Flávio Arthur Oliveira Santos

Advancing Deep Learning Models for Robustness and Interpretability in Image Recognition

Universidade Federal de Pernambuco
posgraduacao@cin.ufpe.br
http://cin.ufpe.br/~posgraduacao

Recife
2023

Flávio Arthur Oliveira Santos

Advancing Deep Learning Models for Robustness and Interpretability in Image Recognition

Tese de Doutorado apresentada ao Programa de Pós-graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, como requisito parcial para obtenção do título de Doutor em Ciência da Computação.

Área de Concentração: Inteligência computacional
Orientador: Dr. Cleber Zanchettin
Coorientador: Dr. Paulo Jorge Freitas de Oliveira Novais

Recife
2023

Catalogação na fonte
Bibliotecário Josias Machado da Silva Junior, CRB4-1690

S237a Santos, Flávio Arthur Oliveira
Advancing deep learning models for robustness and interpretability in image recognition / Flávio Arthur Oliveira Santos – 2024.
113 f.: il., fig., tab.

Orientador: Cleber Zanchettin
Coorientador: Paulo Jorge Freitas de Oliveira Novais
Tese (Doutorado) – Universidade Federal de Pernambuco. CIn, Ciência da Computação, Recife, 2024.
Inclui referências e apêndice.

1. Deep learning. 2. Robustez. 3. Ataques adversários. 4. Interpretabilidade. I. Zanchettin, Cleber (orientador). II. Novais, Paulo Jorge Freitas de Oliveira (coorientador). III. Título.

006.31 CDD (23. ed.) UFPE - CCEN 2024 – 39

Flávio Arthur Oliveira Santos

“Advancing Deep Learning Models for Robustness and Interpretability in Image Recognition”

Tese de Doutorado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de Doutor em Ciência da Computação. Área de Concentração: Inteligência Computacional

Aprovada em: 06/12/2023.

_____

Orientador: Prof. Dr. Cleber Zanchettin

BANCA EXAMINADORA

_____

Prof. Dr. Tsang Ing Ren
Centro de Informática/ UFPE

_____

Prof. Dr. Ricardo Matsumura Araújo
Centro de Desenvolvimento Tecnológico / UFPel

_____

Prof. Dr. Leonardo Nogueira Matos
Departamento de Computação / UFS

_____

Profa. Dra. Dalila Duraes
Departamento de Informática / Universidade do Minho

_____

Prof. Dr. Byron Leite Dantas Bezerra
Escola Politécnica de Pernambuco / UPE

ACKNOWLEDGEMENTS

Firstly, I express my deep gratitude to my advisor, Professor Cleber Zanchettin, for all the patience, corrections on papers and documents (many sent at the last minute), and enriching discussions throughout this thesis. His dedication, combined with the freedom of research, was crucial for the development of this work.

I also thank my co-advisor, Professor Paulo Novais, and my friends from the ISLab at the University of Minho. You were essential during my time in Portugal, providing an important and enriching experience.

My thanks extend to my colleagues, Professor Luís A. Nunes Amaral and Dr. Weihua Lei, from Northwestern University. Your approach to Deep Learning and enriching discussion (primarily with Dr. Weihua Lei) was crucial for my growth as a researcher.

To the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), I am grateful for the scholarship, a key support for the development of this work.

I express my gratitude to all collaborators of the Centro de Informática at UFPE. Having access to this infrastructure, operational 24 hours a day, and having an administrative sector that simplifies bureaucracy is crucial for the development of research.

Finally, but not least, I want to thank a group of people who were crucial throughout this process: my family. To my parents, Jorgeval de Andrade Santos and Rosilda Santos Oliveira, and my sister, Danielly Oliveira Santos, I thank you for always offering me support, encouragement, and freedom in my decisions. This was crucial for everything. I also express my gratitude to my brother, Kleber Tarcisio Oliveira Santos, for introducing me to Computer Science; undoubtedly, without this step, I would not be here. Finally, I thank my wife, Maynara Donato de Souza, for our discussions about Mathematics and Deep learning, her patience during the hard times, her support, and her constant presence, even when the Atlantic Ocean separated us.

