Transforming gradient-based techniques into interpretable methods

Caroline Mazini Rodrigues[a,b,1], Nicolas Boutry[a], Laurent Najman[b]

[a]*Laboratoire de Recherche de l'EPITA – LRE, 14-16, Rue Voltaire, Le Kremlin-Bicêtre, 94270, France*
[b]*Univ Gustave Eiffel, CNRS, LIGM, Marne-la-Vallée, 77454, France*

Abstract

The explication of Convolutional Neural Networks (CNN) through xAI techniques often poses challenges in interpretation. The inherent complexity of input features, notably pixels extracted from images, engenders complex correlations. Gradient-based methodologies, exemplified by Integrated Gradients (IG), effectively demonstrate the significance of these features. Nevertheless, the conversion of these explanations into images frequently yields considerable noise. Presently, we introduce GAD (Gradient Artificial Distancing) as a supportive framework for gradient-based techniques. Its primary objective is to accentuate influential regions by establishing distinctions between classes. The essence of GAD is to limit the scope of analysis during visualization and, consequently reduce image noise. Empirical investigations involving occluded images have demonstrated that the identified regions through this methodology indeed play a pivotal role in facilitating class differentiation.

Keywords: explainable artificial intelligence, convolutional neural network, gradient-based, interpretability

1. Introduction

Post-hoc and *model-agnostic* explainable artificial intelligence (xAI) methods can be useful in explaining many types of general machine learning models. Within this group of methods, there are different types such as example-based explanations ([1], [2],[3]), surrogate models ([4],[5]), and influence methods ([6], [7], [8], [9], [10]). Of particular interest about the latter is that it tries to explain using the inner workings of the model being explained. Essentially, these explanations aim to reflect, to some extent, how the models think internally.

The challenge lies in the complexity of interpreting explanations derived from these methods, even when they are considered reliable. An instance of this difficulty arises with visual explanations that rely on attribution or salience maps, which can sometimes produce unclear or noisy representations. For instance, Figure 2 demonstrates noisy visualizations generated by the Integrated Gradients ([11]) technique applied to a trained ResNet-18 model [12] aiming at providing localized explanations for a sample within a dataset.

Despite the presence of noise in their visualizations, these methods continue to be widely used across domains such as medicine [13], remote sensing [14], and transportation [15]. Even though these methods are good at capturing important features for the model, we must understand their visualizations to use them effectively and safely in these applications.

In our study, we present a method to simplify gradient-based visual explanations, aiming to make it easier to interpret CNN models' reasoning. This approach relies on the concept of class distancing. The fundamental notion involves emphasizing the features crucial for distinguishing between two classes. By narrowing down the focus in this manner, our objective is to offer clearer and more streamlined visual explanations. Thus, we present the Gradient Artificial Distancing (GAD) technique, which employs a gradient-based method as the foundation of our iterative process. This methodology utilizes support regression models to reinforce the determination of important features. Each support model is trained to provide more distance between classes. Ultimately, we identify as important those features that consistently contribute to both the original model and the support models throughout this iterative process.

Additionally, to assess the reliability and human interpretability of these simplifications, we introduce an evaluation methodology. This methodology examines groups of significant pixels represented as regions (polygons), mimicking the way humans interpret these explanations. The experiments are structured into three sections: firstly, we conduct a quantitative assessment of the regions created by groups of significant pixels; secondly, we showcase the explanations derived from these clusters in the form of attribution maps; and finally, we conduct a comparative analysis between the regions generated by the original attribution maps and those generated by GAD.

Our main contributions are:

1. The GAD technique proposed to improve the interpretability of the explanations by simplifying gradient-based visualizations through the concept of class distancing analysis.
2. A methodology inspired by human cognition, to assess attribution maps based on regions.

In Section 2, we present gradient-based methods that can be used together with GAD; Section 3 presents the intuition of the method followed by the three steps of the method GAD in

*Corresponding author.
Email addresses: caroline.mazinirodrigues@esiee.fr (Caroline Mazini Rodrigues), nicolas.boutry@lrde.epita.fr (Nicolas Boutry), laurent.najman@esiee.fr (Laurent Najman)

Section 4; we describe metrics and the obtained results in Sections 5 and 6 respectively; and we conclude this paper presenting the future work in Section 7.

2. Gradient-based methods

This study targets a category of *post-hoc* influence-based explanation techniques, the gradient-based methods. Influence methods aim to elucidate the learned model by presenting the impacts of inputs or internal components on the output ([16]). These methods encompass sensitivity analysis with salience maps [6] or occlusion techniques [7], [8]; *Layer-wise Relevance Propagation* (LRP) [9]; and feature importance methods. For the purpose of this paper, we decided to concentrate on feature importance methods.

Our proposed methodology is founded on gradient-based methods. These methods fall within the category of feature importance methods, offering attribution maps that represent the importance of individual input features in relation to the output. They determine feature importance by manipulating the gradients of models. The fundamental principle behind these methods involves identifying the path that maximizes a specific output, thereby highlighting the most crucial input feature (the starting point of the path). Among the pioneering methods utilizing this approach is the Saliency visualization [6], based on gradient ascent. Other notable techniques in this category encompass Deconvolution [8], Gradients x Inputs [17], Guided Backpropagation [18], and Integrated Gradients (IG) [11].

Attributions, acquired at a pixel level, depend on the model's output, varying according to the examined class. These attributions can be visualized collectively as an image, indicating the importance of pixels in determining class decisions. Figure 2 shows the Integrated Gradients' method, depicting the most crucial pixels influencing the classification of *cat* and *dog*.

Despite outcome noise, as shown in Figure 2, these methods persist in widespread application, including medicine [13]. Wang *et al.* [19] conducted an evaluation of three xAI methods – LRP, IG, and Guided Grad-CAM – in the context of MRI Alzheimer's classification. They observed a substantial overlap among the three methods concerning brain regions, with IG showing the most promising results.

