Rethinking Interpretation: Input-Agnostic Saliency Mapping of Deep Visual Classifiers

Naveed Akhtar, Mohammad Amir Asim Khan Jalwana

The University of Western Australia, 35 Stirling Highway Crawley 6009, Australia.
naveed.akhtar@uwa.edu.au, asimjalwana@gmail.com

Abstract

Saliency methods provide post-hoc model interpretation by attributing input features to the model outputs. Current methods mainly achieve this using a single input sample, thereby failing to answer input-independent inquiries about the model. We also show that input-specific saliency mapping is intrinsically susceptible to misleading feature attribution. Current attempts to use 'general' input features for model interpretation assume access to a dataset containing those features, which biases the interpretation. Addressing the gap, we introduce a new perspective of input-agnostic saliency mapping that computationally estimates the high-level features attributed by the model to its outputs. These features are geometrically correlated, and are computed by accumulating model's gradient information with respect to an unrestricted data distribution. To compute these features, we nudge independent data points over the model loss surface towards the local minima associated by a human-understandable concept, e.g., class label for classifiers. With a systematic projection, scaling and refinement process, this information is transformed into an interpretable visualization without compromising its model-fidelity. The visualization serves as a standalone qualitative interpretation. With an extensive evaluation, we not only demonstrate successful visualizations for a variety of concepts for large-scale models, but also showcase an interesting utility of this new form of saliency mapping by identifying backdoor signatures in compromised classifiers.

Introduction

Deep perceptual models are at the heart of many recent scientific developments (LeCun, Bengio, and Hinton 2015), (Nature 2021). Their applications now reach beyond the rudimentary decision making, to the automation of high-stake domains, e.g., self-driving vehicles (Deng et al. 2021), smart security (DARPA 2020), health-care (Tang et al. 2021). This is a direct consequence of their established human-level ability to discriminate between intricate visual patterns (LeCun, Bengio, and Hinton 2015). However, like any deep learning approach, they are black-box techniques (Vinuesa and Sirmacek 2021). It is normally very hard to interpret what information has been learned by these models and how it influences their predictions.

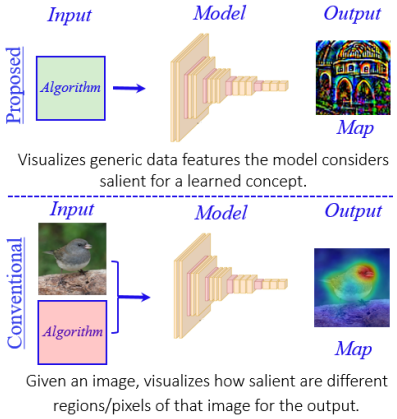

Figure 1: Difference between the proposed and conventional saliency mapping. (Top) We propose to compute maps that are input-agnostic and can attribute outputs to generic data concepts. (Bottom) Conventional approaches allow interpretations that are specific to a given input.

Owing to the significance of the perceptual models, their interpretability is currently considered a mainstream problem in computer vision (Fong, Patrick, and Vedaldi 2019), (Fong and Vedaldi 2017), (Petsiuk, Das, and Saenko 2018), (Zeiler and Fergus 2014), (Jalwana et al. 2021), (Selvaraju et al. 2017). In the literature, it has been translated to the task of identifying the input features deemed salient by the model to make its predictions. Mainly two broad strategies are popular to accomplish this task. The first searches for salient features in an input image by selectively perturbing its different regions and analyzing the resulting effects on the model predictions (Fong, Patrick, and Vedaldi 2019), (Fong and Vedaldi 2017), (Petsiuk, Das, and Saenko 2018), (Zeiler and Fergus 2014). Though effective, these methods may require heuristics to control their search space, potentially compromising the model-fidelity of the map due to the external influence (Jalwana et al. 2021).

The second strategy is commonly known as backpropagation saliency mapping (Selvaraju et al. 2017), (Rebuffi et al. 2020), (Simonyan, Vedaldi, and Zisserman 2013), (Springenberg et al. 2014), (Zeiler and Fergus 2014), (Zhang et al. 2018). It estimates feature saliency by analysing the gra-

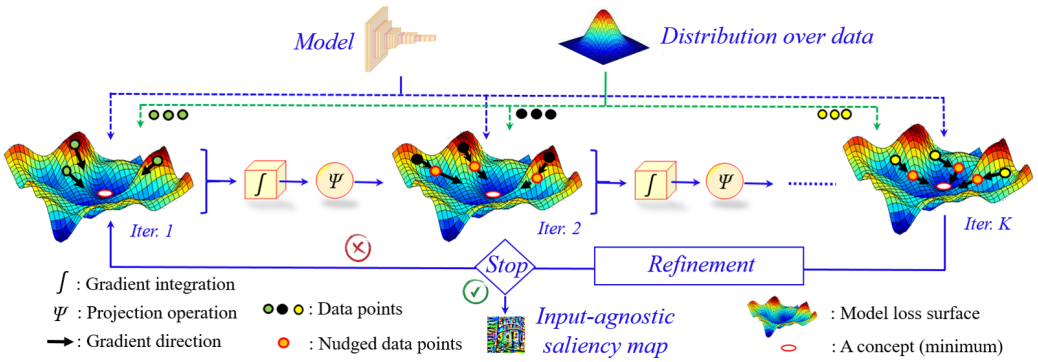

Figure 2: Central idea: We iteratively integrate and refine the gradient information of the model's loss surface to estimate an input-agnostic map that captures geometric input features considered salient by the model. An iteration draws i.i.d. samples from a distribution and nudges them to the nearby local minima that belong to a human-understandable concept (e.g., a class label). By integrating ($\int$) the gradient information from these nudges, and projecting (Ψ) it onto a norm-bounded surface, we amplify the salient geometric patterns w.r.t. the model. A refinement is further employed to improve the visualization without compromising the model-fidelity of the computed map.

dients and activations of the internal layers of the model for a given input. Low resolution map estimates, and failing the basic sanity checks, are known challenges faced by this strategy (Adebayo et al. 2018), (Jalwana et al. 2021). Leaving aside the peculiar issues of both strategies, the eventual saliency map computed by the both are always *input-specific*. Implying, the resulting interpretation is only valid for a single sample, see Fig. 1. This is too restrictive, especially when we consider that the ultimate objective of this research direction is *model* interpretation.

In this work, we approach saliency mapping from a different perspective. Our objective is to map generic (as opposed to input-specific) features of the modelled data which are deemed salient by the model. Under this input-agnostic perspective, the target map is a human-understandable visualization of the salient features attributed by the model to its outputs. The central idea of our technique is illustrated in Fig. 2. To compute the map, we iteratively explore the loss surface of the model by nudging a set of independent data points in the directions of nearby local minima associated with the output. The direction estimated by the nudges are integrated to characterize the landscape of the model's loss surface. By a gradual accumulation of this information and its projection onto a norm-restricted surface, we amplify the geometric correlation in the map which is subsequently refined computationally. When expanded to an image-grid, this map visualizes salient geometric features associated by the model to its output, i.e., a concept.