ABSTRACT

Deep Learning architectures are among the most promising machine learning models today. They are used in various domains, including drug discovery, speech recognition, object recognition, question and answer, machine translation, and image description. Surprisingly, some studies even report superhuman performance, that is, a level of performance superior to human experts in certain tasks. Although these models exhibit high precision and coverage, the literature shows that they also have several limitations: (1) they are vulnerable to adversarial attacks, (2) they have difficulty inferring data outside the training distribution, (3) they provide correct inferences based on spurious information, and (4) their inferences are difficult for a domain expert to interpret. These limitations make it challenging to adopt these models in high-risk applications, such as autonomous cars or medical diagnostics. Overcoming these limitations requires robustness, reliability, and interpretability. This thesis conducts a comprehensive exploration of techniques and tools to improve the robustness and interpretability of Deep Learning models in the domain of image processing. These contributions cover four key areas: (1) the development of the Active Image Data Augmentation (ADA) method to improve model robustness, (2) the proposition of the Adversarial Right for Right Reasons (ARRR) loss function to ensure that models are "right for the right reasons" and adversarially robust, (3) the introduction of the Right for Right Reasons Data Augmentation (RRDA) method, which improves the context of the information to be represented among the training data to stimulate the model's focus on signal characteristics, and (4) the presentation of a new method for interpreting the behavior of models during the inference process. We also present a tool for manipulating visual features and assessing the robustness of models trained under different usage situations. The analyses demonstrate that the ADA method improves the robustness of models without compromising traditional performance metrics. The ARRR method demonstrates robustness against the color bias of images in problems based on the structural information of the images. In addition, the RRDA method significantly improves the model's robustness in relation to background shifts in the image, outperforming the performance of other traditional RRR methods. Finally, the proposed model analysis tool reveals the counterintuitive interdependence of features and assesses weaknesses in the models' inference decisions. These contributions represent significant advances in Deep Learning applied to image processing, providing valuable insights and innovative solutions to challenges associated with the reliability and interpretation of these complex models.

Key-words: deep Learning; robustness; adversarial attacks; interpretability.

RESUMO

As arquiteturas de *Deep Learning* estão entre os modelos de aprendizado de máquina mais promissores na atualidade. Elas são utilizadas em diversos domínios, incluindo descoberta de medicamentos, reconhecimento de fala, reconhecimento de objetos, perguntas e respostas, tradução de automática e descrição de imagens. Surpreendentemente, alguns estudos relatam até mesmo desempenho super-humano, ou seja, um nível de desempenho superior ao de especialistas humanos em determinadas tarefas. Embora esses modelos exibam alta precisão e cobertura, a literatura mostra que também possuem várias limitações: (1) são vulneráveis a ataques adversários, (2) possuem dificuldade em inferir dados fora da distribuição de treinamento, (3) fornecem inferências corretas com base em informações espúrias e, além disso, (4) suas inferências são de difícil interpretação por um especialista do domínio. Essas limitações tornam desafiador adotar esses modelos em aplicações de alto risco, como carros autônomos ou diagnósticos médicos. A superação destas limitações demanda robustez, confiabilidade e interpretabilidade. Nesta tese, é realizada uma exploração abrangente de técnicas e ferramentas, voltadas para aprimorar a robustez e interpretabilidade de modelos de *Deep Learning* no domínio de processamento de imagens. Essas contribuições abrangem quatro áreas-chave: (1) o desenvolvimento do método de aumento de dados de imagem ativo (ADA) para melhorar a robustez do modelo, (2) a proposição da função de perda *adversarial right for right reasons* (ARRR) para garantir que os modelos estejam "certos pelos motivos certos" e adversarialmente robustos, (3) a introdução do método de aumento de dados *right for right reasons* (RRDA) que melhora dentre os dados de treinamento o contexto das informações a serem representadas para estimular o foco do modelo em características de sinal, e (4) a apresentação de um novo método para interpretar o comportamento dos modelos durante o processo de inferência. Apresentamos ainda uma ferramenta para manipular características visuais e avaliar a robustez dos modelos treinados sob diferentes situações de uso. As análises realizadas demonstram que o método ADA melhora a robustez dos modelos sem comprometer métricas tradicionais de desempenho. O método ARRR demonstra robustez ao viés de cor das imagens em problemas baseados em informações estruturais das imagens. Além disso, o método RRDA melhora significativamente a robustez do modelo em relação a deslocamentos de fundo da imagem, superando o desempenho de outros métodos RRR tradicionais. Finalmente, a ferramenta de análise de modelos proposta permite revelar a interdependência contraintuitiva de características e avaliar fraquezas nas decisões de inferência dos modelos. Estas contribuições representam avanços significativos no campo do Deep Learning aplicado ao processamento de imagens, fornecendo *insights* valiosos e soluções inovadoras para desafios associados à confiabilidade e interpretação desses modelos complexos.