In fields like astronomy, gradient-based techniques are employed in research papers, such as the study conducted by Bhambra *et al.* [20] concerning galaxy topology classification. They applied SmoothGRAD [21] and found it to offer satisfactory explanations for their study.

Additionally, broader studies integrate gradient-based techniques in their investigations. Morrison *et al.* [22] utilize gradient methods to contrast attention-based models and CNN architectures with human perception. Woerl *et al.* [23] analyze and improve the robustness of saliency maps for data-driven explanations.

Given the prevalent application of these techniques, particularly in critical fields like medicine, we perceive an underexploration concerning the interpretation of attribution map visualizations derived from gradient-based methods. Figure 2 highlights the difficulty in understanding the model's reasoning solely through these visualizations. Despite their fidelity to the model's gradients and wide range of applications, these methods often do not produce clear and easily interpretable visualizations for humans.

To address this problem, we introduce Gradient Artificial Distancing (GAD), designed to emphasize in the attribution maps solely those image regions that significantly contribute to class differentiation. This approach aims to minimize noise, thereby enhancing human interpretability.

3. Intuition

Human perception tends to group nearby image pixels as a single entity, simplifying interpretation by reducing the number of components to analyze. Drawing clear boundaries around these components would ideally enhance the interpretability of an image region.

However, when the selected pixels lack evident proximity, defining components becomes more challenging. Consider Figure 2 as an illustration, depicting important pixels in black within the second and third images. While there's a concentrated region of pixels that might represent the cat, the majority of the images display scattered black pixels, complicating interpretation.

We believe that we can enhance the interpretability of an explanation by reducing the scattered important pixels (depicted in black) to form smaller and denser regions. However, merely applying an intensity-based filter in the explanation image is not sufficient to maintain the network's accuracy. Such a filter would solely rely on an individual image each time, disregarding the network's comprehensive knowledge.

Our idea is based on a premise: the final activations' magnitudes in a network are negligible as long as, for the same set of images, the activations' order remains consistent. In essence, consider two networks providing activations $(1,-1),(0.5,-0.6),(-0.9,0.8)$ for images $\mathbf{I}_1, \mathbf{I}_2$, and $\mathbf{I}_3$, respectively, exhibiting a similar order of activations for each class individually as another network with activations $(1,-2),(0.5,-1.6),(-1.9,0.8)$. Both networks comprehend the three images similarly.

From the perspective of representation learning, this premise holds true as it maintains the spatial arrangement, preserving each image's position relative to its neighbors (akin to a simple translation in space). Nevertheless, the second set of activations exhibits a more pronounced distinction between the two classes.

Our objective is to leverage this discrepancy between classes to minimize the number of analyzed image parts, emphasizing solely the most critical differences between classes, and reducing noise attributions, all while preserving fidelity to the original model.

4. Gradient artificial distancing

To find these most important class differences, we worked on the class activations from the training samples. Consider

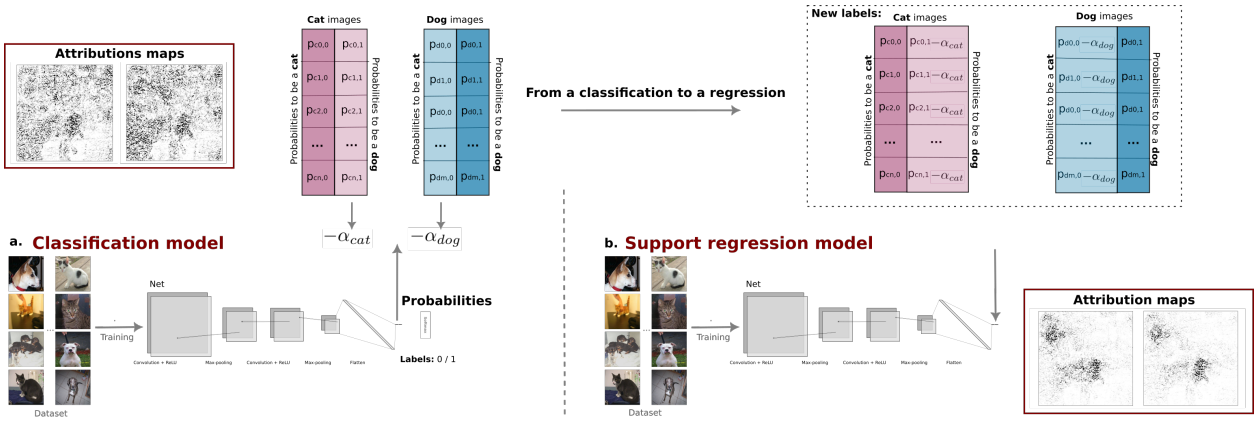

Figure 1: Based on the non-normalized probabilities (before Softmax) for both classes we distance samples from different classes. We define values α_{cat} and α_{dog} that will be subtracted from the probabilities. The value α_{cat} will reduce the probability of cat images being dogs (light pink column) and α_{dog} will reduce the probability of dog images being cats (light blue column). This new probabilities are artificial; however, they preserve relations between samples from the same class.

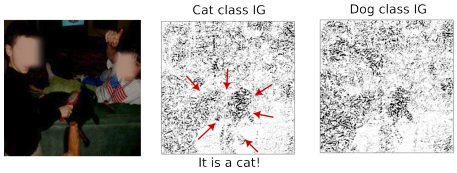

Figure 2: Attribution maps obtained by Integrated Gradients method. The pixels' importance is described from white to black (less to more important) according to a chosen class.

a classification problem with m classes $C = \{c_d\}_{d=1}^{m}$, a dataset $\mathcal{D} = \bigcup_{d=1}^{m} \mathcal{D}_{c_d}$, with n_d samples $\mathcal{D}_{c_d} = \{\mathbf{I}_{c_d,i}\}_{i=1}^{n_d}$ from the class c_d. Our goal is to accentuate class distinctions by artificially augmenting the disparity in final activations, effectively translating (distancing) the classes.

