist.
Code is available at https://github.com/KAISER1997/FACTAI_CARTOONX_Reproduce.

1 Introduction

Image classification models have grown exponentially in size and prevalence over the last few years. This, along with their "black box" nature, highlights the challenge of explaining the results and decisions made by these image classification models in a more interpretable manner for their human operators. There has been considerable research in this domain ([2, 3, 4, 5, 6, 7, 8]) to generate importance mask on the pixel space. However, they frequently result in sparse and shaky explanations.
This paper presents a novel explanation method that operates in the wavelet domain of images instead of the conventional pixel space and aims to extract relevant piecewise smooth parts of an image. This is achieved by demanding sparsity in the wavelet domain, which can further result in piece-wise smooth explanations.

2 Scope of Reproducibility

The paper that we reproduce addresses the difficulties associated with pixel-based explanation methods stemming from the fact that they produce pixel-sparse explanations that incur higher levels of distortion than the suggested CartoonX method. In our reproducibility study, we reproduce the results of the original work, assess the claims made, and present several additional experiments to extend the previous work. In short, we present the following:

- Through the reproduction of qualitative and quantitative results, we verify their claim that CartoonX gives better explanations when compared to other pixel-based explanation methods.
- We extend the previous paper through experimentation with the obfuscation method to improve the distortion values achieved.
- We extend the applications of CartoonX to the domains of semantic segmentation and object detection.
- We further conduct an ablation study on the use of CartoonX over several different deep neural network architectures.

In order to examine and explain the points touched upon, the report first discusses the CartoonX methodology in more detail in section 3, followed by section 4, where the reproduction and extensions made are described. After this, we discuss the results of the reproducibility study and extensions in section 5, followed by the discussion in section 6.

3 CartoonX

CartoonX is a model-agnostic explanation method that can extract explanations from image classifiers via the RDE(rate-distortion explanation)framework. We briefly summarize the main ideas of this methodology in this section.

Let $\Phi : \mathbb{R}^n \to \mathbb{R}^m$ be a deep neural network and $x \in \mathbb{R}^n$ a data point with a data representation $x = f(h_1, ..., h_k)$ where f is the inverse discrete wavelet transform operator. Here, coefficients $[h_i]_{i=1}^n$ can be obtained by applying discrete wavelet transform on x. Let the mask be $s \in [0,1]^k$, $\mathcal{V}$ be a probability distribution over $\prod_{i=1}^k \mathbb{R}^c$. Then the *obfuscation* of x concerning s and $\mathcal{V}$ is defined as $y := f(s \odot h + (1-s) \odot v)$. Thus the expected distortion is given by

$$D(x, s, \mathcal{V}, \Phi) := \mathop{\mathbb{E}}_{v \sim \mathcal{V}} \Big[d\Big(\Phi(x), \Phi(y)\Big) \Big],$$

where d is a measure of distortion and Φ represent the model decision .The aim is to find a sparse mask s and minimize $D(x, s, \mathcal{V}, \Phi)$. This results in the following constrained optimization problem.

$$\min_{s \in [0,1]^k} \quad D(x, s, \mathcal{V}, \Phi) + \lambda \|s\|_1, \tag{1}$$

This objective can be optimized via stochastic gradient descent (SGD). Also note that they approximate $D(x, s, \mathcal{V}, \Phi)$ with i.i.d. samples from $v \sim \mathcal{V}$. Finally they visualize the mask s as a piece-wise smooth image in pixel space by multiplying the mask with the DWT coefficients of the greyscale image, followed by applying inverse DWT.

