SUNY: A Visual Interpretation Framework for Convolutional Neural Networks from a Necessary and Sufficient Perspective

Xiwei Xuan[1,*] Ziquan Deng[1] Hsuan-Tien Lin[2] Zhaodan Kong[1] Kwan-Liu Ma[1]
[1]University of California, Davis [2]National Taiwan University *Corresponding Author
{xwxuan, ziqdeng}@ucdavis.edu htlin@csie.ntu.edu.tw {zdkong, klma}@ucdavis.edu

Abstract

In spite of the ongoing evolution of deep learning, Convolutional Neural Networks (CNNs) remain the de facto choice for numerous vision applications. To foster trust, researchers have proposed various methods for visually interpreting CNNs via heatmaps, which highlight the input regions important to a specific model decision. However, in terms of the underlying design logic, existing approaches often concentrate on model parameters, overlooking the fundamental "why" question integral to human cognition. Thus they fail to embrace the two critical and complementary sides in reasoning: necessity and sufficiency. To address these issues, we introduce SUNY, a framework designed to rationalize the explanations toward better human understanding from both necessary and sufficient perspectives in a bi-directional manner. Extensive evaluations justify that SUNY not only yields more informative and convincing explanations from both angles, but also achieves performances competitive to other approaches across different CNN architectures over different datasets.

1. Introduction

Despite the unprecedented strides in deep learning, the interpretation of Convolutional Neural Networks (CNNs) continues to be an essential field of study, attributed to their pervasive application [32, 36], proven robustness [2, 17, 28], and inherently opaque nature [9]. This paper addresses the eXplainable Artificial Intelligence (XAI) [10] problem corresponding to CNN for natural image classification, i.e., reasoning why a classifier makes particular decisions. Specifically, we study visual explanation techniques that present heatmaps highlighting image portions associated with a model's class prediction. A series of gradient-weighted CAMs [3, 15, 20] in the CAM [37] family are widely-adopted in applications. However, gradients' saturation and vanishing issues can lead to noise explanations for such CAMs [7]. To bypass the shortcomings of gradients, Score-CAM [26] and Group-CAM [35] weight feature maps by contribution scores, referring to the corresponding input features' importance to the model output. Reflecting on these methods, the design of Score-CAM and Group-CAM, which measures model's prediction (outcome) by retaining specific input features (cause), aligns with the concept of causal *sufficiency (S)*. Conversely, the design principle behind perturbation-based techniques [16, 18, 34], measuring model's prediction (outcome) when changing input features (cause), aligns with the idea of causal *necessity (N)*. In general, *N* involves changing hypothetical causes and measuring the resultant differences in outcomes, while *S* investigates whether preserving specific causes can maintain the outcome by quantifying outcome stability. Many studies underscore the cruciality of both *N* and *S* as *"two desirable and essential perspectives for a successful explanation"*, as they resonate with the bi-directional counterfactual thinking intrinsic to human cognition [8, 12, 29, 31].

Given the significance of these two facets, we propose an explanation framework called **SU**fficiency and **N**ecessit**Y** explanation (*SUNY*), which interprets the CNN classifier by regarding the input features as the hypothesized causes and quantifying each cause's importance towards the class prediction from angles of both *N* and *S*. Based on the qualitative evaluation, including the semantic evaluation and the sanity check, we demonstrate that *SUNY* provides a more faithful and interpretable visual explanation for CNN models. Comprehensive experiment evaluations on benchmark datasets, including ILSVRC2012 [19] and CUB-200-2011 [30], validate that *SUNY* outperforms other popular heatmap-based visual explanation methods.

2. The Proposed Approach

Our method aims to (1) measure the importance of each individual cause in a group of coordinating causes ($G1$), and (2) quantifying their actual impact on the outcome while considering both *N* and *S* ($G2$).