We aim to separate classes while preserving the original output space structure, maintaining the distribution of samples within the final activations' space. This strategy compels the network to focus solely on image regions crucial for creating the gap between classes, also reducing possible noisy visualizations.

With this method, we want to ensure that:

- The updated activations' output space can be approximated using the identical network architecture and the original model's weights. Consequently, the optimization process aims to yield a close approximation with the initialization of the original weights.

- The final explanation is expected to have fewer highly important pixels, but it must not introduce new crucial pixels that were absent in the original explanation. Incorporating new important pixels would deviate the explanation from the knowledge embedded within the original model.

Three steps compose this process, as developed hereafter.

4.1. Choosing classes of interest to increase distances

We selected two specific classes from the set of trained classes, C, with the aim of creating a distinct separation between them. For instance, consider the classes *cat* and *dog*. Our objective is to augment the separation by minimizing the likelihood of a dog being classified as a cat and vice-versa, all while maintaining the original learned structure.

We opted to modify the pre-Softmax activations instead of the post-Softmax ones to assess the increase in class separation, avoiding the normalization inherent in the Softmax function (which scales values between 0 and 1).

Let us call the final non-normalized activations of a model for a dataset of m classes and n images as **out**. Choosing two classes c_k and c_l we create a class artificial distancing by subtracting an α value from some activations in **out**. The new matrix **out**$'$ will have values $\mathbf{out}'_{i,l} = \mathbf{out}_{i,l} - \alpha_k$ for images $\mathbf{I}_i \in \mathcal{D}_{c_k}$ and $\mathbf{out}'_{i,k} = P_{i,k} - \alpha_l$ for images $\mathbf{I}_i \in \mathcal{D}_{c_l}$. We use this idea in Algorithm 1 to generate the artificial distances for the next step of the methodology.

For example, if c_k is the *cat* class in an image dataset and c_l the *dog* class, we directly alter the non-normalized activations of the likelihood of an image being a dog within $\mathcal{D}_{cat}$ (cat images) by subtracting a value α_{cat}. Similarly, we subtract a value α_{dog} from the probabilities of $\mathcal{D}_{dog}$ samples (dog images) being classified as cats, depicted in Figure 1**a**. This process aims to diminish the consideration of cats as dogs and vice-versa, all artificially manipulated for this analysis.

4.2. Training regressions

This artificial distancing process necessitates training new models to prompt the networks to generate these revised cat/dog output activations. These models are trained as regression problems, aiming to precisely obtain the altered artificial activations. While the inputs x remain the images from $\mathcal{D}$ and the weights are based on the analyzed classification network for initialization, the expected outputs y now represent the modified probabilities **targets** obtained by Algorithm 1. The primary goal is not to achieve more generalized models; rather, it's to attain

Algorithm 1: Artificial distance algorithm

Input: a trained model Ξ; α_k and α_l;
Data: Images $\mathbf{I}_{c_k,i} \in \mathcal{D}_{c_k}$ from class c_k and $\mathbf{I}_{c_l,j} \in \mathcal{D}_{c_l}$ from class c_l;
Output: Regression targets $\mathbf{targets}_{k,l}$ for images in $\mathcal{D}_{c_k}$ and $\mathcal{D}_{c_l}$.

1 $\mathbf{out}_k := \Xi(\mathcal{D}_{c_k})$ `// outputs from images in class` c_k
2 $\mathbf{out}_l := \Xi(\mathcal{D}_{c_l})$ `// outputs from images in class` c_l
3 $\mathbf{out}_k(., l) := \mathbf{out}_k(., l) - \alpha_k$ `// reducing` l `in class` k
4 $\mathbf{out}_l(., k) := \mathbf{out}_l(., k) - \alpha_l$ `// reducing` k `in class` l
5 $\mathbf{targets}_{k,l} := concatenate(\mathbf{out}_k, \mathbf{out}_l)$

the most accurate approximations of these new outputs, thereby discerning the persistently important features from the original explanation. To facilitate this, we replace the Softmax activation with a Linear one and employ Mean Squared Error (MSE) for regression training. Figure 1**b** illustrates this transformation in the model learning process, with labels converted into a probability vector.

For every different pair of values $(\alpha_k, \alpha_l)_s$, we will generate new target values $\mathbf{targets}_s$ based on Algorithm 1 to train a distinct model Ξ_s for regression, as explained earlier. These models will serve as the support regression models.

4.3. Choosing important features

For precision in identifying crucial features, we propose iterating through steps 1 and 2 (Sections 4.1 and 4.2) multiple times, progressively increasing the values of α_k and α_l and training for each couple of α a support regression model. Upon their training, each network is subject to the chosen Gradient-based xAI method when examining images. The significant features extracted should mirror those of the original model under explanation. However, we anticipate that the consistently important features across all networks will hold greater significance in differentiating the two classes. Figure 3 illustrates the utilization of four support regression networks to delineate important regions. The analyzed image, in which Integrated Gradients (IG) was applied, was erroneously classified as a dog by the network. We show the in Algorithm 2 how to combine the explanations of each support regression model to obtain a final attribution map **attr**.

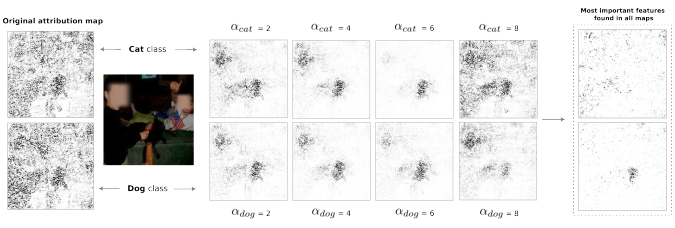

Figure 3: Example of choosing important features. After training regression networks, we apply the IG method for each resulting model and both classes. The final attribution map includes only features present in all five attribution maps, the initial one (classification model analyzed), and the four support regression models. We obtain filtered attribution maps, including the regions of the image that most separate classes.