4 Methodology

In order to replicate the results generated by the original authors, we used the script provided by the authors. Their step-by-step documentation detailing how to install specific dependencies, such as the *PyTorch Wavelets* module, made it relatively easy to reproduce some of their qualitative results for the CartoonX and Pixel RDE implementations. However, the other explainability methods shown in the qualitative comparisons required additional implementation. Additional scripts were also required for the sensitivity analysis.
Apart from this, we also attempt to extend the applications of CartoonX to Semantic Segmentation and Object Detection tasks. Semantic segmentation deals with assigning a class to each pixel of an image. Since this classification problem is applied to each pixel, it is fairly straightforward to plug CartoonX to visualize the explanations. Object detection, however, is more difficult with assigning bounding boxes around objects in an image and their identification. Specifically, in our experiment, we work with the YOLO algorithm. Applying CartoonX directly on YOLO wouldn't work because with every update of the wavelet mask, the network outputs change, and so the Non- max suppression module of the YOLO ends up with a different set of objects every update. This makes computing the loss difficult. To circumvent this, we calculated the loss among the predictions of the perturbed and the non- perturbed input just before the Non-max suppression stems. This trick works well in practice, as we will see from the results.

4.1 Model descriptions

The paper predominantly used the Pixel RDE and CartoonX methodologies, where PixelRDE was used as a baseline methodology. Two pre-trained models were also used to assess the explanation capabilities of the two methodologies. These were the VGG-16 [9] and MobileNetv3-small [10] models pre-trained on ImageNet, obtained via PyTorch. The CartoonX method implemented within the original paper stems from the contributions made by [2] and [11] via rate distortion explanations (RDE). This RDE methodology was then applied not in the pixel domain but in the Wavelet basis of the images to use the more relevant, smooth portions of the signal.

4.2 Datasets

The original paper considers random ImageNet [12] samples as its data for producing the results. Specifically, they computed explanations for 100 random ImageNet samples for comparison between different methods. To make the comparison fair, they resize the images to 256×256 before passing them to models.

4.3 Hyperparameters

The original paper and we use the same hyperparameter setting. We use Adam optimizer, a learning rate of $\epsilon = 0.001$, $L = 64$ adaptive Gaussian noise samples, and $N = 2000$ steps. We specify the sparsity level in terms of the number of mask entries k, i.e., by choosing the product λk. For CartoonX, we consider $\lambda k \in [20, 80]$, whereas we consider $\lambda k \in [3, 20]$ for Pixel RDE, as it typically requires a smaller sparsity level than CartoonX. When using the CartoonX methodology, the Daubechies 3 (db3) method was the baseline wavelet system, with a discrete wavelet transform scale of 5.

4.4 Computational requirements

The experiments were conducted using several types of CUDA-based GPUs, namely the RTX 3060 and the GTX 2060Ti. The qualitative results ran on the RTX 3060 consisted of the results in Figure 1, Figure 2, Figure 3, Figure 4, Figure 6, and all of Appendix A. The quantitative results executed on the RTX 3060 shown in Figure 6 required the most computation time to be allotted, totaling 36.25 hours.

4.5 Reproduction Setup

Our experimental setup is closely based on the original paper. For verifying the quantitive analysis in the paper, we show three plots (i) between distortion and non-randomized relevant components and (ii) between distortion and randomized relevant components. (iii) between distortion and non-sparsity. We also plot an accuracy vs. non-randomized relevant component to understand the first plot better.

4.6 Extension Setup

Obfuscation Deviation Scaling — As an extension to the original paper, a simple improvement to the obfuscation method was devised. This improvement took the original standard deviations of the individual wavelets derived from an image. It scaled them according to their resolution, where the scaling was done linearly, with the lowest wavelet being weighted the highest and the highest being weighted the lowest. This scaling depended on the number of wavelet components used within the Daubechies (db3) method. The equation $\sigma_{i_{new}} = \frac{i}{J}\sigma_{i_{old}}$ is used as the scaling for each i wavelet deviation out of the total of J wavelets. The inverse of this was used as well $\sigma_{i_{new}} = \frac{J-i}{J}\sigma_{i_{old}}$. This scaling was also applied inversely and exponentially to determine a method with the lowest amount of distortion while still producing qualitatively good results. The equation $\sigma_{i_{new}} = 2^{\frac{-i}{J}}\sigma_{i_{old}}$ was used for the inverse exponential, where J denoted the number of wavelets, and i is the index of a particular wavelet, starting with 1.