Palavras-chaves: deep Learning; robustez; ataques adversários; Interpretabilidade.

LIST OF FIGURES

LIST OF TABLES

SPECIAL TERMS

ADA Active Image Data Augmentation

Ada-SISE Adaptive Semantic Input Sampling for Efficient Explanation of Convolutional Neural Networks

ARRR Adversarial Right for the Right Reasons

ATCNN Adversarial Trained Convolutional Neural Networks

BIG Blur Integrated Gradient

CD Contextual Decomposition

CDEP Contextual Decomposition Explanation Penalization

CLIP Contrastive Language-Image Pretraining

CNN Convolutional Neural Network

DL Deep learning

DSC Dice score

DTD Deep Taylor Decomposition

EG Expected Gradient

FAT Free Adversarial Training

FGSM Fast Gradient Sign Method

FII Feature Impact index

Grad-Cam Gradient-Weighted Class Activation Mapping

HC Hill Climbing

IG Integrated Gradients

IPHA Iterative post-hoc attribution

LLM Large Language Model

LRP Layer-wise relevance propagation

MLP Multilayer-Perceptron

MRI Magnetic Resonance Imaging

PA PatternAttribution

PGD Projected Gradient Descent

PN PatternNet

RRDA Right Reasons Data Augmentation

RRR Right for the right reasons

SCGM Spinal Cord Grey Matter Segmentation

SGD Stochastic gradient descent

SISE Semantic Input Sampling for Explanation

UA U Analysis

VLM Visual Language Model

XIL Explanatory Interactive Learning

CONTENTS

1 INTRODUCTION

Deep learning (DL) models have achieved state-of-the-art performance in various tasks and domains, including drug discovery (LI et al., 2021), speech recognition (PARK et al., 2020), object recognition (DOSOVITSKIY et al., 2021), question and answer (ZHU et al., 2021), machine translation (TAKASE; KIYONO, 2021), image description (PAN et al., 2020), natural language understanding (KHURANA et al., 2022), and image understanding (ZHAI et al., 2022). Some studies even report 'superhuman' performance, suggesting its performance surpasses that of human experts (FUCHS et al., 2021). Notably, recent advancements in Visual Language Model (VLM) such as CLIP (RADFORD et al., 2021) and ALIGN (JIA et al., 2021a) have enabled applications to achieve remarkable zero-shot image classification performances. This is achieved by simply querying a pre-trained model without requiring any additional model training or fine-tuning.

Such claims have created a self-reinforcing cycle of increasing popularity, leading to the adoption of deep learning models to ever more areas of research and applications (GOOGLE..., ; AMAZON..., ; POWERED...,). However, the endorsement of these models in high-stakes domain applications (e.g., healthcare and legal systems) has been limited due to their lack of interpretability and their bias towards spurious signals (GEIRHOS et al., 2020).

Investigations to understand the decisions of DL models have uncovered several situations in which the models can fail. For instance, Szegedy et al. (2013) found counter-intuitive properties of DL models, demonstrating that adding minimal noise to the model input can lead the model to change its decision to an incorrect prediction. This fragility is known as an Adversarial Attack. While it exposes potential issues with the robustness of DL models, existing literature argues that the data used in these attacks is artificially generated and falls in the out-of-distribution data concept. Despite these counterarguments, several works demonstrate that DL models can fail drastically, even when dealing with natural images (HENDRYCKS et al., 2021). Unfortunately, adversarial attacks are not the only reason for model failure. We can find DL models making wrong decisions due to background information or structure similarity (NGUYEN; YOSINSKI; CLUNE, 2015; ALCORN et al., 2019; BEERY; HORN; PERONA, 2018). Figure 1 illustrates some of these cases, suggesting the models make decisions based on incorrect information, such as background information (XIAO et al., 2021), shortcut learning (GEIRHOS et al., 2020), or spurious correlations between contextual features and the input label (EISENSTEIN, 2022).