Algorithm 2: Features selection algorithm

Input: Original model Ξ, set of trained support regression models $\{\Xi_0, ..., \Xi_s\}$, xAI gradient-based technique *explain*;
Data: Image $\mathcal{I}_i$ to be explained;
Output: Attribution map **attr**.

1 $\mathbf{attr}_{orig} := explain(\Xi, \mathcal{I}_i)$ `// original attribution map`
2 $\mathbf{attr}_{aux}(\mathbf{attr}_{orig} > 0) := 1$ `// 1's if positive attributions 0's otherwise`
`/* for each support regression network */`
3 **for** $\Xi_j \in \{\Xi_0, ..., \Xi_s\}$ **do**
4 $\quad \mathbf{attr}_j := explain(\Xi_j, \mathcal{I}_i)$ `// obtain attribution map`
5 $\quad \mathbf{attr} := \mathbf{attr}_j \times \mathbf{attr}_{aux}$ `// maintain attributions also in previous map`
6 $\quad \mathbf{attr}_{aux}(\mathbf{attr} > 0) := 1$ `// update` $\mathbf{attr}_{aux}$ `as line 2`

4.4. Extension to a multi-class problem

Until now, we've outlined the methodology applied to a two-class problem. In this section, we explore two approaches for adapting GAD to a multi-class problem.

One vs. all (OvA): With this strategy, we would compare one specific class, c_k, against all other classes. The adaptation involves selecting c_k as the target class and treating all other images as belonging to a single class, c_l, in Algorithm 1. This approach would emphasize the distinct characteristics of c_k while contrasting them with the features of all other classes.

Split output space (Half): With this strategy, we aim to create a more gradual effect on the output space compared to the **One vs. All** approach. The concept involves separating classes into two clusters based on their proximity in the output space, and then applying the artificial distancing (Algorithm 1) between these two clusters. For instance, in a four-class problem, suppose classes 0 and 2 form one cluster (corresponding to c_k), while classes 1 and 3 form another cluster (corresponding to c_l), based on their output similarities. We subtract α_k from the output of images belonging to classes 0 and 2 at positions 1 and 3 (to reduce the influence of c_l): $\mathbf{out}_k(., [1, 3]) := \mathbf{out}_k(., [1, 3]) - \alpha_k$ (line 3 of Algorithm 1). We do a similar process for classes 1 and 3: $\mathbf{out}_l(., [0, 2]) := \mathbf{out}_l(., [0, 2]) - \alpha_l$ (line 4 Algorithm 1).

5. Metrics for evaluation

We propose to evaluate our method using two criteria: *Complexity* and *Sensitivity*.

Regarding *complexity*, our aim is to prioritize visualizations that are less complex, indicating reduced noise levels. In terms of *sensitivity*, we seek to assess the consequences of occluding essential features on the model's output. We compare occluding important features identified by GAD (our method) against occluding the remaining important features – constituting a supplementary set found in the original explanation but absent in GAD's explanation. The anticipation is that with GAD, fewer important features will be identified, potentially yielding a more pronounced impact on the output.

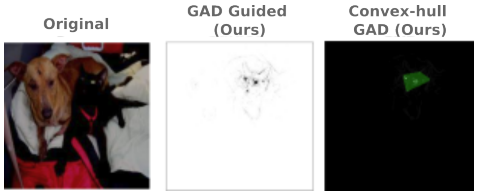

Figure 4: Example of convex hull applied to an attribution map. We involve with the convex-hull the pixels reaching more than 10% of the highest importance according to the GAD – Guided Propagation attribution map.

As mentioned earlier, we aim to simulate human perception, which tends to group nearby pixels as one unit. To account for this in our evaluation, we create a convex hull involving a selected set of the most important pixels. Figure 4 illustrates an instance of the produced convex hull. Therefore, concerning *complexity*, we assess the ratio *RC* by comparing the area of the convex hull in the original explanation ($A(CH_{Orig})$) to the area of the convex hull in the GAD explanation ($A(CH_{\text{GAD}})$):

$$RC = \frac{A(CH_{\text{GAD}})}{A(CH_{Orig})} \tag{1}$$

where $0 < RC < 1.0$ represents a smaller important region for GAD explanation.

Regarding *sensitivity*, we leverage the created convex hulls to conduct occlusions. These convex hulls, denoted as CH_{Orig} and CH_{GAD}, are represented in image masks M_{Orig} and M_{GAD}, respectively, with a value of 1. Internal or boundary pixels within the convex hulls are assigned a value of 1, while the remaining pixels receive 0 (forming a binary mask). Utilizing these masks, we generate two occlusions: one using M_{GAD} and another employing $M_{Orig} - M_{\text{GAD}}$. Our anticipation is that the occlusion using M_{GAD} will exert a more substantial impact.

To quantify this impact, we propose calculating the ratio *RS*, derived from the difference between the output before and after occlusion, divided by the area of the occluded section:

$$RS(\mathbf{I}, M, a) = \frac{|N_a(\mathbf{I}) - N_a(\mathbf{I} \times (\mathbf{1} - M^{(a)}))|}{\sum_{i=0}^{m}\sum_{j=0}^{n} M_{i,j}^{(a)}} \tag{2}$$

where $N_a(\mathbf{I})$ is the output a of the network for an image $\mathbf{I} \in \mathcal{D}$ of size $m \times n$ and $M_{i,j}^{(a)}$ is the pixel (i, j) from a mask of size $m \times n$ corresponding to the output a. We expect $RS(\mathbf{I}, M_{\text{GAD}}, a) > RS(\mathbf{I}, M_{orig} - M_{\text{GAD}}, a), \forall a \in C$ (for all classes).