Application on Semantic Segmentation and Object Detection — We use a pretrained DeepLabV3 model with a ResNet-50 backbone to experiment with Semantic Segmentation and a pretrained YOLO-V5 network for Object detection.

5 Results

The qualitative results aligned with the initial paper's qualitative results. However, the same cannot be said for the quantitative results generated using the same hyperparameters specified within the paper.

5.1 Reproducibility Study

After completing the reproduction process of this paper, the qualitative results were similar to those achieved by the original paper. This applies to both the CartoonX and PixelRDE implementations, which are displayed in Figure 1 where the original hyperparameters MobileNetv3-small model. Additional results can be found in Appendix A.

Figure 1. Qualitative Results with original hyperparameters

When compared to the qualitative results generated via the original method described within the paper, several similarities are apparent for both the PixelRDE and CartoonX methodologies. For the PixelRDE method, correctly classified image explanations are mostly black, save for some outlines and highlighted key features. In contrast, the outline remains clear for the CartoonX explanation method, as do key defining features. Qualitative results for misclassified images were also generated, shown in Figure 2. This

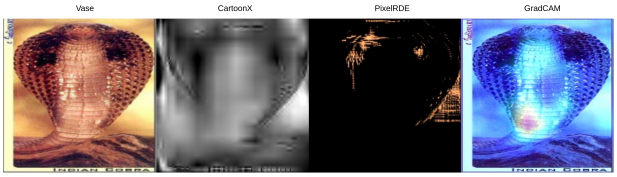

Figure 2. Image of an Indian Cobra misclassified as a vase.

methodology's qualitative results were also explored using the VGG16 model, resulting in Figure 3.

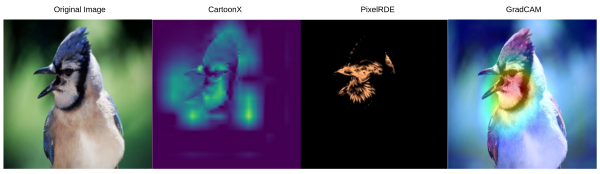

Figure 3. Images resulting from the use of the VGG16 model.

The results from the qualitative sensitivity analysis are displayed in Figure 4. The sensitivity analysis used the hyperparameters specified in both the figures and the text provided. No additional tweaking was required to complete these experiments, except for a minor addition to what is provided in the script, as the weighted-L2 distortion measure required.

When reproducing the quantitative analysis, it was uncovered that several quantitative plots used hyperparameters that were not properly included. The initial assumption was to use 2000 steps, as this was the specified number of steps for both the PixelRDE and CartoonX methodologies. However, this was deemed an unfair assessment, as some methods used 100 steps. One hundred steps were used to compare all methodologies due to this reasoning. The results of this experiment were similar to those produced by the authors and can be seen in Figure 5.