A unique property of our saliency map is that it enables a holistic visualization of the model's understanding of its own outputs. This can lead to many interesting applications. To illustrate one, this work also presents a case study to detect Trojan trigger patterns in compromised classifiers that contain backdoors (Wang, Hassan, and Akhtar 2022). We leverage our input-agnostic visualization to map the geometric patterns to which the classifier's output nodes are more sensitive. A model that has a backdoor, leads to visualisations that contain traces of the patterns used to trigger the backdoor, thereby allowing Trojan detection. The contributions of this paper are as follows.

- Systematically highlighting the limitations of input-specific saliency estimation, it introduces a new perspective of mapping generic data features attributed by a model to its outputs - input-agnostic saliency mapping.
- It devises a first-of-its-kind method for the proposed input-agnostic saliency mapping using model gradients.
- It showcases a utility of the new saliency mapping with backdoor trigger identification in classifiers.
- It introduces a quantitative metric for the newly proposed mapping framework and performs evaluation on large-scale ImageNet models to support the claims.

Related Work

For perceptual model interpretation through saliency mapping, there are two popular streams of methods. The first perturbs different regions of the input and records the effects of these perturbations to estimate the contribution of regions to the model prediction (Fong and Vedaldi 2017), (Fong, Patrick, and Vedaldi 2019), (Petsiuk, Das, and Saenko 2018), (Sundararajan, Taly, and Yan 2017), (Erion et al. 2021). The second uses activations of the deeper layers of the underlying neural network and back-propagated model gradients to estimate a saliency map (Selvaraju et al. 2017), (Jalwana et al. 2021), (Simonyan, Vedaldi, and Zisserman 2013). Due to the rising importance of model interpretation in numerous applications, there is a wide interest of the community in developing methods along both the directions. Influential contributions for both are discussed below.

Among the *perturbation-based saliency* methods, RISE (Petsiuk, Das, and Saenko 2018) and Occlusion (Zeiler and Fergus 2014) estimate the maps by weighting the perturbation masks with respect to the changes in the model confidence score. Other methods, for instance, Extremal perturbations (Fong, Patrick, and Vedaldi 2019), Meaningful perturbations (Fong and Vedaldi 2017),

Real-time saliency (Dabkowski and Gal 2017) and (Ribeiro, Singh, and Guestrin 2016), cast the underlying problem into an optimization objective. Although the perturbation methods are generally effective, they do not specifically shield the computed maps from external influence. A major challenge behind this problem is that the search nature of these methods requires exploring a vast solution space. Hence, the techniques rely on heuristics, priors or external constraints for tractability. This can affect the model-fidelity of the resulting interpretations (Jalwana et al. 2021).

Within the perturbation-based methods, there is also a branch of techniques that takes axiomatic approach towards saliency map creation (Sundararajan, Taly, and Yan 2017), (Erion et al. 2021), (Pan, Li, and Zhu 2021), (Srinivas and Fleuret 2019). Defining a path from a baseline image to the input, this paradigm integrates the effects of perturbation to images along this path to compute a saliency map. Although techniques vary in defining the paths and signal integration schemes, all of them compute saliency map for a single image. These methods do provide certain desirable theoretical properties, however they also entail high computational cost for image-specific interpretations.

The *back-propagation saliency methods* (Simonyan, Vedaldi, and Zisserman 2013), (Selvaraju et al. 2017), (Zeiler and Fergus 2014), (Jalwana et al. 2021) are highly popular as perceptual model interpretation tools. Normally, they construct a saliency map for the regions of an input image. These methods are also relatively computationally efficient (Zintgraf et al. 2017), (Kapishnikov et al. 2019). Simonyan, Vedaldi, and Zisserman (2013) were among the first to use model gradients for interpretation. Numerous improvements to this idea have been subsequently proposed to handle the noise sensitivity of the model gradients. Springenberg et al. (2014) proposed Guided back-prop, while Zeiler and Fergus (2014) altered the back-propagation rules for the ReLU layers of the model for that purpose. Similarly, SmoothGrad (Smilkov et al. 2017) computes the average gradients over the samples in the close vicinity of the original input to mitigate the gradient noise sensitivity.

Along a similar line of thought, DeepLIFT (Shrikumar, Greenside, and Kundaje 2017), Excitation Backprop (Zhang et al. 2018) and LRP (Bach et al. 2015), recast the back-propagation rules for saliency mapping to restrict the sum of attribution signal to unity. In another effort to control the signal noise, Sundararajan, Taly, and Yan (2017) combined multiple attribution maps. There are a number of methods that estimate the saliency map by merging layer activations of a model and its gradient information. Popular examples include CAM (Zhou et al. 2016), linear approximation (Kindermans et al. 2016), GradCAM (Selvaraju et al. 2017), NormGrad (Rebuffi et al. 2020) and CAMERAS (Jalwana et al. 2021). Though appearing late in the literature, these methods are currently dominating the backpropagation-based saliency mapping corpus of the literature.

Both of the above streams have an obvious limitation. They both explain the model behavior using only the features present in the given input sample. Realizing this restriction, works such as (Bau et al. 2017) and (Ghorbani et al. 2019) use 'general' object features to explain the model. However, these general features are extracted from a dataset. Thus, the eventual explanation is intrinsically biased to that data. Moreover, similar to the above-discussed streams, these methods must still use individual inputs to visualize the explanations. In this work, we devise a saliency mapping technique that truly liberates model explanation from the input, resulting in a first-of-its-kind input-agnostic saliency mapping mechanism.

Proposed Saliency Mapping

To motivate the idea and relate it to the existing practice, we start our discussion with the conventional saliency mapping.

Conventional Saliency Mapping

Let $\mathcal{K} : \mathcal{K}(\boldsymbol{I}) \rightarrow \boldsymbol{y}$ be a visual classifier that maps an image $\boldsymbol{I} \in \mathbb{R}^{h \times w \times c}$ with 'c' channels to a prediction vector $\boldsymbol{y} \in \mathbb{R}^{L}$. The ℓ^{th} coefficient y_ℓ of $\boldsymbol{y}$ is the largest if $\boldsymbol{I}$ belongs to class 'ℓ'. Assume a set $\mathcal{P}_I = \{p_I^1, p_I^2, ..., p_I^{w \times h}\} \subset \mathbb{R}^c$, which contains the pixels of $\boldsymbol{I}$. The broad common objective of the saliency-based interpretation methods for visual classifiers is to compute an ordered array $\mathcal{W}_I \subset \mathbb{R}$, s.t. $|\mathcal{W}_I| = |\mathcal{P}_I|$ and the i^{th} element of this array, i.e. $w_I^i \in \mathcal{W}_I$, encodes a weight for the corresponding element in $\mathcal{P}_I$. The $\mathcal{W}_I$ can be an array containing only binary values, which represents a mask for $\boldsymbol{I}$ that suppresses the irrelevant $p_I^i \in \mathcal{P}_I$ for the transform $\mathcal{K}(\boldsymbol{I}) \rightarrow \boldsymbol{y}$. Or, it can contain real values encoding the importance of all $p_I^i \in \mathcal{P}_I$ for the performed transform.