6. Experiments and results

We experimented with two CNN architectures: VGG16 [24] and ResNet18 [12], utilizing two distinct datasets. The first dataset involves a binary classification of cats and dogs [1], while the second dataset, the CUB-200-2011 dataset [25], focuses on bird classification. For the bird classification task, we specifically chose the *warbler* and *sparrow* species as the two classes (the original dataset comprises 20 classes each, which we put together for this analysis).

We applied values $\alpha = 0, 2, 4, 6, 8$ for both classes. The support regression networks initiated the training using weights from the analyzed classification model (original model). Each support model underwent training for 10 epochs, utilizing the Adam optimizer and a learning rate set to 4×10^{-5}.

We conducted experiments applying GAD to five explanation techniques commonly used in the literature: Saliency [6], Deconvolution [8], Gradient x Input [17], Guided-Backpropagation [18], and Integrated Gradients [11]. These techniques served as the xAI technique *explain* in Algorithm 2. We present the results of GAD alongside the original explanations produced by each xAI technique for comparison.

In the subsequent experiments on *complexity* and *sensitivity*, we employed a total of 512 images (for both datasets), evenly distributed with 256 images per class. To create the convex hull, we filtered out 50% of the pixels, retaining only the most important half.

6.1. Complexity

Following the computation of *RC* (Equation 1) for all 512 images, we count the number of images where $RC < 1.0$, indicating a point in favor of GAD, and those where $RC \geq 1.0$, indicating a point in favor of the original visualization.

Table 1: For most methods, GAD is able to reduce interest area in visualizations. Evaluation of *RC* for five literature methods, Saliency (S), Deconvolution (D), Gradient x Input (GxI), Guided-Backpropagation (GB) and Integrated Gradients (IG); two architectures, ResNet18 and VGG16; and two datasets, Cat x Dog and Birds. Orig.1 and Orig.2 represent the original methods' visualization for the two classes. GAD 1 and GAD 2 are the application of our method to both classes visualizations.

			S	D	GxI	GB	IG
Cat x Dog	VGG	**Orig.1**	**357**	0	15	4	17
		GAD **1 (ours)**	155	**512**	**490**	**427**	**480**
		Orig.2	159	0	2	3	0
		GAD **2 (ours)**	**348**	**512**	**494**	**428**	**486**
	ResNet	**Orig.1**	122	0	2	10	7
		GAD **1 (ours)**	**390**	**512**	**492**	**453**	**497**
		Orig.2	130	0	3	5	2
		GAD **2 (ours)**	**380**	**512**	**490**	**460**	**488**
Birds	VGG	**Orig.1**	**270**	0	9	0	1
		GAD **1 (ours)**	240	**511**	**488**	**450**	**500**
		Orig.2	**301**	0	10	0	0
		GAD **2 (ours)**	210	**510**	**494**	**451**	**504**
	ResNet	**Orig.1**	153	0	1	0	0
		GAD **1 (ours)**	**358**	**512**	**504**	**472**	**506**
		Orig.2	174	0	2	0	0
		GAD **2 (ours)**	**337**	**512**	**506**	**470**	**504**

We show, in Table 1, the number of images where GAD (**Ours**) and the original method exhibited smaller complexity (smaller important areas). Through the experiments involving five literature methods – Saliency (S), Deconvolution (D), Gradient x Input (GxI), Guided-Backpropagation (GB), and Integrated Gradients (IG) – we observed that GAD successfully reduces the area of importance for nearly all images and methods.

6.2. Sensitivity

Upon computing $RS(\mathbf{I}, M, a)$ (Equation 2) for the four occlusion masks, which include two masks per class a (M_{GAD} and

[1] https://www.kaggle.com/competitions/dogs-vs-cats-redux-kernels-edition/data

$M_{Orig} - M_{\text{GAD}}$), we acquire four respective *RS* values each time. We designated the mask $M_{Orig} - M_{\text{GAD}}$ as the *supplementary* mask, denoted as **Sup.**, and M_{GAD} simply as GAD. Therefore, the resulting four values correspond to: $RS(\mathbf{I}, \text{GAD}_1, 1)$, $RS(\mathbf{I}, \text{GAD}_2, 2)$, $RS(\mathbf{I}, Sup._1, 1)$, and $RS(\mathbf{I}, Sup._2, 2)$.

Table 2: Four out of five literature methods indicate a bigger impact when occluding the important regions found by GAD. Evaluation of *RS* for five literature methods, Saliency (S), Deconvolution (D), Gradient x Input (GxI), Guided-Backpropagation (GB) and Integrated Gradients (IG); two architectures, ResNet18 and VGG16; and two datasets, Cat x Dog and Birds. Sup.1, Sup.2, GAD 1, and GAD 2 indicate the average *RS* values using these for different masks for occlusion. Sup.1 and Sup.2 represent the supplementary masks for the two classes. GAD 1 and GAD 2 are our method masks for both classes. Values in 10^2 scale.

			S	D	GxI	GB	IG
Cat x Dog	VGG	**Sup.1**	0.0800	0.0029	0.0052	0.1100	0.0100
		GAD **1 (ours)**	**0.1300**	**0.0056**	**0.1200**	**1.5900**	**0.1300**
		Sup.2	0.2400	0.0029	0.0051	0.1500	0.0096
		GAD **2 (ours)**	**0.2500**	**0.0062**	**0.5600**	**1.8200**	**0.4900**
	ResNet	**Sup.1**	**0.0600**	0.0013	0.0016	0.1300	0.0016
		GAD **1 (ours)**	0.0300	**0.0024**	**0.0700**	**0.5100**	**0.1200**
		Sup.2	0.0500	0.0012	0.0016	0.1000	0.0020
		GAD **2 (ours)**	**0.0700**	**0.0024**	**0.1600**	**0.4700**	**0.1600**
Birds	VGG	**Sup.1**	**0.2500**	0.0027	0.0800	0.0100	0.0100
		GAD **1 (ours)**	0.2000	**0.0056**	**0.5400**	**1.2800**	**0.5500**
		Sup.2	**0.1500**	0.0030	0.0092	0.0100	0.0072
		GAD **2 (ours)**	0.1300	**0.0068**	**0.3600**	**1.4100**	**0.3800**
	ResNet	**Sup.1**	**0.1100**	0.0020	0.0045	0.0100	0.0094
		GAD **1 (ours)**	0.0400	**0.0048**	**0.1700**	**1.0300**	**0.2900**
		Sup.2	0.0500	0.0019	0.0037	0.0100	0.0026
		GAD **2 (ours)**	0.0500	**0.0048**	**0.1600**	**1.0400**	**0.2000**