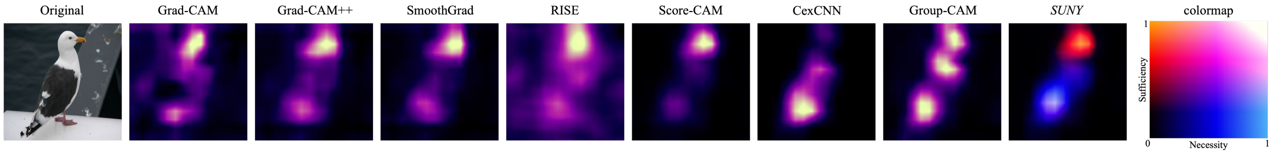

Figure 1. Visual comparison of heatmap explanations provided by different methods. All heatmaps in this paper follow the colormap on the right (X-axis: *Necessity*; Y-axis: *Sufficiency*; $y = x$: *Importance* (applicable to methods with one-dimensional information, "importance").

2.1. Bi-directional Importance Quantification

The Shapley value's framework for assessing marginal contributions across various coalitions lays the groundwork for achieving $G1$. Additionally, we further define the necessity and sufficiency value functions to accomplish $G2$. However, previous SHAP [13] image analyses segment input images into equally-sized patches, restricting finer distinctions. Our model-integrated method regards a single feature map (or a set of feature maps) as a cause f_i (or a set of causes F_*), thereby providing more granular explanations. We utilize a general formulation of Shapley values in the following definition.

For a set of causes (i.e., a set of feature maps) to be analyzed as F_*, we define **N** value function to measure the degree of the outcome change when removing F_*:

$$E_N(F_*) = [p_c(I) - p_c(\text{do } (F \setminus F_*))]/p_c(I), \quad (1)$$

where $\text{do}(F \setminus F_*)$ represents the intervention of removing F_*. And $p_c(\bullet)$ refers to the model's prediction probability w.r.t. a target class c, where $p_c(I)$ is its original value without any intervention; $p_c(\text{do}(F \setminus F_*))$ represents the value after the removal intervention. Similarly, **S** value function is defined as:

$$E_S(F_*) = p_c(\text{do } (F_*))/p_c(I), \quad (2)$$

where $\text{do}(F_*)$ represents the intervention of only keeping F_*, i.e., removing $\{F \setminus F_*\}$.

Different from covering all elements, we tend to focus on more necessary (sufficient) ones. $\forall f_i \in F$, where f_i is a single cause, we set $F_* = \{f_i\}$ to calculate $E_N(f_i)$ ($E_S(f_i)$) and construct a set $F_N \subseteq F$ ($F_S \subseteq F$) by combining the relatively more necessary (*sufficiency*) f_i. Then to analyze a single cause $f_n \in F_N$, we calculate the N Shapley value as:

$$R_N(f_n) = \sum_{F' \subseteq \{F_N \setminus f_n\}} \frac{|F'|!(|F_N| - |F'| - 1)!}{|F_N|!} \times [E_N(F' \cup f_n) - E_N(F')]. \quad (3)$$

Similarly, we can calculate S Shapley value for $f_s \in F_S$ as:

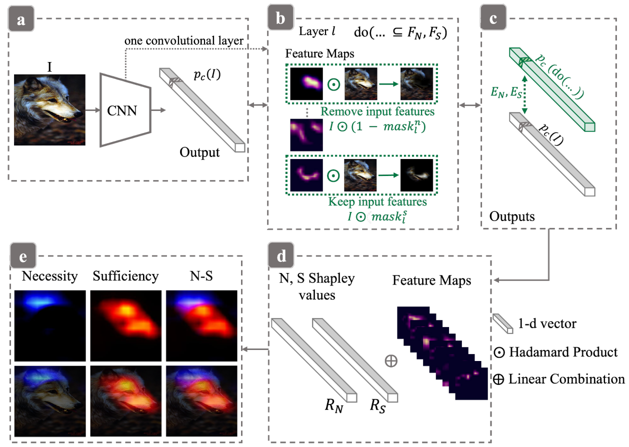

Figure 2. Overview of *SUNY* framework. Phase a is a forward pass of input image I through a CNN model. In Phase b, we obtain feature maps of a specified layer, and intervene on model filters or the corresponding input features. We get new prediction probabilities after the intervention and calculate E_N, E_S in Phase c, which are fed back to Phase b to construct hypothesized cause sets F_N and F_S. Through intervening on coalitions in F_N and F_S (Phase b), we can obtain their E_N, E_S (Phase c) and R_N and R_S (Phase d). The saliency maps are generated by linearly combining feature maps and examples of *SUNY* results are shown in Phase e.

$$R_S(f_s) = \sum_{F' \subseteq \{F_S \setminus f_s\}} \frac{|F'|!(|F_S| - |F'| - 1)!}{|F_S|!} \times [E_S(F' \cup f_s) - E_S(F')]. \quad (4)$$

In the implementation, we reduce the amount of computation by estimating Eqns.(3), (4) using Shapley sampling values method [24]. Additionally, for $f'_n \in \{F \setminus F_N\}$ and $f'_s \in \{F \setminus F_S\}$, we set $R_N(f'_n) = 0$ and $R_S(f'_s) = 0$.