The above formalization presents a unified view of the objective of the prevailing *input-specific* saliency methods. To this end, the existing techniques seek the function $\mathcal{S} : \mathcal{S}(\mathcal{K}, \boldsymbol{I}) \rightarrow \mathcal{W}_I$ for saliency mapping. We refer to $\mathcal{S}$ as the saliency function in the text to follow.

Problem with the Input-Specific View: Though useful for certain objectives, we identify that the sought saliency function only weakly depends on the classifier itself. This makes it susceptible to providing misleading model interpretations. We make a formal proposition about it.

Proposition 1: *Due to the weak dependence of the saliency function $\mathcal{S}$ on the classifier $\mathcal{K}$, $\mathcal{S}$ is susceptible to compute $\mathcal{W}_I$ for a canonical classifier $\mathcal{K}^*$ instead of $\mathcal{K}$.*

To establish Prop. (1), we need to first define *canonical classifier* and *weak dependence*, as understood in this work.

Definition 1: (Canonical classifier) *Provided that 'ℓ' is the known label of an input $\boldsymbol{I}$, a canonical classifier $\mathcal{K}^*$ behaves as $\mathcal{K}^*(\boldsymbol{I}) \rightarrow \boldsymbol{y}^*$, where $y_\ell \rightarrow 1$ for $\boldsymbol{y}^*$.*

Definition 2: (Weak dependence) *For a model $\mathcal{M} : \mathcal{M}(\boldsymbol{I}) \rightarrow \boldsymbol{y}$ whose prediction behavior can be expressed as a piece-wise function*

$$\mathcal{M}(\boldsymbol{I}) = \begin{cases} \mathcal{M}_1 : \mathcal{M}_1(\boldsymbol{I}) \rightarrow \boldsymbol{y}_1 & \boldsymbol{I} \in \mathcal{U}_1 \\ \vdots & \vdots \\ \mathcal{M}_n : \mathcal{M}_n(\boldsymbol{I}) \rightarrow \boldsymbol{y}_n & \boldsymbol{I} \in \mathcal{U}_n, \end{cases}$$

where $\mathcal{U}_i$ is the i^{th} open disconnected set of the nearby samples of $\boldsymbol{I}$, a saliency function $\mathcal{S}$ only weakly depends on $\mathcal{M}$ when $\mathcal{W}_I^p \approx \mathcal{W}_I^q$ for $\mathcal{S}(\mathcal{M}_p, \boldsymbol{I}) \rightarrow \mathcal{W}_I^p$ and $\mathcal{S}(\mathcal{M}_q, \boldsymbol{I}) \rightarrow \mathcal{W}_I^q$ for the sets $\mathcal{U}_p$ and $\mathcal{U}_q$, despite $\mathcal{U}_p \cap \mathcal{U}_q = \emptyset$.

The above-defined notion of weak dependence is partially inspired by the work of Srinivas and Fleuret (2019). However, the context and application of our definition is different. To understand the weak dependence property as stipulated by Def. (2), imagine an image of a 'panda' on a chair, and an image of a 'queen' on a throne. Due to their significant content dissimilarity, these images exist in different sets $\mathcal{U}_p$ and $\mathcal{U}_q$. Let $\mathcal{M}$ be a classifier that correctly predicts the labels of both the images. This will naturally result in largely different corresponding prediction vectors $\boldsymbol{y}_p$ and $\boldsymbol{y}_q$. Implying, the underlying classifier behavior for these images is modelled by considerably different components $\mathcal{M}_p$ and $\mathcal{M}_q$ of the piece-wise function $\mathcal{M}$. A strong dependence of $\mathcal{S}$ on $\mathcal{M}$ asserts that $\mathcal{S}$ computes proportionally different $\mathcal{W}_I^p$ and $\mathcal{W}_I^q$. However, an input-specific saliency function $\mathcal{S}$ may commit to very similar maps for the two images, i.e., $\mathcal{W}_I^p \approx \mathcal{W}_I^q$, when the silhouettes of the panda and the queen in the images are very similar, despite $\mathcal{U}_p \cap \mathcal{U}_q = \emptyset$.

The above example illustrates the phenomenon of weak dependence of the saliency function on the classifier under the traditional input-specific saliency mapping. Now, we turn to establishing Prop. (1). We focus on the pursuit of estimating a perfect map $\mathcal{W}_I^*$ using $\mathcal{S}$. To exemplify, if the sought $\mathcal{W}_I^*$ is a binary mask, it suppresses all the pixels in $\boldsymbol{I}$ that are irrelevant to the object of category ℓ. Notice that, existing methods perform saliency mapping using apriori knowledge of ℓ. In effect, they pose the query, *"provided that $\boldsymbol{I}$'s label is ℓ, compute $\mathcal{W}_I$"*. Irrespective of the used saliency function, the ideal solution, i.e., $\mathcal{W}_I^*$, to this query is a map that identifies the pixels of $\boldsymbol{I}$ that maximize the value of y_ℓ. Following Def. (2), since $\mathcal{S}$ is only weakly dependent on the model itself, the quest of computing $\mathcal{W}_I^*$ can easily force $\mathcal{S}$ to select a piece-wise component of $\mathcal{M}$ that favors $y_\ell \rightarrow 1$, irrespective of the actual prediction vector of $\boldsymbol{I}$. Such a solution would actually be performing saliency mapping for the canonical classifier, as defined in Def. (1), not the observed model behavior.

Our analysis above indicates a pitfall in the pursuit of precise input-specific saliency mapping. Its obvious implication is that highly refined image saliency maps can actually misinterpret the actual model behavior. The literature already identifies multiple saliency methods failing basic sanity checks (Adebayo et al. 2018). Interestingly, sanity checks failure is much more common among the methods aiming at a higher map precision. This phenomenon is naturally explainable through our above-provided analysis.

Remark: Under the completeness axiom (Sundararajan, Taly, and Yan 2017), y_ℓ is bounded to $\delta \leq 1$ in Def. (1). To that end, our analysis still holds for $y_\ell \rightarrow \delta$. Implying, the input-specific methods satisfying the completeness property can still suffer from misleading maps.