We show, in Table 2, the average *RS* values across all 512 images for GAD 1, GAD 2, Sup.1, and Sup.2, for the five literature methods, two architectures, and two datasets. For Deconvolution, Gradient x Input, Guided-Backpropagation, and Integrated Gradients, our method (GAD) effectively identifies the most impactful regions for each decision, excluding the supplementary area (from the original visualization) that does not exhibit a greater impact (per pixel). However, in the case of Saliency visualization, both the areas selected by GAD and the supplementary areas display similar impacts, indicating limited benefits of our method for this particular technique.

6.3. GAD *attribution maps*

We present in Figure 5 the qualitative results of employing GAD with five gradient-based techniques. Each set of two rows illustrates the original attribution maps in the first row and the corresponding GAD maps in the second row. These pairs of rows represent the tested methods: Saliency, Deconvolution, Gradient x Input, Guided-Backpropagation, and Integrated Gradients. The results for VGG are displayed on the right side, while those for ResNet are on the left. The classification results (cat or dog) are presented at the bottom. Notable examples highlight the enhancements in visualizations: the original attribution maps are represented in blue, and the corresponding GAD visualizations are shown in red. For instance, Image 3 (highlighted in both networks) demonstrates a noteworthy distinction: both VGG and ResNet original attribution maps (in blue) appear similar in identifying the image as a dog. However, upon applying GAD, distinct attribution maps (in red) emerge, with VGG focusing on the dog while ResNet emphasizes the cat, despite both networks classifying the image as a dog.

6.4. Finding visual knowledge clues

Figure 6 illustrates a comparative visualization between original and GAD-interest areas. Aligned with the metrics detailed in Section 5, we generated the convex hulls involving the most important pixels per visualization. We show the five less activated images for each assigned class (original label), for VGG (on the right) and ResNet (on the left), within the bird dataset (at the top), and the cat vs. dog dataset (at the bottom). In these visualizations, the GAD-obtained regions are highlighted in green, while the original regions are marked in red. We expect small green regions.

Additionally, we aimed to comprehend why these images were the least activated for their respective classes. While the bird dataset displays consistent behavior, the cat vs. dog dataset contains some images featuring both animals and occasionally people. Specifically, for VGG, the initial image showcases both animal types, with the dog (as per the larger green area) deemed more important than the cat. The final image features both a dog and Santa Claus, where both the animal and the person are deemed important. Regarding ResNet, the last image is intriguing as the dog is correctly identified as an important region, but the vase's presence likely diminishes the dog's activation for the dog class. In both models, the second-worst image is the same, containing both animal types, exemplifying differences between the models: VGG appears to recognize parts of both animal types, whereas ResNet emphasizes the cat, despite the dog's prominence impacting the cat's activation.

6.5. Extension for a multi-class problem

To test the multi-class extension idea presented in Section 4.4, we employed the VGG architecture trained on the CIFAR-10 dataset [26, 27] and Integrated Gradients (IG) explanation as base method. Results of the **OvA** and **Half** adaptations are illustrated in Figure 7.

We observed two key points: firstly, both adaptations (**OvA** and **Half**) effectively reduce noise in the explanations. Secondly, **OvA** appears to yield more discernible features from the class with diminished regions of interest, as evidenced by the highlighted bird and deer in red.

6.6. Discussion of the findings

Reducing the sparsity of the most important visualized areas enhances interpretability, enabling humans to focus on smaller, more informative regions. Our experiments (Figures 5 and 7) demonstrate that GAD effectively minimizes noise in explanation visualizations and highlights less sparse regions of interest. These areas are vital for the model's performance, as evidenced by the *Sensitivity* experiment (Table 2). Moreover, our *Complexity* experiment (Table 1 and Figure 6) underscores GAD's ability to produce concise explanations. Future work will involve utilizing density metrics, such as density histograms, and human-based evaluations to further validate interpretability quality metrics.