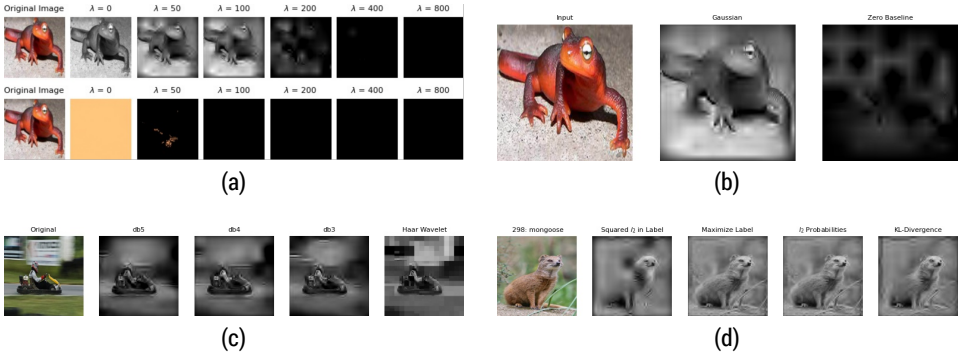

Figure 4. Qualitative Sensitivity Analysis

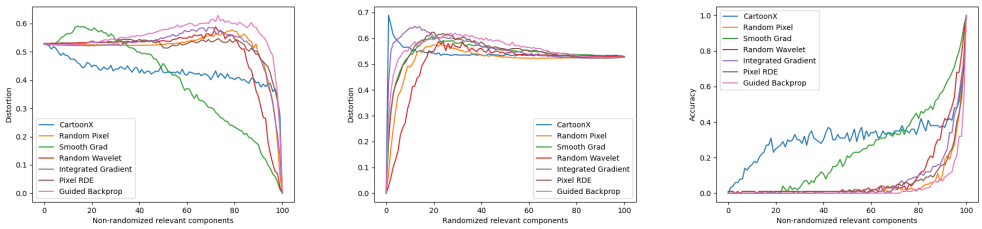

Figure 5. Quantitative results achieved with 100 steps per image instead of the specified 2000.

When attempting to reproduce the figure detailing the changing distortions throughout several different λ values, it was also noted that the plot differed from the original. A similar methodology was applied regarding these λ values as was done for the distortion plots, leading to the plots for 100, 500, and 2000 steps seen in Figure 6. The difference in plots can be attributed to an unclear specification of hyperparameters.

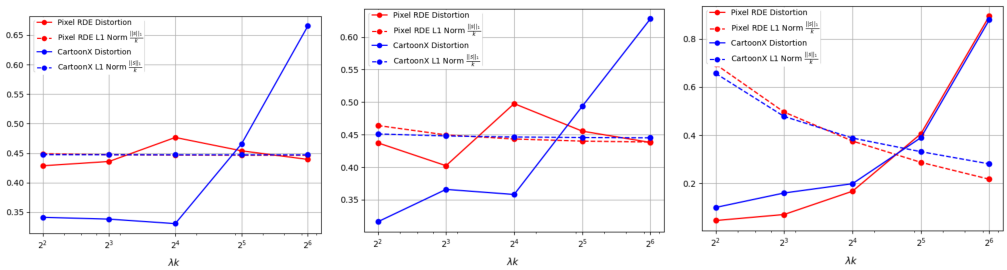

Figure 6. Quantitative results achieved with 100 steps per image, 500 steps and 2000 steps.

5.2 Results beyond original paper

As an extension to the original paper, we conducted several experiments with the goal of improving the methodology used within the paper, as well as examining the robustness of the original methodology. As an addition to the original methodology, improvements were made to the obfuscation methodology in the form of a wavelet distribution scaling method. We further investigate whether CartoonX is model and task-agnostic.

Obfuscation Distribution Scaling — This method was selected as an extension to the original paper as a way of investigating the importance of certain wavelet distributions used during the creation of the obfuscation. The goal was to exploit the different resolu-

tions for each wavelet in a manner that would produce less distorted explanations, while still maintaining the overall qualitative results. Ultimately, all three methods examined proved to be improvements over the original obfuscation methodology, on average. The results of this experiment are provided in Table 1 quantitatively and Figure 7 qualitatively.

	Original	Linear	Exponential	Inverse Linear
Average Distortion	0.01370	**0.006538**	0.008475	0.007605
Deviation	0.01032	0.004349	0.004960	0.004101
Average Improvement per Image	-	0.007164	0.005227	0.006097
Improvement per Image Deviation	-	0.007286	0.007212	0.008652
% Improvement	-	**46.92%**	32.76%	36.34%

Table 1. Values found during the obfuscation modification experiments.

Figure 7. Obfuscation experimentation qualitative results for the image of a goldfish.

This shows that not only was there a reduction in the average distortion with the use of the modified obfuscation method, but there was also a reduction in the overall standard deviation of these distortions relative to the average distortions across all experiments presented. This means that this slight alteration not only improves the overall distortion values, but also produces distortions that are more consistent across different images.