The need for input-agnostic view: Not only that input-specific maps are susceptible to misleading interpretations, an inaccurate map for a given sample becomes a fatal error for these methods because they can only offer interpretation with respect to a single sample. Addressing the problem at the grass-root level requires the interpretation to be agnostic to the input samples. To represent the model well, such an interpretation must strongly depend on the model itself. This view of model explanation can not only allow us to ensure model-fidelity of the interpretations, but also enable us to answer more general queries about the model. Hence, we develop a saliency mapping method under this view, considering it as a complementary interpretation tool.

Input-Agnostic Saliency Mapping

We first provide a concise definition of the desired input-agnostic saliency mapping function $\mathcal{S}_\mathcal{K}$ - the subscript indicates that the function depends on $\mathcal{K}$.

Definition 3: (Input-agnostic saliency mapper) *For a given classifier $\mathcal{K}$, $\mathcal{S}_\mathcal{K} : \mathcal{S}_\mathcal{K}(c^i_\Im) \rightarrow \boldsymbol{\nu}^i_{c_\Im}$ is an input-agnostic saliency mapper, where $\boldsymbol{\nu}^i_{c_\Im} \in \mathbb{R}^{h\times w\times c}$ is a human-understandable visualization of a semantic concept $c^i_\Im \in \mathcal{C} = \{c^1_\Im, c^2_\Im, ..., c^L_\Im\}$ and $\Im$ denotes the input data distribution over which $\mathcal{K}$ is induced.*

Multiple aspects in Def. (3) need emphasis. First, notice the absence of $\boldsymbol{I}$ in favor of the use of distribution $\Im$ - inline with the key idea of input-agnostic saliency. Second, the saliency map is now with respect to a semantic concept $c^i_\Im$, not an input sample. Here, the semantic concept (or simply the 'concept') is a high-level human-understandable notion that is also discernible to the model. In the context of visual classifiers, this work considers the concept to be a class label, hence $|\mathcal{C}| = L$. It is noteworthy though, we do not enforce any other constraint over $c^i_\Im$. It is implicit that the concept emerges from $\Im$, and it is understood by $\mathcal{K}$ because the classifier is learned using $\Im$. Lastly, the output of the saliency function is now an image $\boldsymbol{\nu}^i_{c_\Im} \in \mathbb{R}^{h\times w\times c}$ instead of a set of weights. This image visualizes a human-defined concept, as understood by the classifier. Since a concept can be a broad and complex idea, it is likely to exhibit plentiful visual manifestations. Hence, $\boldsymbol{\nu}^i_{c_\Im}$ is not expected to be unique. This makes $\mathcal{S}_\mathcal{K}$ a one-to-many mapping function.

Optimization objective: To compute the desired visualization, we need to optimize for

$$\max_{\boldsymbol{\nu}} \mathrm{P}\left(\mathcal{K}(\boldsymbol{\nu}_{c_\Im}) \rightarrow \boldsymbol{y}_{\ell_c} \;\middle|\; \Im\right) \;\text{ s.t. }\; \boldsymbol{\nu}_{c_\Im} = \mathcal{F}(\mathcal{K}), \qquad (1)$$

where $\mathrm{P}(.|.)$ denotes a conditional probability, ℓ_c is the class label for the concept $c_\Im$, and $\mathcal{F}(.)$ is a non-trivial function ensuring that the visualization strictly depends on the classifier $\mathcal{K}$. The input-agnostic saliency mapper $\mathcal{S}_\mathcal{K}$ in Def. (3) instantiates $\mathcal{F}(\mathcal{K})$, where we let $c^i_\Im \in \mathcal{C}$ as an input parameter of the function to allow visualizations for the multi-class classifiers. In Eq. (1) and text below, we ignore the superscript 'i' to avoid clutter. We also ignore $\Im$ for clarity when emphasis on the data distribution is not required.

Algorithm: Following Eq. (1), the implementation objective of $\mathcal{S}_\mathcal{K}$ is to maximize the probability $\mathrm{P}(\mathcal{K}(\mathcal{S}_\mathcal{K}(c_\Im)) \rightarrow \ell_c)$ in an input-agnostic manner. We achieve this with the saliency mapper given as Alg. (1). The algorithm leverages insights from Lemma (1) below.

Lemma 1: *For $\mathcal{K}$ with cross-entropy loss $\mathcal{J}_{CE}$, $P(\mathcal{K}(\boldsymbol{I}) \rightarrow \boldsymbol{y}_{\ell_c} | \boldsymbol{I} \sim \Im)$ increases along $-\mathop{\mathbb{E}}_{\boldsymbol{I}\sim\Im}[\nabla_{\boldsymbol{I}} \mathcal{J}_{CE}(\Im, \boldsymbol{\theta}, \ell_c)]$.*

Proof: *Denote "$\mathcal{K}(\boldsymbol{I}) \rightarrow \boldsymbol{y}_{\ell_c} | \boldsymbol{I} \sim \Im$" by $\zeta_{\boldsymbol{I}}$. The $P(\zeta_{\boldsymbol{I}})$ increases along $\nabla_{\boldsymbol{I}}(\log P(\zeta_{\boldsymbol{I}}))$, where $\nabla_{\boldsymbol{I}}$ is the derivative*

w.r.t. $\boldsymbol{I}$. *For* $\mathcal{K}$ *with the loss* $\mathcal{J}_{CE}$ *and model parameters* $\boldsymbol{\theta}$, *this is the same direction as* $-\nabla_{\boldsymbol{I}}\mathcal{J}_{CE}(\Im, \boldsymbol{\theta}, \ell_c)$. *Thus,* $P(\zeta_{\boldsymbol{I}})$ *will increase along* $-\mathop{\mathbb{E}}_{\boldsymbol{I}\sim\Im}[\nabla_{\boldsymbol{I}}\mathcal{J}_{CE}(\Im, \boldsymbol{\theta}, \ell_c)]$.

In Alg. (1), we perform a guided stochastic gradient descent over the loss surface of $\mathcal{K}$ with the help of data distribution $\Im$. The distribution is approximated with a set of its samples in $\overline{\mathcal{I}}$. In an iteration, the algorithm computes the model's loss gradients for the concept label ℓ_c w.r.t. a mini-batch of the samples from $\Im$. Conceptually, these gradients point to the directions of the local minima associated with ℓ_c. In the light of Lemma (1), we compute Expected value of the gradients and further guide the descent with the first and second moments - lines 5-7. The use of moments is inspired by the Adam optimizer (Kingma and Ba 2014). Hence, following Adam, we also fix the values of the hyper-parameters β_1 and β_2. We additionally guide the descent with a binary selection between the original and the flipped direction if the latter identifies a better local solution - lines 8 -13[1]. We empirically found it beneficial in our experiments. Collectively, the process from line 3 to 14 in Alg. (1) implements the integration of gradient information in Fig. 2.