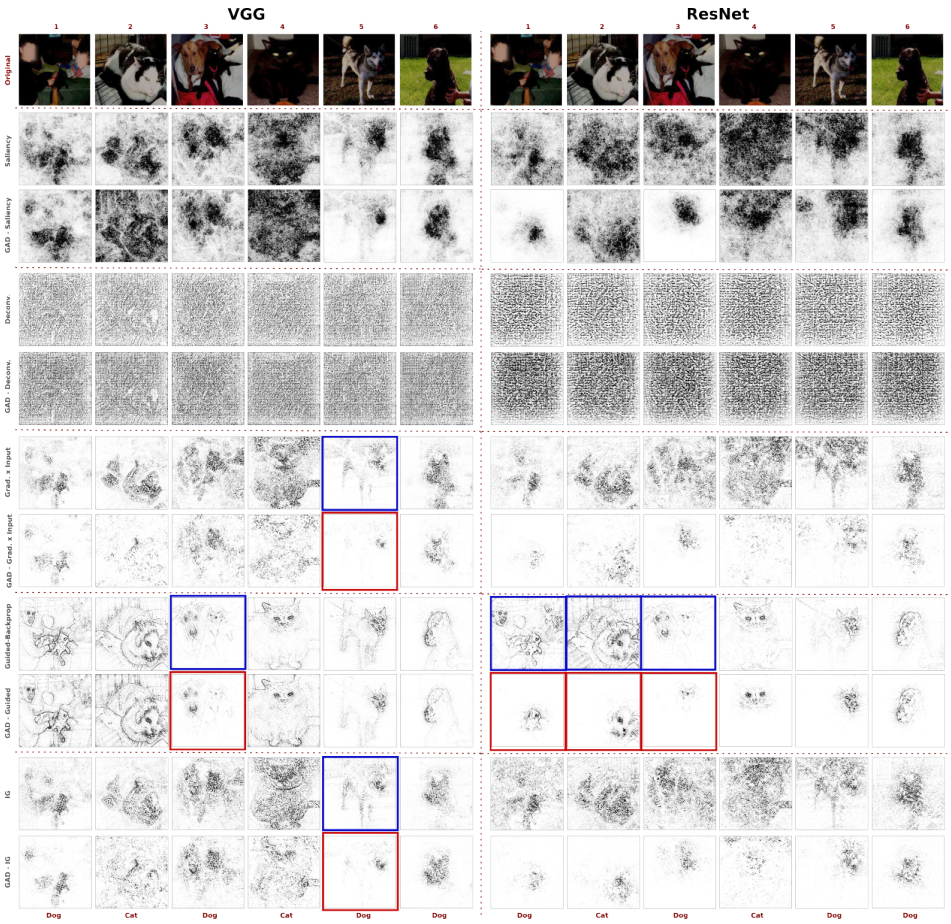

Figure 5: **Our final visualizations:** GAD improves interpretability of attribution maps. We present two-by-two rows of original attribution maps and GAD maps from five gradient-based techniques: Saliency, Deconvolution, Gradient x Input, Guided-Backpropagation, and Integrated Gradients. We present on the right side the VGG results and on the left side the ResNet ones. At the bottom, we present the obtained classification for each image (cat or dog).

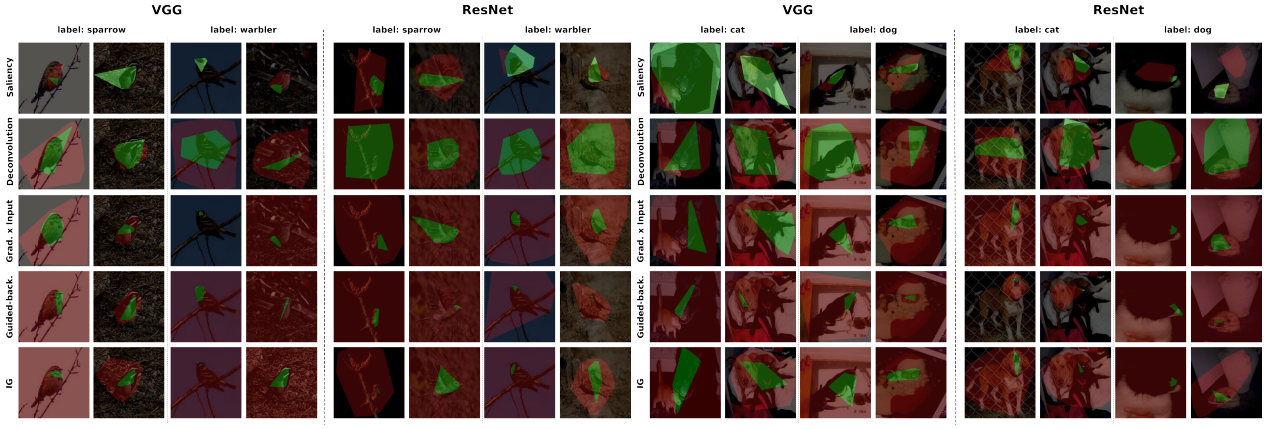

Figure 6: **Convex-hull comparisons for evaluation:** The convex-hull of GAD's important features considerably reduces the images' interest regions. The comparison between original and GAD convex hulls is able to give insights into why misclassifications happen. We show, in red, the original convex hull, and in green, the GAD convex hull, for the five literature techniques (IG, Guided-Backpropagation, Gradient x Input, Deconvolution, and Saliency), the two architectures and the two datasets. We chose to visualize the less activated images for the corresponding label class in each model.

7. Conclusion

The experiments demonstrated GAD's efficacy in pinpointing the key regions influencing class relationships. Through mask-based occlusions, GAD effectively highlighted pivotal areas in decision-making, even when dealing with smaller significant regions. Across datasets and models, GAD reduced complexity and enhanced the sensitivity of five gradient-based explainability techniques. These findings deepened our interpretation of the networks' knowledge and provided insights into

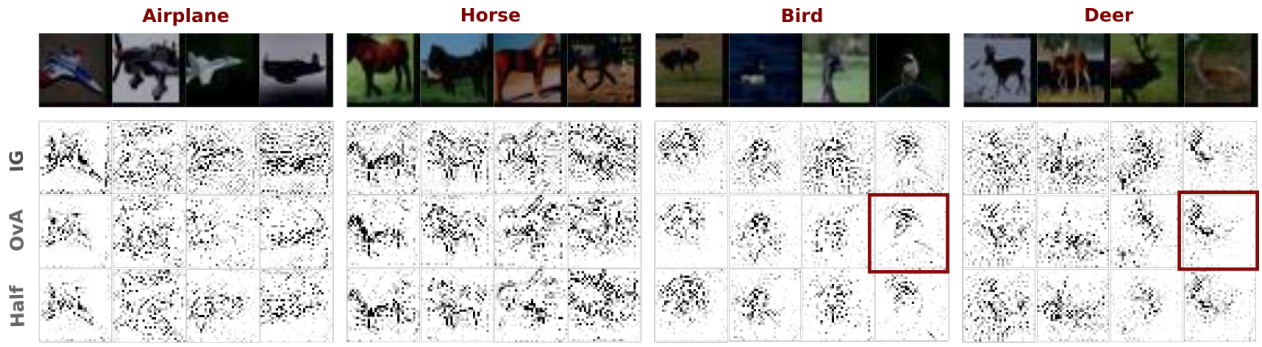

Figure 7: Examples of the multi-class GAD extension applied to Integrated Gradients (IG). We showcase the results of two strategies: One-vs-All (**OvA**) and Split output space (**Half**) for four classes from the CIFAR-10 dataset, using a VGG16 trained model. We observe a reduction in noise for both strategies, with smaller regions of interest highlighted by **OvA**.