2.2. *SUNY* Implementation

Fig. 2 presents the *SUNY* framework applied to explain a CNN classifier w.r.t. the predicted class of an input image (a). Aligned with existing approaches ([3, 20, 22, 26, 35]), we consider feature maps of a convolutional layer as feature extractors to support our intervention. Specifically, each feature map can be upsampled into the image size as a weighted mask, including values in the range of [0, 1]. Utilizing them, *SUNY* can remove/keep corresponding input regions for intervention, providing explanations for any convolutional layer. We first obtain its feature maps by forwarding an image into the model (Phase b). Next, we calculate E_N and E_S (refer to Eqns. (1)(2)) by intervening on every single cause, which is conducted by masking out specific input features ($I \bigodot (1- mask_l^n)$; $I \bigodot mask_l^s$) (Phase b c). We then construct F_N and F_S by combining single causes with higher E_N and E_S, respectively, and repeat the aforementioned intervention operations on F_N and F_S (Phase b c). Finally, we calculate importance scores R_N and R_S based on Eqns. (3)(4)) and generate final visualizations (Phase d e).

Note that the general definition in Sec.2.1 is not limited to feature maps as causes. In Appendix, we detail the analysis when considering model filters as alternative causes.

3. Experiments

This section compares *SUNY*'s effectiveness with established visual explanation benchmarks.

3.1. Experimental Setup

Baseline Methods. To evaluate the effectiveness of our proposed method in pinpointing the crucial region for models' decisions, we carefully chose relevant methods for a comparative analysis with *SUNY*. The criteria for selection were twofold. First, these methods must possess class-discriminative capabilities, meaning they should provide unique explanations for different specified classes. Second, they should generate heatmaps that localize input regions relevant to the model's decision. Our selection encompasses seven methods that satisfy these requirements, covering a broad spectrum of approaches predominant in CNN visual explanations. These include gradient-based methods (Grad-CAM [20], Grad-CAM++ [3], and SmoothGrad [22]), a score-based method (Score-CAM [26]), causality-driven methods (CexCNN [5], and Group-CAM [35]), and a perturbation-based method (RISE [16]). Notably, we exclude pixel-space gradient visualizations such as Guided Backpropagation [23] or LRP [14] from our comparison due to their lack of class-discriminative ability, ensuring our baseline selection is aligned with our objectives.

Datasets and CNN Models. The experiments involve two datasets, ILSVRC2012 (ILSVRC) [19] with 1000 classes and 50k images and CUB-200-2011 (CUB) [30] with 200 classes and 5794 images. We use all explanation methods to explain three CNN models with different architectures, including VGG16 [21], Inception-v3 [25], and ResNet50 [11].

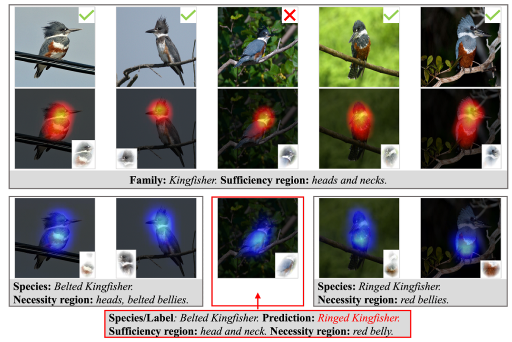

Figure 3. *SUNY* for a VGG16 trained on CUB. The first row displays bird images from four species across two families, with correct and incorrect predictions marked by ✔ and ✖. Misclassifications occur within the same family. The second and third rows show *sufficiency* and *necessity* heatmaps, with a small image in each heatmap's bottom corner highlighting the image portion.