In an iteration, $\boldsymbol{\nu}_k$ is able to encode patterns related to ℓ_c because it is computed under the objective of nudging random samples to the local minima related to the concept - lines 8 -13. The model must associate the features encoded in $\boldsymbol{\nu}_k$ to ℓ_c for the nudging to work. However, an unbounded construction of $\boldsymbol{\nu}_k$ can lead to uninteresting solutions where the algorithm maximizes only the influential component(s) of $\boldsymbol{\nu}_k$ to achieve its objective. This does not help the cause of *human-meaningful* visualization with $\boldsymbol{\nu}_k$. To encourage correlation among the components of $\boldsymbol{\nu}_k$, we project it onto a bounded ℓ_2-ball in each iteration - line 15, which implements the projection operation of Fig. 1. The underlying constraint resulting from this projection, i.e., $||\boldsymbol{\nu}_k||_2 \leq \eta$, leads to a collaborative behavior among the coefficients of $\boldsymbol{\nu}_k$ that emerges into meaningful geometric patterns when the vector is visualized as an image.

In Alg. (1), we allow multiple hyper-parameters as inputs along the classifier $\mathcal{K}$ and class label ℓ_c. Considering the above discussion and the fact that Alg. (1) solves a stochastic gradient descent problem, the significance of these parameters is self-explanatory, except for the seed $\boldsymbol{\nu}$. As shown in Fig. 2, we further refine the visualization after K iterations of Alg. (1). Subsequently, Alg. (1) is again applied to improve the visualization. The seed $\boldsymbol{\nu}$ is the output of the refinement process. In the first round, $\boldsymbol{\nu} \in \mathbb{R}^{h\times w\times c}$ is a black image.

Map refinement: The output of Alg. (1) is a visualization of the salient geometric features associated by $\mathcal{K}$ to a concept, i.e., label ℓ_c. However, this map is constructed with the model's *gradient* information, which can be noisy. Hence, we need to further refine the map. To that end, we solve for the following optimization problem

$$\min_{\boldsymbol{\nu}_c} \mathbb{E}\Big[\mathcal{J}_{CE}(\Im - \boldsymbol{\nu}_c, \boldsymbol{\theta}, \ell_c) + \lambda\big(\boldsymbol{\nu}_c \odot (\mathbf{1} - \boldsymbol{\Xi})\big)\Big], \quad (2)$$

[1]Clip is the standard clipping function that clips out any value exceeding the image dynamic range.

Algorithm 1: Input-agnostic saliency mapper $\mathcal{S}_{\mathcal{K}}$

Input: Classifier $\mathcal{K}$, concept label ℓ_c, sample set $\overline{\mathcal{I}}$, mini-batch size b, total iterations K, ball norm η, seed $\boldsymbol{\nu}$.
Output: Visualisation $\boldsymbol{\nu}_c \in \mathbb{R}^{h\times w\times c}$.

1: Initialize $\boldsymbol{\nu}_0 = \boldsymbol{\nu}$, $\boldsymbol{\mu}_0$, $\boldsymbol{\sigma}_0$ to $\mathbf{0}$ and $k = 0$.
Set $\beta_1 = 0.9$, $\beta_2 = 0.999$.
2: **for** $k = 0$ to K **do**
3: $\quad\mathcal{I} \sim \overline{\mathcal{I}}$ s.t. $|\mathcal{I}| = b$ and apply $\forall \boldsymbol{I}_i \in \mathcal{I}, \text{Clip}\,(\boldsymbol{I}_i - \boldsymbol{\nu}_k)$
4: $\quad\boldsymbol{x}_k \leftarrow \mathop{\mathbb{E}}_{\boldsymbol{I}_i\in\mathcal{I}}\big[\nabla_{\boldsymbol{I}_i}\mathcal{J}(\boldsymbol{\theta}, \ell_c)\big]$
5: $\quad\boldsymbol{\mu}_k \leftarrow \beta_1\boldsymbol{\mu}_{k-1} + (1-\beta_1)\boldsymbol{x}_k$
6: $\quad\boldsymbol{\sigma}_k \leftarrow \beta_2\boldsymbol{\sigma}_{k-1} + (1-\beta_2)(\boldsymbol{x}_k \odot \boldsymbol{x}_k)$
7: $\quad\boldsymbol{v} \leftarrow \Big(\boldsymbol{\mu}_k\sqrt{1-\beta_2^k}\Big) \odot \big(\sqrt{\boldsymbol{\sigma}_k}(1-\beta_1^k)\big)^{-1}$
8: $\quad\mathcal{I}_{\boldsymbol{v}}^{+} \leftarrow \big\{\overline{\boldsymbol{I}}_i : \overline{\boldsymbol{I}}_i = \text{Clip}\Big(\boldsymbol{I}_i - (\boldsymbol{v}_{k-1} + \frac{\boldsymbol{v}}{||\boldsymbol{v}||_2})\Big)\big\}\forall \boldsymbol{I}_i \in \mathcal{I}$
9: $\quad\mathcal{I}_{\boldsymbol{v}}^{-} \leftarrow \big\{\overline{\boldsymbol{I}}_i : \overline{\boldsymbol{I}}_i = \text{Clip}\Big(\boldsymbol{I}_i - (\boldsymbol{v}_{k-1} - \frac{\boldsymbol{v}}{||\boldsymbol{v}||_2})\Big)\big\}\forall \boldsymbol{I}_i \in \mathcal{I}$
10: $\quad$**if** $\mathbb{E}\big[\mathcal{K}(\mathcal{I}_{\boldsymbol{v}}^{+}) \rightarrow \ell_c\big] \geq \mathbb{E}\big[\mathcal{K}(\mathcal{I}_{\boldsymbol{v}}^{-}) \rightarrow \ell_c\big]$ **then**
11: $\quad\quad\boldsymbol{\nu_k} \leftarrow \boldsymbol{\nu}_{k-1} + \boldsymbol{v}$
12: $\quad$**else**
13: $\quad\quad\boldsymbol{\nu_k} \leftarrow \boldsymbol{\nu}_{k-1} - \boldsymbol{v}$
14: $\quad$**end if**
15: $\quad\boldsymbol{\nu_k} \leftarrow \Psi(\boldsymbol{\nu}_k)$, s.t. $\Psi(\boldsymbol{\nu}_k) = \boldsymbol{\nu_k} \odot \min\Big(1, \frac{\eta}{||\boldsymbol{\nu}_k||_2}\Big)$
16: **end for**
17: $\boldsymbol{\nu}_c = \boldsymbol{\nu}_K$
18: return