Applications of CartoonX — We use CartoonX to visualize the explanations for several applications namely Semantic Segmentation and object detection.From this we can claim that perhaps CartoonX is task agnostic. One thing common among all these applications is that they are end to end differentiable in nature which allows CartoonX to select the wavelet mask.

Semantic Segmentation Using CartoonX we can visualize the explanations specific to the segmentation of a particular class. This allows us to make sure the segmentation network pays attention to right object of the image to predict the segmentation map. From Figure 8 we can see how cartoonX is able to focus on objects of the same class while blurring the others. That is inorder to predict the segmentation of the cow it blurs the cars and viceversa.

Object Detection Again we qualitatively evaluate CartoonX on the task of Object detection via YOLO [13]. From Figure 9 we can see that CartoonX does well in focusing on the objects of relevance while blurring the background. In the figure CartoonX explanation focuses exactly on the Eagle while blurring the field it is present in. This is as expected because the background is not essential for YOLO to detect the object of interest.

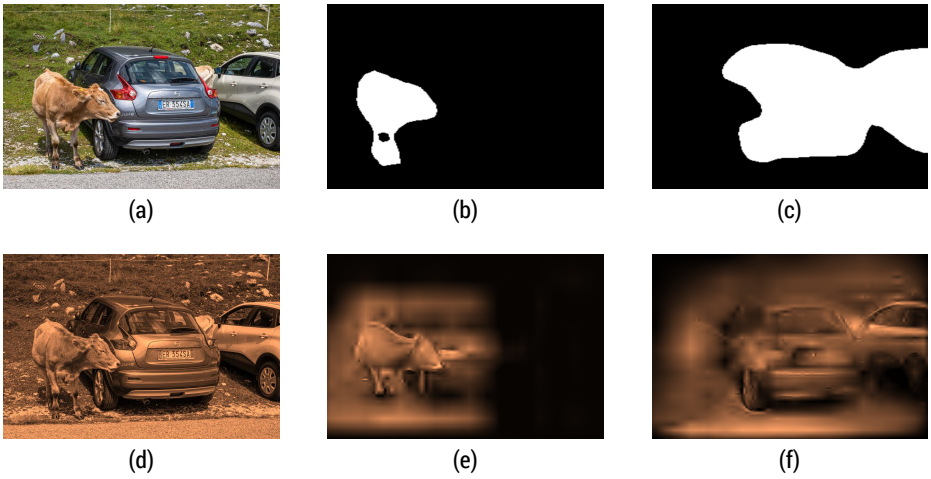

Figure 8. CartoonX on Image segmentation.

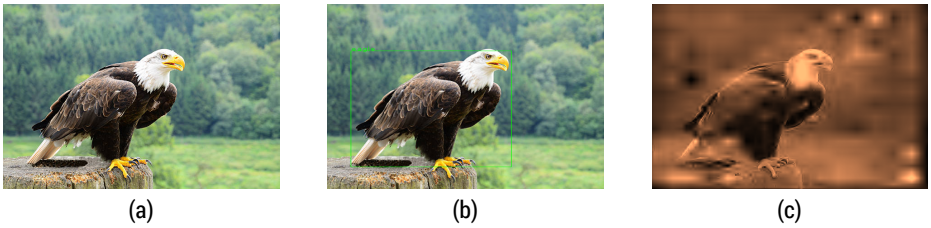

Figure 9. CartoonX on Object Detection.

Effect of Network Architecture on the explainations — As another extension to the original paper, we test CartoonX for different model architectures. We plot the rate-distortion curve by keeping the most relevant coefficients and randomizing the others. The original paper does this for the MobileNetV3 pre-trained on the ImageNet dataset. We extend this for ResNet-18 [14], ResNet-50[14], and ViT-B/16[15] models. Consecutively, we use the pre-trained models on ImageNet and take the average results for 100 images. Figure 10 shows the results for rate-distortion curves.