3.2. Qualitative Evaluation

In Fig. 1, we visually compare *SUNY* with other explanation approaches and observe two advantages: (1) Saliency maps provided by *SUNY* contain fewer noises. (2) *SUNY* uniquely provides both *necessary* and *sufficient* information to support interpretation. For example, in Fig. 1, *SUNY* tells that the bottom wing is *necessary* and the head is *sufficient* for `Gull` prediction.

***SUNY* explanations for failure cases.** Fig. 3 presents CUB images of two bird species (i.e., `belted kingfisher` and `ringed kingfisher`) of one family (i.e., `kingfisher`). The *sufficiency* heatmaps in the second row reveal family-specific features: all `kingfisher` display characteristic heads and necks. This explains why the model correctly identifies the bird family for every image. The *necessity* heatmaps in the third row provide additional insights to distinguish between different species within the same family: the belted bellies are highlighted in images predicted as `belted kingfisher`, while the red bellies are highlighted in the images predicted as `ringed kingfisher`. This explains why the third image is mistaken – the red belly is easily observable through this view and is identical to a `ringed kingfisher`. Examples in Fig. 3 demonstrate that *sufficiency* and *necessity* provide semantically-complementary explanations to better support model behavior interpretation.

3.3. Deletion and Insertion Evaluation

We evaluated model prediction relevancy to highlighted regions using *deletion* and *insertion* experiments, following [16]. *Deletion* gauges prediction impact when key pixels are removed, while *insertion* tracks prediction changes as pixels are added back in order of importance. Using Area Under the Curve (AUC) for quantification [16, 35], superior

Dataset	Methods	VGG 16			Inception-v3			ResNet50		
		Deletion ↓	Insertion ↑	Overall ↑	Deletion ↓	Insertion ↑	Overall ↑	Deletion ↓	Insertion ↑	Overall ↑
ILSVRC	Grad-CAM[20]	0.1098	0.6112	0.5015	0.1276	0.6567	0.5291	0.1796	0.6889	0.5093
	Grad-CAM++[3]	0.1155	0.6033	0.4878	0.1309	0.6476	0.5167	0.1847	0.6799	0.4952
	SmoothGrad [22]	0.1136	0.6023	0.4887	0.1317	0.6465	0.5148	0.1849	0.6800	0.4951
	RISE [16]	0.1185	0.6188	0.5003	0.1404	0.6444	0.5040	0.1303	0.6932	0.5629
	Score-CAM [26]	0.1070	0.6382	0.5312	0.1309	0.6528	0.5219	0.2319	0.6218	0.3898
	CexCNN [5]	0.1161	0.6025	0.4864	0.1355	0.6543	0.5188	0.1886	0.6443	0.4557
	Group-CAM [35]	0.1138	0.6218	0.5080	0.1292	0.6545	0.5253	0.1794	0.6904	0.5110
	SUNY	0.1005	0.6468	0.5462	0.1215	0.6603	0.5388	0.1323	0.6988	0.5665
	SUNY-N	0.1057	0.6038	0.4981	0.1257	0.6453	0.5196	0.1374	0.6552	0.5178
	SUNY-S	0.1144	0.6389	0.5245	0.1309	0.6530	0.5221	0.2220	0.6922	0.4702
CUB	Grad-CAM[20]	0.0558	0.7617	0.7059	0.0963	0.7323	0.6360	0.0930	0.6452	0.5522
	Grad-CAM++[3]	0.0589	0.7541	0.6951	0.0950	0.7281	0.6331	0.0972	0.6407	0.5434
	SmoothGrad [22]	0.0594	0.7489	0.6895	0.0977	0.7244	0.6266	0.0974	0.6405	0.5431
	RISE [16]	0.0560	0.7583	0.7023	0.0855	0.7168	0.6314	0.0570	0.6567	0.5996
	Score-CAM[26]	0.0542	0.7575	0.7033	0.0901	0.7326	0.6424	0.0995	0.6351	0.5355
	CexCNN [5]	0.0630	0.7389	0.6760	0.1017	0.7283	0.6267	0.1014	0.6173	0.5159
	Group-CAM [35]	0.0606	0.7521	0.6915	0.0971	0.7290	0.6318	0.0926	0.6458	0.5532
	SUNY	0.0518	0.7591	0.7073	0.0842	0.7361	0.6519	0.0562	0.6645	0.6083
	SUNY-N	0.0537	0.7497	0.6960	0.0854	0.7165	0.6311	0.0667	0.6443	0.5776
	SUNY-S	0.0555	0.7577	0.7022	0.0894	0.7328	0.6434	0.0939	0.6577	0.5638