where λ is a regularizer and $\boldsymbol{\Xi}$ is a weighting matrix - discussed shortly, the other symbols follow from above. In Eq. (2), $\Im - \boldsymbol{\nu}_c$ signifies (another) nudge of the inputs towards the local minima associated with ℓ_c. This follows from the central idea of Alg. (1), and is possible because $\boldsymbol{\nu}_c$ is already available to us. However, now we would like it to be mainly influenced by the regular geometric features of ℓ_c, disregarding the noisy component of $\boldsymbol{\nu}_c$. The second term in Eq. (2) imposes that. Here, $\boldsymbol{\Xi} \in \mathbb{R}^{h\times w\times c}$ is a matrix computed by forward passing $\boldsymbol{\nu}_c$ through $\mathcal{K}$ and projecting the activations of the convolutional base of $\mathcal{K}$ onto $\mathbb{R}^{h\times w\times c}$ with interpolation. The process resembles the use of activations in computing GradCAM maps (Selvaraju et al. 2017). However, instead of computing the map, we use it to clean the map. Hence, we normalize the coefficients of $\boldsymbol{\Xi}$ in the range [0,1] and perform an element-wise weighting of $\boldsymbol{\nu}_c$ with $\mathbf{1} - \boldsymbol{\Xi}$, denoted by $\odot$ in Eq. (2). This suppresses the component of $\boldsymbol{\nu}_c$ that does not contain perceptually meaningful information, as deemed by the model itself.

We achieve the optimization objective of Eq. (2) using the Adam optimizer. The resulting cleaned up map is further clipped and projected onto an ℓ_2-ball for a seamless subsequent processing with Alg. (1). Our method cycles between Alg. (1) and the refinement stage to compute the final visualization. It is noteworthy that the complete signal in our eventual map originates in the model itself, and no part of it is generated by an external operator to maintain model fidelity (Jalwana et al. 2021). The only external operator we use is the interpolation function in the refinement stage. However, that is used to 'remove' the unwanted signal from

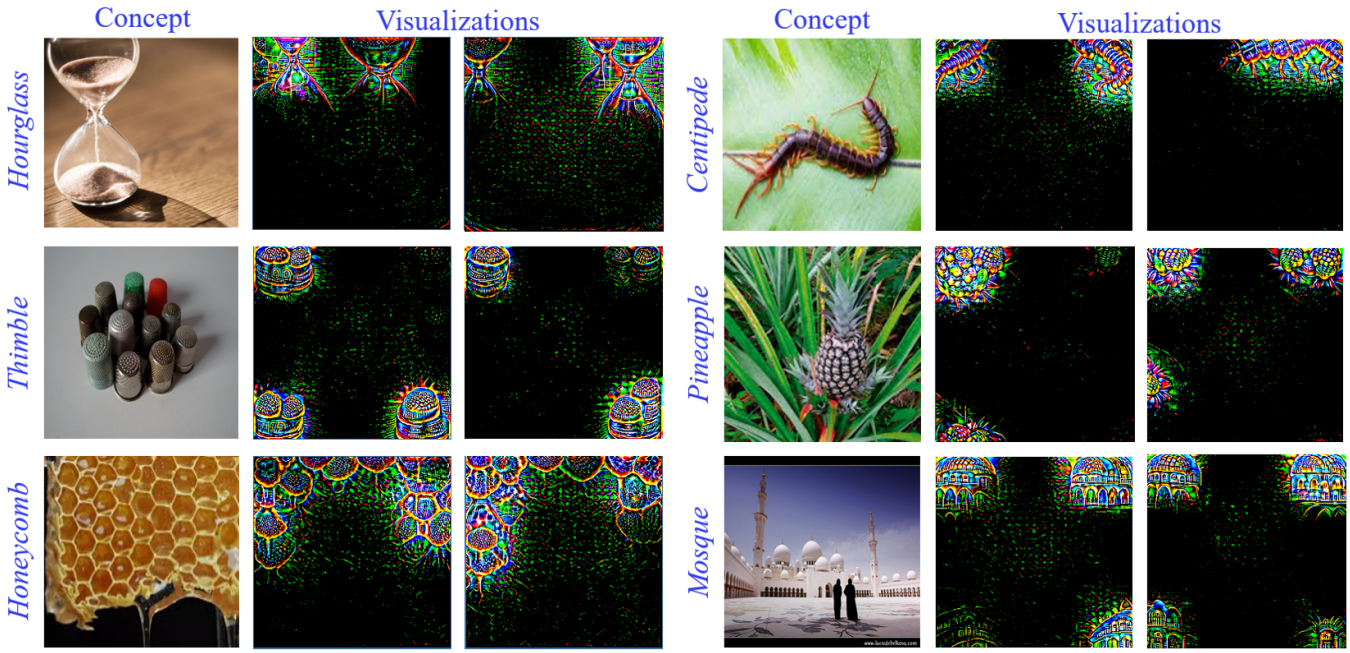

Figure 3: Representative input-agnostic visualizations of human-defined concepts, as understood by VGG-16. Natural image for each concept is provided for reference only.

the map. Jalwana et al. (2021) noted such model-fidelity to be highly desirable for reliable model interpretation.

Experiments

Visualizations

To verify our approach, we apply our technique to visualize ImageNet concepts (Deng et al. 2009), as understood by VGG-16 and ResNet-50 models. We use ImageNet pre-trained Pytorch models, and randomly pick 10 labels to visualize. Recall, in the context of this work, a concept is a high-level human-understandable notion that is also discernible for the model. Hence, the experiments consider ImageNet class labels as the concepts. In Fig. 3, we show example visualizations of representative concepts resulting from the proposed technique for VGG-16. .

We observe in the computed images that the patterns are visually relatable to the concepts mentioned along side. Moreover, we are able to generate multiple visualizations of a given concept that are slightly different from each other, but represent the concept well. Multiple images for a given concept are generated by simply executing the method multiple times. The complementary visualizations for different runs indicate generalizable concept understanding by the deep visual classifier. Notice that, the shown results are useful in answering queries such as, *"what concept the model associates to its specified output neuron?"*, or *"what generic geometric patterns the model attributes to a learned concept/class label?"*. These kind of queries are not answerable with image-specific view of saliency mapping.

Quantitative results: Whereas our method is a qualitative interpretation technique, we also allow its quantitative evaluation. To that end, we must introduce a new metric to evaluate our first-of-its-kind input-agnostic saliency mapping technique. Termed 'model-score' - $\mathrm{M}_{\text{score}}$, the proposed metric is inspired by Inception-score (Salimans et al. 2016), which is commonly used to benchmark generative model performance. Since our method eventually results in a 'visualisation', it can also be seen as an image generation technique. Inception-score measures the meaningfulness of generated images by selecting an Inception model as a proxy for the human visual system. Let us call the classifier used to generate our visualization, a 'source' classifier $\mathcal{K}_s$. For evaluation, we seek to quantify meaningfulness of our visualizations w.r.t. any 'target' classifier $\mathcal{K}_t$ that is induced over the same data distribution used to train $\mathcal{K}_s$. Hence, in essence, we seek a generalized version of the Inception-score that also accounts for the fact that the image is generated with respect to a given source classifier.