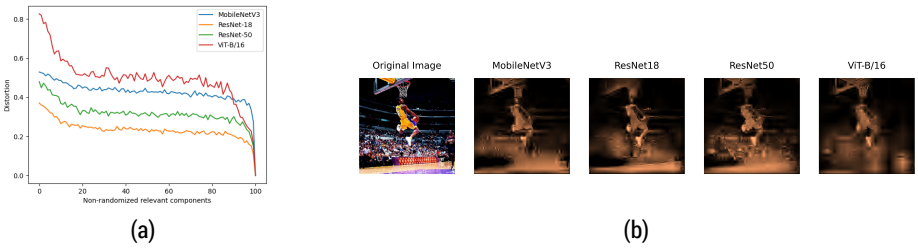

Figure 10. (a) Rate-distortion curve for different convolutional and vision transformer architectures. (b) CartoonX explanations on different model architectures for "basketball" class.

We see that different architectures have different rate-distortion values at the beginning, but ResNet and ViT architectures decay faster than MobileNetV3. ViT starts with the most rate-distortion value but also decays fastest. Figure 10 further shows the difference in CartoonX representations for models. For example, for ViT the "basket" is more

highlighted, suggesting a possible key point that could show why a model performs better at certain tasks such as classification.

6 Discussion

In order to conduct a proper reproduction of the paper by Kolek et al. we executed several qualitative and quantitative experiments with the goal of also extending on the original work. While the majority of experiments conducted validated the overall reproducibility of the paper, some figures produced results that were not quite as similar to those produced by the original authors initially. Mainly, when conducting the quantitative analysis, we found the only after changing the number of iterations used to create the non-randomized and randomized relevant component distortion plots shown in Figure 5 were the results actually comparable to those produced in the original. Additionally, although the λ experimentation produced plots with similar trends as those shown in the original, the results produced, especially those for the PixelRDE, were in fact not similar. This may have been due to the hyperparameters not being clearly specified for this plot .

Thanks to the code being provided not only in a very structured manner but also with clear instructions, the initial qualitative results were produced quickly and without any additional scripting.

6.1 What was easy

Overall, the initial qualitative result process was easy due to the initial layout of both the repository and the procedures that were presented within the paper. The ease of this process can be specifically attributed to the organization of the repository making it easily navigable, as well as the documentation also provided within the repository, detailing a step-by-step process on installing all dependencies and setting up the code for an initial qualitative run. This led to our ability to add multiple extensions onto the original paper, ultimately adding a layer of robustness to the claims made and the work done by the original authors.

6.2 What was difficult

Some important hyperparameters required to reproduce the quatitative results were missing. The code for multiple baseline methods used for comparison against CartoonX was not provided in the codebase so we had to code them on our own.And also all the hyperparameters for the baselines were not clear making fair comparison a bit difficult.

6.3 Communication with original authors

In order to clarify some points within the quantitative results, some questions were posed to the authors regarding the λ values used within the quantitative results. while this did help to clarify some hyperparameters used such as the λ values, other questions arose later in the process that were unfortunately left unanswered due to the short timeline provided. One question we did manage to answer independently was how many steps were used to produce some of the quantitative results, specifically those pertaining to the amount of distortion when using a certain portion of randomized and non-randomized components. We found the plots for 100 steps more closely resembling those in the original paper, even though we were unsure about using 2000 steps or 100 steps.