the underlying causes of certain misclassification errors. Experiments in multi-class classification underscored the potential of GAD in enhancing explanations and reducing noise. Overall, GAD offers a straightforward approach to improving the interpretability of gradient-based methods for both binary and multi-class models by minimizing noise and directing attention to specific areas of interest. However, the method has its limitations, its interpretability hinges on human analysis. Consequently, if there is an overlap of important patterns, it may not be clear how to interpret and prioritize them. In future work, we aim to explore the automation of values for α, additional metrics for interpreting quality, such as density measures and human-based evaluations, and a mechanism for clustering similar concepts (account for semantics) in different images to mitigate ambiguity of interpretation arising from overlapping important patterns.

References

[1] D. Erhan, Y. Bengio, A. Courville, P. Vincent, Visualizing higher-layer features of a deep network, Technical Report, Univeristé de Montréal (2009) 1–13.

[2] J. Bien, R. Tibshirani, Prototype selection for interpretable classification, Annals of Applied Statistics 5 (4) (2012) 1–23.

[3] B. Kim, R. Khanna, O. Koyejo, Examples are not enough, learn to criticize! Criticism for interpretability, in: 30th International Conference on Neural Information Processing Systems (NeurIPS), 2016, pp. 2288–2296.

[4] M. T. Ribeiro, S. Singh, C. Guestrin, "Why Should I Trust You?": Explaining the predictions of any classifier, in: 22nd International Conference on Knowledge Discovery and Data Mining (KDD), 2016, pp. 1135–1144.

[5] J. J. Thiagarajan, B. Kailkhura, P. Sattigeri, K. N. Ramamurthy, Treeview: Peeking into deep neural networks via feature-space partitioning (2016).

[6] K. Simonyan, A. Vedaldi, A. Zisserman, Deep inside convolutional networks: Visualising image classification models and saliency maps, in: Workshop at 2nd International Conference on Learning Representations (ICLR), 2014, pp. 1–8.

[7] L. Merrick, Randomized ablation feature importance (2019).

[8] M. D. Zeiler, R. Fergus, Visualizing and understanding convolutional networks, in: 13th European Conference on Computer Vision (ECCV), 2014, pp. 818–833.

[9] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Muller, W. Samek, On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation, PLoS One 10 (7) (2015) 1–46.

[10] S. M. Lundberg, S.-I. Lee, A unified approach to interpreting model predictions, in: 31st International Conference on Neural Information Processing Systems (NeurIPS), 2017, pp. 4768–4777.

[11] M. Sundararajan, A. Taly, Q. Yan, Axiomatic attribution for deep networks, in: 34th International Conference on Machine Learning (ICML), 2017, pp. 3319–3328.

[12] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: 29th IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778.

[13] K. Borys, Y. A. Schmitt, M. Nauta, C. Seifert, N. Krämer, C. M. Friedrich, F. Nensa, Explainable ai in medical imaging: An overview for clinical practitioners – saliency-based xai approaches, European Journal of Radiology 162 (2023) 110787.

[14] I. Kakogeorgiou, K. Karantzalos, Evaluating explainable artificial intelligence methods for multi-label deep learning classification tasks in remote sensing, International Journal of Applied Earth Observation and Geoinformation 103 (2021) 102520.

[15] H.-S. Kim, I. Joe, An xai method for convolutional neural networks in self-driving cars, PLOS ONE 17 (8) (2022) 1–17.

[16] A. Adadi, M. Berrada, Peeking inside the black-box: A survey on explainable artificial intelligence (XAI), IEEE Access 6 (2018) 1–23.

[17] A. Shrikumar, P. Greenside, A. Kundaje, Learning important features through propagating activation differences, in: 34th International Conference on Machine Learning (ICML), 2017, pp. 3145–3153.

[18] J. T. Springenberg, A. Dosovitskiy, T. Brox, M. Riedmiller, Striving for simplicity: The all convolutional net, in: 3rd International Conference on Learning Representations (ICLR), 2015, pp. 1–14.

[19] D. Wang, N. Honnorat, P. T. Fox, K. Ritter, S. B. Eickhoff, S. Seshadri, M. Habes, Deep neural network heatmaps capture alzheimer's disease patterns reported in a large meta-analysis of neuroimaging studies, NeuroImage 269 (2023) 1–12.

[20] P. Bhambra, B. Joachimi, O. Lahav, Explaining deep learning of galaxy morphology with saliency mapping, Monthly Notices of the Royal Astronomical Society 511 (4) (2022) 5032–5041.

[21] D. Smilkov, N. Thorat, B. Kim, F. Viégas, M. Wattenberg, Smoothgrad: removing noise by adding noise, in: arXiv, 2017.

[22] K. Morrison, A. Mehra, A. Perer, Shared interest...sometimes: Understanding the alignment between human perception, vision architectures, and saliency map techniques, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, 2023, pp. 3776–3781.

[23] A.-C. Woerl, J. Disselhoff, M. Wand, Initialization noise in image gradients and saliency maps, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2023, pp. 1766–1775.

[24] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, in: 3rd International Conference on Learning Representations (ICLR), 2015.

[25] X. He, Y. Peng, Fine-grained visual-textual representation learning, IEEE Transactions on Circuits and Systems for Video Technology PP (2019) 1–12.

[26] A. Krizhevsky, Learning multiple layers of features from tiny images, Tech. rep. (2009).

[27] A. Krizhevsky, V. Nair, G. Hinton, Cifar-10 (canadian institute for advanced research).