Table 1. Comparative evaluation between *SUNY* and baselines w.r.t. the *deletion*, *insertion*, and *overall* AUC, where lower *deletion*, higher *insertion*, and higher *overall* indicate a better explanation. The first and second best performances are marked in green and blue, respectively. *SUNY-N* and *SUNY-S* are not included for performance ranking.

model explanations are reflected by lower *deletion*, higher *insertion* and higher *overall (insertion-deletion)* scores. As shown in Table 1, *SUNY* consistently equals or exceeds baselines in most comparisons.

3.4. Saliency Attack

Researchers have proposed a series of local adversarial attack approaches [4, 6, 27, 33] guided by saliency maps, which is to fool CNN models by perturbing a small image region highlighted by saliency maps. These methods require the saliency maps to be "minimal and essential" [4]. Inspired by these insights, we propose an evaluation metric, $Attack_{score} = \frac{FlipRate}{AvgAttackSize}$, to validate whether *SUNY* explanations can detect the most important regions w.r.t. the model's decision. After applying Gaussian noise to the saliency regions, we check any decision changes: $Flip = 1$ if $argmax(p(I)) \neq argmax(p(I'))$. To validate whether the region is "minimal", we include $AvgAttackSize$, which is the average size of all saliency maps. The results reported in Table 2 proves that *SUNY* are better at highlighting the most important image region corresponding to the model's decision.

Dataset	Methods	Saliency $Attack_{score}$ ↑		
		VGG16	Inception-v3	ResNet50
ILSVRC	Grad-CAM[20]	0.9615	1.0435	0.7674
	Grad-CAM++[3]	0.9991	0.9821	0.8751
	SmoothGrad[22]	1.0449	0.9675	0.8776
	RISE[16]	0.9928	0.7353	1.0259
	Score-CAM[26]	0.5326	0.9673	0.3378
	CexCNN[5]	1.6341	1.0653	0.6393
	Group-CAM[35]	1.1556	1.0200	0.8020
	SUNY	2.0452	1.9874	1.5619
	SUNY-N	1.6344	1.0885	1.0726
	SUNY-S	0.5434	0.9685	0.5564
CUB	GradCam[20]	0.5969	0.5985	0.4694
	GradCam++[3]	0.6670	0.5950	0.5257
	SmoothGrad[22]	0.7783	0.5929	0.5260
	RISE [16]	0.5063	0.3860	1.1286
	Score-CAM[26]	1.2215	0.5989	0.8027
	CexCNN[5]	1.2673	0.5898	0.4171
	Group-CAM[35]	0.6742	0.5951	0.4991
	SUNY	2.8111	1.0475	1.7747
	SUNY-N	1.5863	0.7658	1.2083
	SUNY-S	1.2317	0.5884	0.8238

Table 2. Comparative evaluation between *SUNY* and baselines w.r.t. saliency attack scores (higher is better). (First and second best performances. *SUNY-N* and *SUNY-S* are not included for performance ranking.)

3.5. Localization Evaluation

This section evaluates saliency map localization using the energy-based pointing game [26], aiming to measure the localization ability of saliency maps using the ground-truth bounding box of the target class, $bbox$. The input image is binarized with $bbox$ by assigning the inside and outside regions with 1 and 0, respectively. Then, we apply the Hadamard product between the binarized input and the saliency map, the summary of which can quantify how much "energy" falls into $bbox$. The performance is measured by $Proportion = \frac{\sum map[i,j]_{(i,j)\in bbox}}{\sum map[i,j]}$. Table 3 reveals that *SUNY*, by simultaneously leveraging **N** and **S** features, outperforms other methods in terms of localization ability.