References

1. S. Kolek, D. A. Nguyen, R. Levie, J. Bruna, and G. Kutyniok. "Cartoon explanations of image classifiers." In: **Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XII**. Springer. 2022, pp. 443–458.
2. J. MacDonald, S. Wäldchen, S. Hauch, and G. Kutyniok. "A rate-distortion framework for explaining neural network decisions." In: **arXiv preprint arXiv:1905.11092** (2019).
3. R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. "Grad-cam: Visual explanations from deep networks via gradient-based localization." In: **Proceedings of the IEEE international conference on computer vision**. 2017, pp. 618–626.
4. M. Sundararajan, A. Taly, and Q. Yan. "Axiomatic attribution for deep networks." In: **International conference on machine learning**. PMLR. 2017, pp. 3319–3328.
5. D. Smilkov, N. Thorat, B. Kim, F. Viégas, and M. Wattenberg. "Smoothgrad: removing noise by adding noise." In: **arXiv preprint arXiv:1706.03825** (2017).
6. J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller. "Striving for simplicity: The all convolutional net." In: **arXiv preprint arXiv:1412.6806** (2014).
7. S. Lapuschkin, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek. "On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation." In: **PLoS ONE** 10 (July 2015), e0130140.
8. M. T. Ribeiro, S. Singh, and C. Guestrin. "" Why should i trust you?" Explaining the predictions of any classifier." In: **Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining**. 2016, pp. 1135–1144.
9. K. Simonyan and A. Zisserman. "Very deep convolutional networks for large-scale image recognition." In: **arXiv preprint arXiv:1409.1556** (2014).
10. A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, et al. "Searching for mobilenetv3." In: **Proceedings of the IEEE/CVF international conference on computer vision**. 2019, pp. 1314–1324.
11. C. Heiß, R. Levie, C. Resnick, G. Kutyniok, and J. Bruna. "In-distribution interpretability for challenging modalities." In: **arXiv preprint arXiv:2007.00758** (2020).
12. J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. "Imagenet: A large-scale hierarchical image database." In: **2009 IEEE conference on computer vision and pattern recognition**. Ieee. 2009, pp. 248–255.
13. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. "You only look once: Unified, real-time object detection." In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. 2016, pp. 779–788.
14. K. He, X. Zhang, S. Ren, and J. Sun. "Deep residual learning for image recognition." In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. 2016, pp. 770–778.
15. A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. "An image is worth 16x16 words: Transformers for image recognition at scale." In: **arXiv preprint arXiv:2010.11929** (2020).

A Qualitative Results

This section shows more qualitative results generated via an initial run of the code provided with the original hyperparameters as well. It is divided into results generated via correctly classified images, and results generated via incorrectly classified images.

A.1 Correctly Classified Explanations

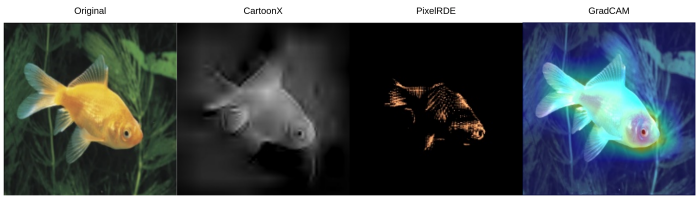

Figure 11. Qualitative results using the goldfish example image.

Figure 12. Qualitative results using the jay example image.

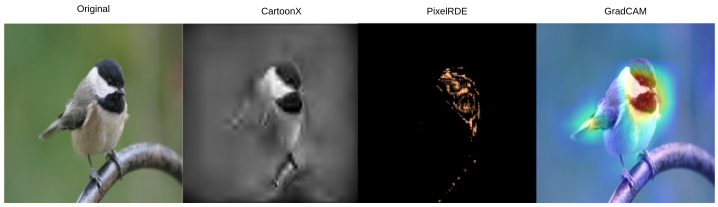

Figure 13. Qualitative results using the chickadee example image.

A.2 Incorrectly Classified Images

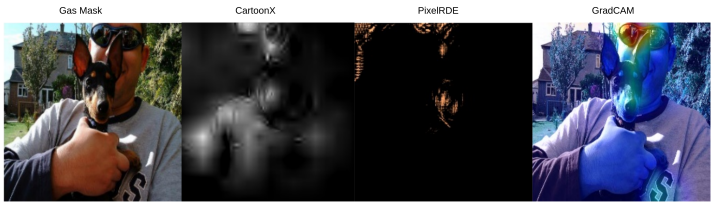

Figure 14. Qualitative results using the terrier example image.