In the face of these shortcomings and limitations of DL models, there is a need to explore new architectures and methods that can mitigate these issues. One such approach is the zero-shot image classification model based on VLMs, which classifies an image based on its description, thus being naturally interpretable. The standard pipeline of this zero-shot approach is to compute the similarity score between the query image embedding and various category name embeddings to determine how close the image's content is to each category

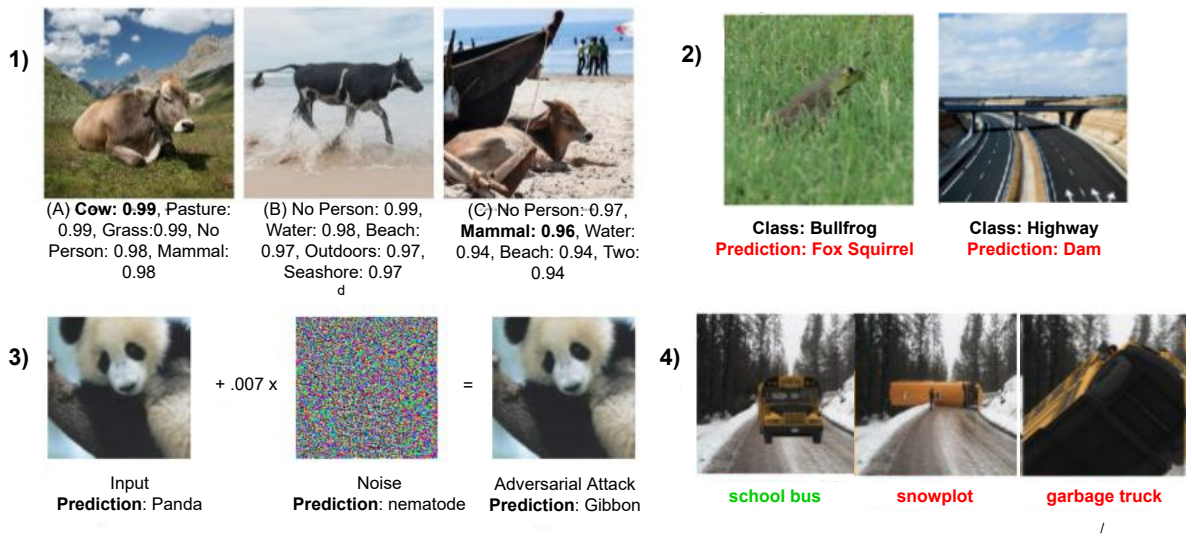

Figure 1 – **Example of image classification failures.** Debugging DL models is important to diagnose failures and help understand model decisions. Several works explore model decisions with different types of input information. For example, situation 1) shows that the model fails to classify a *cow* when it is present on a background different than usual. In situation 2), a bullfrog is misclassified as a fox squirrel and a highway as a dam. Unlike the first two examples, situation 3) presents an example of an adversarial attack, demonstrating that after adding noise to the input, the model fails drastically, even though it made the correct decision on the original image. Situation 4) shows that the model fails to classify an image correctly while maintaining the same background but changing the object position. The main figures of the plots were obtained from (NGUYEN; YOSINSKI; CLUNE, 2015; ALCORN et al., 2019; HENDRYCKS et al., 2021).

name. The category with the highest similarity score to the query image is chosen as the classification result. In this direction, a new approach has been proposed to combine Large Language Model (LLM) with VLMs to perform zero-shot image classification (MENON; VONDRICK, 2023), specifically using the ChatGPT (OPENAI, 2023) with OpenAI's Contrastive Language-Image Pretraining (CLIP) (RADFORD et al., 2021). Instead of using the straight category name embedding, the ChatGPT+CLIP method computes the similarity score between the category descriptions extracted from ChatGPT and the image embedding using CLIP. This zero-shot approach achieves approximately 75% accuracy on the ImageNet dataset (DENG et al., 2009a; MENON; VONDRICK, 2023). Despite this remarkable success, it is crucial to understand its limitations as we know for other deep learning models (Figure 1). For instance, as these VLMS are trained using hundreds of millions of (image, text) pairs, are they biased to background information such as standard models?

All of these discussion highlights that the still open problem is how to train robust models that make inferences based on the correct information, such as signal features or information relevant to the problem being solved. Several works have been proposing new loss

functions to guide the model to focus on signal features (ROSS; HUGHES; DOSHI-VELEZ, 2017; SCHRAMOWSKI et al., 2020; VIVIANO et al., 2021; SIMPSON et al., 2019; RIEGER et al., 2020; ERION et al., 2019), thereby using signal information instead of contextual information in the inference process. These methods are referred in the literature as Right for the Right Reasons (RRR). These loss functions generally use second-order gradient optimization (DRUCKER; CUN, 1992) and incorporate a right reasons factor into the loss function. The right reasons factor encourages the model to use the signal information in decision-making.