We compute the proposed $\mathrm{M}_{\text{score}}$ by measuring the probabilities $\mathrm{P}^{t|s}_{\text{cond}}$ and $\mathrm{P}^{t|s}_{\text{marg}}$, where

$$\mathrm{P}^{t|s}_{\text{cond}} = p(\mathcal{K}_t(\boldsymbol{\nu}_c) \rightarrow y_{\ell_c} | \boldsymbol{\nu}_c = \mathcal{S}_{\mathcal{K}_s}(c)), \quad (3)$$

$$\mathrm{P}^{t|s}_{\text{marg}} = \sum_i p(\mathcal{K}_t(\boldsymbol{\nu}^i_c) \rightarrow y_{\ell_c} | \boldsymbol{\nu}^i_c = \mathcal{S}_{\mathcal{K}_s}(c)). \quad (4)$$

In the above expressions, $\mathrm{P}^{t|s}_{\text{cond}}$ and $\mathrm{P}^{t|s}_{\text{marg}}$ are respectively the conditional and marginal probabilities that the target classifier correctly predicts the class label of the concept visualized by $\boldsymbol{\nu}_c$, where the label is determined by applying the saliency function $\mathcal{S}_{\mathcal{K}_s}$ to $\mathcal{K}_s$. The $\mathrm{M}_{\text{score}}$ is then computed as

$$\mathrm{M}_{\text{score}}(t|s) = \exp\left(\mathbb{E}\left[\mathrm{KL}\left(\mathrm{P}^{t|s}_{\text{cond}} \,||\, \mathrm{P}^{t|s}_{\text{marg}}\right)\right]\right)/L, \quad (5)$$

Source(s) \ Target(t)	VGG-16	ResNet-50	DenseNet-121	Avg.
VGG-16	0.99	0.71	0.81	0.84
ResNet-50	0.69	0.99	0.94	0.89

Table 1: $\mathrm{M}_{\text{score}}(t|s)$ of the visualized concepts for VGG-16 and ResNet-50. Maximum possible value for any target-source pair is 1. Larger values are more desirable.

where KL denotes the Kullback–Leibler divergence, and L is the total number of visualized concepts.

The proposed $\mathrm{M}_{\text{score}}$ is a comprehensive metric for a given source-target classifier pair trained on the same data distribution $\Im$. By definition it values range in $[\frac{1}{L}, 1]$, where larger values as more desirable. To keep experiments computationally manageable, we assume that the source and target classifiers in our experiments only understand the chosen L = 10 concepts. We generate 10 visualizations per concept for both VGG-16 and ResNet-50 and compute $\mathrm{M}_{\text{score}}$ using different target models. The results are summarized in Table 1.

In Table 1, the visualizations generated for a given (source) model have $\mathrm{M}_{\text{score}} \approx 1$ when the same model is used as the target. This verifies that our visualizations are correctly mapping the model's understanding of the underlying concepts with high fidelity. When we change the target model to other ImageNet models (same $\Im$), the score slightly drops. However, it still remains considerably high. This signifies that the visualizations are indeed generically understandable by the different models of $\Im$. Our results align perfectly with the intuition that $\mathrm{M}_{\text{score}}(t|s)$ should be larger when $t \neq s$ is a more accurate classifier for the concepts in $\Im$. The large $\mathrm{M}_{\text{score}}$ values across different well-trained target classifiers conclusively establish successful visual mapping of the generic concepts by our method.

Backdoor Detection

Backdoor (a.k.a. Trojan) attacks manipulate visual models by forcing them to misbehave when exposed to a 'trigger' in the input (Wang, Hassan, and Akhtar 2022). These attacks are stealthy because the model behaves normally for clean inputs, and the model user is unaware of the trigger pattern. Since the presence of a trigger in the input is not known, it is not possible to use image-specific saliency to identify backdoor in the model. The ability to visualize the patterns associated with a model's outputs in an input-agnostic manner can resolve the issue. We conduct a study in which three *compromised* VGG-16 models are trained using ImageNet. These models behave normally for a clean input, but always predict a given (random) target label when the input contains a trigger pattern.

In Fig. 4, we show the trigger patterns used in our experiments, which are chosen at random based on the literature (Wang, Hassan, and Akhtar 2022). We apply the proposed input-agnostic saliency mapping to the compromised models. It was found that for the (false) target class, the constructed maps contained a discernible visual footprint of the trigger pattern. The figure shows example maps for the Castle class of a compromised model, when the trigger (Nike sign) caused the model to predict Castle as the label of any image containing the Nike sign.

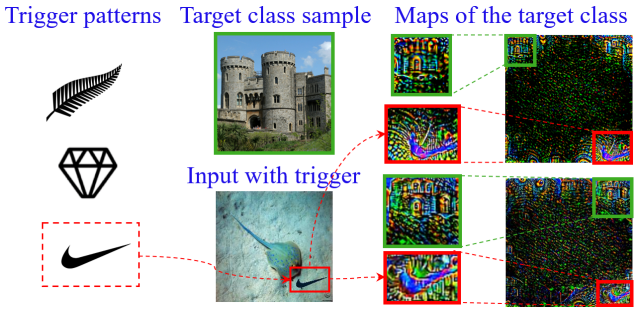

Figure 4: Backdoor identification. Left: Used trigger patterns. Center: Example images of the Target class (Castle) and a triggered input. The model correctly predicts the label of castle images, and also predicts any triggered image as Castle. Right: Representative examples of computed maps for the concept Castle, capturing discernible trigger patterns.

We note that modern backdoor attacks are not limited to only using 'visible' trigger patterns (Wang, Hassan, and Akhtar 2022). Detection of non-visible triggers is not covered by our study. Our intention here is to showcase a utility of our novel view of input-agnostic saliency mapping. We believe, further exploration along this view will allow multiple interesting applications, and open new avenues for the model interpretation methods.

Hyper-Parameter Settings

It may appear that Alg. (1) and map refinement use multiple hyper-parameters. However, those parameters are mostly related to help the underlying gradient descent schemes, and are widely understood, which helps in to easily selecting their reasonable values. The finally selected parameter values are $b = 128$, $K = 650$, $\eta = 30$ for Alg. (1). For the refinement, we use 150 iterations while keep the other related parameter values the same, and let $\lambda = 50$. We cycle between Alg. (1) and refinement 2 times. This requires on average $\sim$19 and $\sim$14 minutes respectively to generate an image for VGG-16 and ResNet-50 on NVIDIA RTX 3090 with 24GB RAM using Pytorch implementation.