3.6. Sanity Check

The sanity check [1] verifies if a visual explanation method reliably reflects the model's behavior. We conduct cascading randomization to the model's weights from the top to

Dataset	Methods	Proportion (%) ↑		
		VGG16	Inception-v3	ResNet50
ILSVRC	Grad-CAM[20]	57.68	66.35	59.84
	Grad-CAM++[3]	61.31	65.93	61.74
	SmoothGrad[22]	62.18	65.78	61.75
	RISE[16]	58.93	59.26	59.48
	Score-CAM[26]	64.25	65.94	66.72
	CexCNN[5]	65.24	66.33	57.39
	Group-CAM[35]	62.70	66.17	60.68
	SUNY	65.61	66.71	68.02
CUB	Grad-CAM[20]	43.06	40.05	39.02
	Grad-CAM++[3]	45.45	40.45	41.25
	SmoothGrad[22]	47.12	40.34	41.28
	RISE[16]	37.28	34.74	36.32
	Score-CAM[26]	49.68	40.67	47.42
	CexCNN[5]	37.13	41.38	41.22
	Group-CAM[35]	43.53	41.08	40.36
	SUNY	49.97	41.96	43.21

Table 3. Comparative evaluation w.r.t. energy-pointing games' proportion (higher is better). (First and second best performances.)

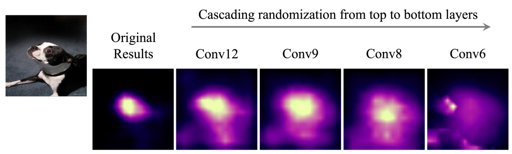

Figure 4. Sanity check of *SUNY*. The first column is the original heatmap visual explanation, and the following columns show results after randomizing specific layers.

the bottom layer successively and generate explanations every time after the randomization. If saliency maps are consistent across models with different parameters, the method does not pass the sanity check. Fig. 4 indicates *SUNY* pass the sanity check.

4. Conclusion

We design *SUNY*, a framework that offers bidirectional visual explanations of CNNs, integrating both *necessity* and *sufficiency* aspects. Qualitative assessments confirm that *SUNY* generates deeper, more meaningful visualizations, illustrating the added value of combining *necessity* and *sufficiency*. Furthermore, *SUNY* also passes the sanity check and quantitatively outperform seven other visual explanation methods in deletion and insertion evaluation, saliency attack, and localization evaluations across multiple CNN architectures and benchmark datasets.

Acknowledgement

This work is supported in part by NIBIB under grant no. P41 EB032840. H.-T. Lin is partially supported by the Ministry of Science and Technology in Taiwan via MOST 111-2918-I-002-006 and 112-2628-E-002-030.

References

[1] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. *arXiv preprint arXiv:1810.03292*, 2018. 4

[2] Yutong Bai, Jieru Mei, Alan L Yuille, and Cihang Xie. Are transformers more robust than cnns? *Advances in Neural Information Processing Systems*, 34:26831–26843, 2021. 1

[3] Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian. Grad-CAM++: Generalized gradient-based visual explanations for deep convolutional networks. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 839–847. IEEE, 2018. 1, 2, 3, 4, 5

[4] Zeyu Dai, Shengcai Liu, Ke Tang, and Qing Li. Saliency attack: Towards imperceptible black-box adversarial attack. *arXiv preprint arXiv:2206.01898*, 2022. 4

[5] Hichem Debbi. Causal explanation of convolutional neural networks. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 633–649. Springer, 2021. 3, 4, 5

[6] Xiaoyi Dong, Jiangfan Han, Dongdong Chen, Jiayang Liu, Huanyu Bian, Zehua Ma, Hongsheng Li, Xiaogang Wang, Weiming Zhang, and Nenghai Yu. Robust superpixel-guided attentional adversarial attack. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12895–12904, 2020. 4

[7] Amirata Ghorbani, Abubakar Abid, and James Zou. Interpretation of neural networks is fragile. In *Proceedings of the AAAI conference on artificial intelligence*, pages 3681–3688, 2019. 1

[8] Guy Grinfeld, David Lagnado, Tobias Gerstenberg, James F Woodward, and Marius Usher. Causal responsibility and robust causation. *Frontiers in Psychology*, 11:1069, 2020. 1

[9] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. A survey of methods for explaining black box models. *ACM computing surveys (CSUR)*, 51(5):1–42, 2018. 1