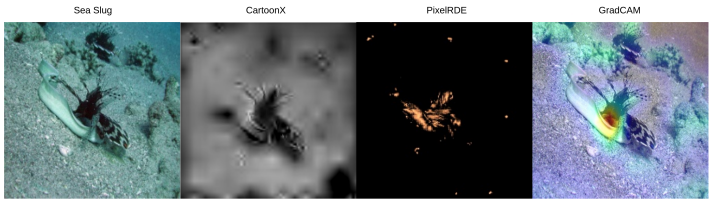

Figure 15. Qualitative results using the lionfish example image.

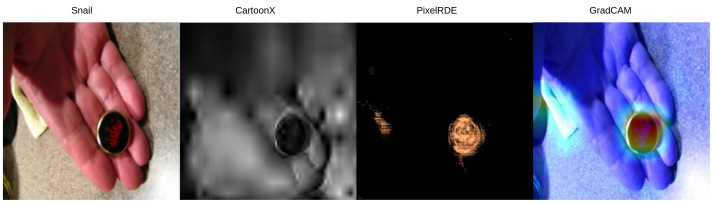

Figure 16. Qualitative results using the bottlecap example image.

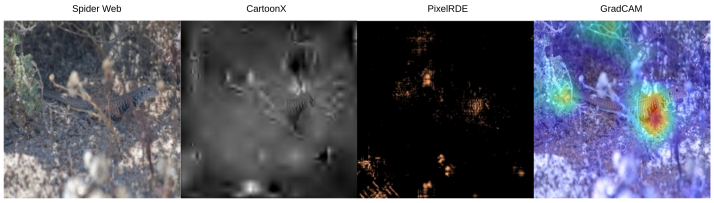

Figure 17. Qualitative results using the lizard example image.

A.3 Additional VGG-16 Images

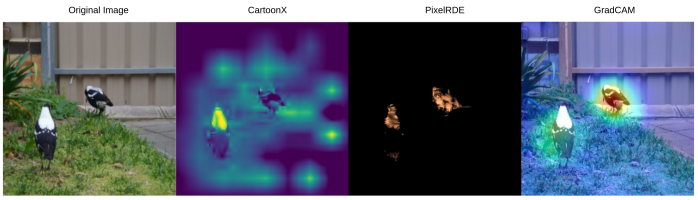

Figure 18. PixelRDE and CartoonX outputs when using the VGG16 model.

A.4 Obfuscation Distribution Scaling Example Images

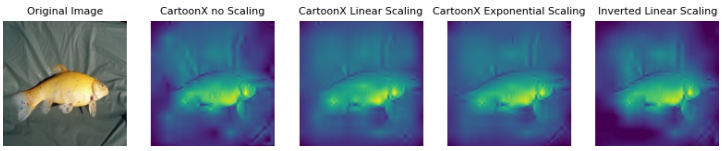

Figure 19. Obfuscation experimentation qualitative results for the image of a tench.

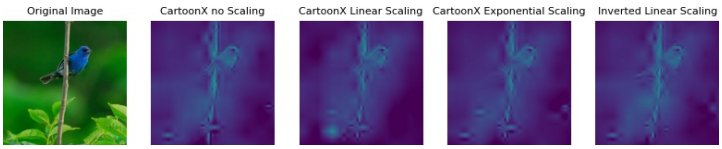

Figure 20. Obfuscation experimentation qualitative results for the image of an indigo bird.

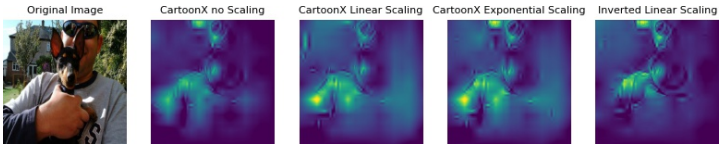

Figure 21. Obfuscation experimentation qualitative results for the image of a terrier.