Conclusion

We provide a novel perspective on saliency mapping of visual classifiers that maps generic geometric features associated by the model with its outputs. Our input-agnostic map construction gradually accumulates the gradient information of the model's loss surface with respect to its training distribution. We also motivate the need of such a map theoretically, highlighting a critical limitation of the input-specific saliency mapping paradigm. We demonstrate a utility of the newly found saliency mapping in backdoor detection of compromised models. Our novel perspective is likely to instigate interesting methods and their uncharted utilities in model interpretation domain.

Acknowledgments

Dr. Naveed Akhtar is a recipient of the Office of National Intelligence, National Intelligence Postdoctoral Grant (project number NIPG-2021-001) funded by the Australian Government.

References

Adebayo, J.; Gilmer, J.; Muelly, M.; Goodfellow, I.; Hardt, M.; and Kim, B. 2018. Sanity checks for saliency maps. *Advances in neural information processing systems*, 31.

Bach, S.; Binder, A.; Montavon, G.; Klauschen, F.; Müller, K.-R.; and Samek, W. 2015. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7): e0130140.

Bau, D.; Zhou, B.; Khosla, A.; Oliva, A.; and Torralba, A. 2017. Network dissection: Quantifying interpretability of deep visual representations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 6541–6549.

Dabkowski, P.; and Gal, Y. 2017. Real time image saliency for black box classifiers. In *Advances in Neural Information Processing Systems*, 6967–6976.

DARPA. 2020. AI Next Campaign. https://www.darpa.mil/work-with-us/ai-next-campaign. [Online; accessed 24-Jun-2022].

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 248–255. Ieee.

Deng, Y.; Zhang, T.; Lou, G.; Zheng, X.; Jin, J.; and Han, Q.-L. 2021. Deep learning-based autonomous driving systems: a survey of attacks and defenses. *IEEE Transactions on Industrial Informatics*, 17(12): 7897–7912.

Erion, G.; Janizek, J. D.; Sturmfels, P.; Lundberg, S. M.; and Lee, S.-I. 2021. Improving performance of deep learning models with axiomatic attribution priors and expected gradients. *Nature machine intelligence*, 3(7): 620–631.

Fong, R.; Patrick, M.; and Vedaldi, A. 2019. Understanding deep networks via extremal perturbations and smooth masks. In *Proceedings of the IEEE/CVF international conference on computer vision*, 2950–2958.

Fong, R. C.; and Vedaldi, A. 2017. Interpretable explanations of black boxes by meaningful perturbation. In *Proceedings of the IEEE International Conference on Computer Vision*, 3429–3437.

Ghorbani, A.; Wexler, J.; Zou, J. Y.; and Kim, B. 2019. Towards automatic concept-based explanations. *Advances in Neural Information Processing Systems*, 32.

Jalwana, M. A.; Akhtar, N.; Bennamoun, M.; and Mian, A. 2021. CAMERAS: Enhanced resolution and sanity preserving class activation mapping for image saliency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 16327–16336.

Kapishnikov, A.; Bolukbasi, T.; Viégas, F.; and Terry, M. 2019. Xrai: Better attributions through regions. In *Proceedings of the IEEE International Conference on Computer Vision*, 4948–4957.

Kindermans, P.-J.; Schütt, K.; Müller, K.-R.; and Dähne, S. 2016. Investigating the influence of noise and distractors on the interpretation of neural networks. *arXiv preprint arXiv:1611.07270*.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

LeCun, Y.; Bengio, Y.; and Hinton, G. 2015. Deep learning. *nature*, 521(7553): 436–444.

Nature. 2021. Gravity, AlphaFold and neural interfaces: a year of remarkable science. https://www.nature.com/articles/d41586-021-03730-w. [Online; accessed 24-Jun-2022].

Pan, D.; Li, X.; and Zhu, D. 2021. Explaining deep neural network models with adversarial gradient integration. In *Thirtieth International Joint Conference on Artificial Intelligence (IJCAI)*.

Petsiuk, V.; Das, A.; and Saenko, K. 2018. Rise: Randomized input sampling for explanation of black-box models. *arXiv preprint arXiv:1806.07421*.

Rebuffi, S.-A.; Fong, R.; Ji, X.; and Vedaldi, A. 2020. There and Back Again: Revisiting Backpropagation Saliency Methods. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8839–8848.

Ribeiro, M. T.; Singh, S.; and Guestrin, C. 2016. " 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*, 1135–1144.

Salimans, T.; Goodfellow, I.; Zaremba, W.; Cheung, V.; Radford, A.; and Chen, X. 2016. Improved techniques for training gans. In *Advances in neural information processing systems*, 2234–2242.

Selvaraju, R. R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; and Batra, D. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, 618–626.

Shrikumar, A.; Greenside, P.; and Kundaje, A. 2017. Learning important features through propagating activation differences. *arXiv preprint arXiv:1704.02685*.

Simonyan, K.; Vedaldi, A.; and Zisserman, A. 2013. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*.

Smilkov, D.; Thorat, N.; Kim, B.; Viégas, F.; and Wattenberg, M. 2017. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*.

Springenberg, J. T.; Dosovitskiy, A.; Brox, T.; and Riedmiller, M. 2014. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*.

Srinivas, S.; and Fleuret, F. 2019. Full-gradient representation for neural network visualization. *Advances in neural information processing systems*, 32.

Sundararajan, M.; Taly, A.; and Yan, Q. 2017. Axiomatic attribution for deep networks. *arXiv preprint arXiv:1703.01365*.

Tang, S.; Wang, C.; Nie, J.; Kumar, N.; Zhang, Y.; Xiong, Z.; and Barnawi, A. 2021. EDL-COVID: ensemble deep learning for COVID-19 case detection from chest x-ray images. *Ieee Transactions On Industrial Informatics*, 17(9): 6539–6549.

Vinuesa, R.; and Sirmacek, B. 2021. Interpretable deep-learning models to help achieve the Sustainable Development Goals. *Nature Machine Intelligence*, 3(11): 926–926.

Wang, J.; Hassan, G. M.; and Akhtar, N. 2022. A Survey of Neural Trojan Attacks and Defenses in Deep Learning. *arXiv preprint arXiv:2202.07183*.

Zeiler, M. D.; and Fergus, R. 2014. Visualizing and understanding convolutional networks. In *European conference on computer vision*, 818–833. Springer.

Zhang, J.; Bargal, S. A.; Lin, Z.; Brandt, J.; Shen, X.; and Sclaroff, S. 2018. Top-down neural attention by excitation backprop. *International Journal of Computer Vision*, 126(10): 1084–1102.

Zhou, B.; Khosla, A.; Lapedriza, A.; Oliva, A.; and Torralba, A. 2016. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2921–2929.

Zintgraf, L. M.; Cohen, T. S.; Adel, T.; and Welling, M. 2017. Visualizing deep neural network decisions: Prediction difference analysis. *arXiv preprint arXiv:1702.04595*.