[10] David Gunning. Explainable artificial intelligence (xai). *Defense Advanced Research Projects Agency (DARPA), nd Web*, 2, 2017. 1

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 3

[12] Peter Lipton. Contrastive explanation. *Royal Institute of Philosophy Supplements*, 27:247–266, 1990. 1

[13] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017. 2

[14] Grégoire Montavon, Alexander Binder, Sebastian Lapuschkin, Wojciech Samek, and Klaus-Robert Müller. Layer-wise relevance propagation: an overview. *Explainable AI: interpreting, explaining and visualizing deep learning*, pages 193–209, 2019. 3

[15] Daniel Omeiza, Skyler Speakman, Celia Cintas, and Komminist Weldermariam. Smooth Grad-CAM++: An enhanced inference level visualization technique for deep

convolutional neural network models. *arXiv preprint arXiv:1908.01224*, 2019. 1

[16] Vitali Petsiuk, Abir Das, and Kate Saenko. Rise: Randomized input sampling for explanation of black-box models. *arXiv preprint arXiv:1806.07421*, 2018. 1, 3, 4, 5

[17] Maithra Raghu, Thomas Unterthiner, Simon Kornblith, Chiyuan Zhang, and Alexey Dosovitskiy. Do vision transformers see like convolutional neural networks? *Advances in Neural Information Processing Systems*, 34:12116–12128, 2021. 1

[18] Marco Tulio Ribeiro, Sameer Singh, and Carlos 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*, pages 1135–1144, 2016. 1

[19] 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 (IJCV)*, 115 (3):211–252, 2015. 1, 3

[20] 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 *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017. 1, 2, 3, 4, 5

[21] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 3

[22] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017. 2, 3, 4, 5

[23] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014. 3

[24] Erik Štrumbelj and Igor Kononenko. Explaining prediction models and individual predictions with feature contributions. *Knowledge and information systems*, 41(3):647–665, 2014. 2

[25] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016. 3

[26] Haofan Wang, Mengnan Du, Fan Yang, and Zijian Zhang. Score-CAM: Improved visual explanations via score-weighted class activation mapping. *arXiv preprint arXiv:1910.01279*, 2019. 1, 2, 3, 4, 5

[27] Jie Wang, Zhaoxia Yin, Jing Jiang, and Yang Du. Attention-guided black-box adversarial attacks with large-scale multi-objective evolutionary optimization. *International Journal of Intelligent Systems*, 2022. 4

[28] Zeyu Wang, Yutong Bai, Yuyin Zhou, and Cihang Xie. Can cnns be more robust than transformers? *arXiv preprint arXiv:2206.03452*, 2022. 1

[29] David S Watson, Limor Gultchin, Ankur Taly, and Luciano Floridi. Local explanations via necessity and sufficiency: Unifying theory and practice. In *Uncertainty in Artificial Intelligence*, pages 1382–1392. PMLR, 2021. 1

[30] Peter Welinder, Steve Branson, Takeshi Mita, Catherine Wah, Florian Schroff, Serge Belongie, and Pietro Perona. Caltech-ucsd birds 200. 2010. 1, 3

[31] James Woodward. Sensitive and insensitive causation. *The Philosophical Review*, 115(1):1–50, 2006. 1

[32] Ting-Wei Wu, Hua Zhang, Wei Peng, Fan Lü, and Pin-Jing He. Applications of convolutional neural networks for intelligent waste identification and recycling: A review. *Resources, Conservation and Recycling*, 190:106813, 2023. 1

[33] Tao Xiang, Hangcheng Liu, Shangwei Guo, Tianwei Zhang, and Xiaofeng Liao. Local black-box adversarial attacks: A query efficient approach. *arXiv preprint arXiv:2101.01032*, 2021. 4

[34] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014. 1

[35] Qinglong Zhang, Lu Rao, and Yubin Yang. Group-cam: group score-weighted visual explanations for deep convolutional networks. *arXiv preprint arXiv:2103.13859*, 2021. 1, 2, 3, 4, 5

[36] Xiaorui Zhang, Jie Zhou, Wei Sun, and Sunil Kumar Jha. A lightweight cnn based on transfer learning for covid-19 diagnosis. *Computers, Materials & Continua*, 72(1), 2022. 1

[37] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016. 1