Though RRR has shown promising results, these methods have not been evaluated on a large-scale benchmark and generally use interpretability methods in the loss function, making them dependent on interpretability accuracy (CARVALHO; PEREIRA; CARDOSO, 2019; MOHSENI, 2019; TORRES et al., 2023), dependent to specific interpretability methods implementation (RIEGER et al., 2020; ERION et al., 2019), and have an additional computational cost as we need to compute second-order derivatives during training.

These critical concerns make it evident that the reliance on interpretability methods, as observed in RRR methods, introduces notable challenges. Indeed, there is not even agreement on the definition of interpretability and explainability (FLORA et al., 2022), nor an objective definition on what does mean feature importance, feature attribution, feature reliance, and so on. This lack of formalism implies that if an interpretability method attributes high importance to an input feature, we can not make any assumptions on the model behavior (BILODEAU et al., 2022). Thus, these issues make it evident that we must define interpretability methods as an end-task, objectively expressing what we want to interpret.

Based on these observations and discussions, we may raise some questions: Can we ensure that the model's decisions align with human knowledge in a more agnostic manner (without needing specific interpretability methods? How can we define the interpretability method as an end task? How can we ensure that a model is using some specific feature information in the inference decision-making?

1.1 OBJECTIVES

This thesis focuses on deep learning-based image recognition models. Our main objective is to improve these models in relation to robustness and interpretability. Our main hypothesis asserts that training these models with appropriate data and appropriate loss functions may improve their robustness and make them base their decisions on signal features rather than background or non-informative features. In addition, we also hypothesize that with well-defined and objective pipelines, we can extract better insights about the model's interpretability.

Motivated by the recent development of interpretability methods and adversarial training, we explore how to use these techniques to generate new data and enhance the model training process to align model decisions with the human decision. Besides, we explore how to use interpretability maps to structure model decision analysis and how to define interpretability objectively as an end task.

We formulated the following research questions to guide our investigation: How can interpretability maps be leveraged to generate curated training data? How can human knowledge contribute to the generation of new training samples? Do adversarial samples impact the right for the right reasons methods? How does this new data impact model decision-making? How can we evaluate which input image information mostly impacts the model decisions? Finally, can we define the interpretability method as an optimization problem?

Instead of relying on a single method or analysis to answer these research questions, we employ a holistic methodology to incorporate various methods and components to provide a more comprehensive and nuanced understanding of these points. The methods are present from chapter 3 to 7, where each chapter is self-contained so the reader can read each independently. We group the methods into two categories: (1) data methods that generate new data samples to guide the model alignment with human decision-making and (2) methods to evaluate which input region impacts the model decisions. Figure 2 summarizes the data methods, while Figure 3 summarizes the methods to analyze the input feature impact on model prediction.

We applied all the proposed methods in several generic and specific scenarios. All contributions aim to drive deep learning approaches in image recognition toward reliable, interpretable, robust, and trustworthy models. In the following, we present an outline of the proposed contributions distributed in different chapters.

Chapter 2, State-of-the-art: This thesis is centered around key aspects of deep learning, including Interpretability, Adversarial robustness, and Right for the right reasons methods. In this chapter, we provide an overview of the state-of-the-art for each topic.

Chapter 3, Active image data augmentation, published in (SANTOS et al., 2019) and (SANTOS et al., 2022): In this chapter, we propose the Active image data augmentation (ADA) (SANTOS et al., 2019) method which uses the interpretability maps to generate new training data by removing the non-signal information the model attribute high importance. Using this approach, we argue that the model will learn to produce the right answer without using these non-signal features. As there are several Interpretability methods, we perform an extensive experiment in (SANTOS et al., 2022) to evaluate their impact on ADA.

Chapter 4, Right reasons data augmentation, published in (SANTOS; ZANCHETTIN, 2023)[1]: Despite the widely used interpretability methods, there are still concerns regarding its relation with model causality, so we can not guarantee that the input feature with high attribution values drives the model prediction. Thus, we proposed the Right reasons data augmentation (RRDA), which explores the input signal masks as a priori information and generate new input samples to force the model to make inference based on the signal features instead of background features.

Chapter 5, Adversarial right for the right reasons, published in (SANTOS; SOUZA; ZANCHETTIN, 2023b): Several works have shown that the interpretability maps of models trained with adversarial samples highlight the signal information or the objects edges more than

[1] Best paper of the LXAI Workshop at ICCV 2023.

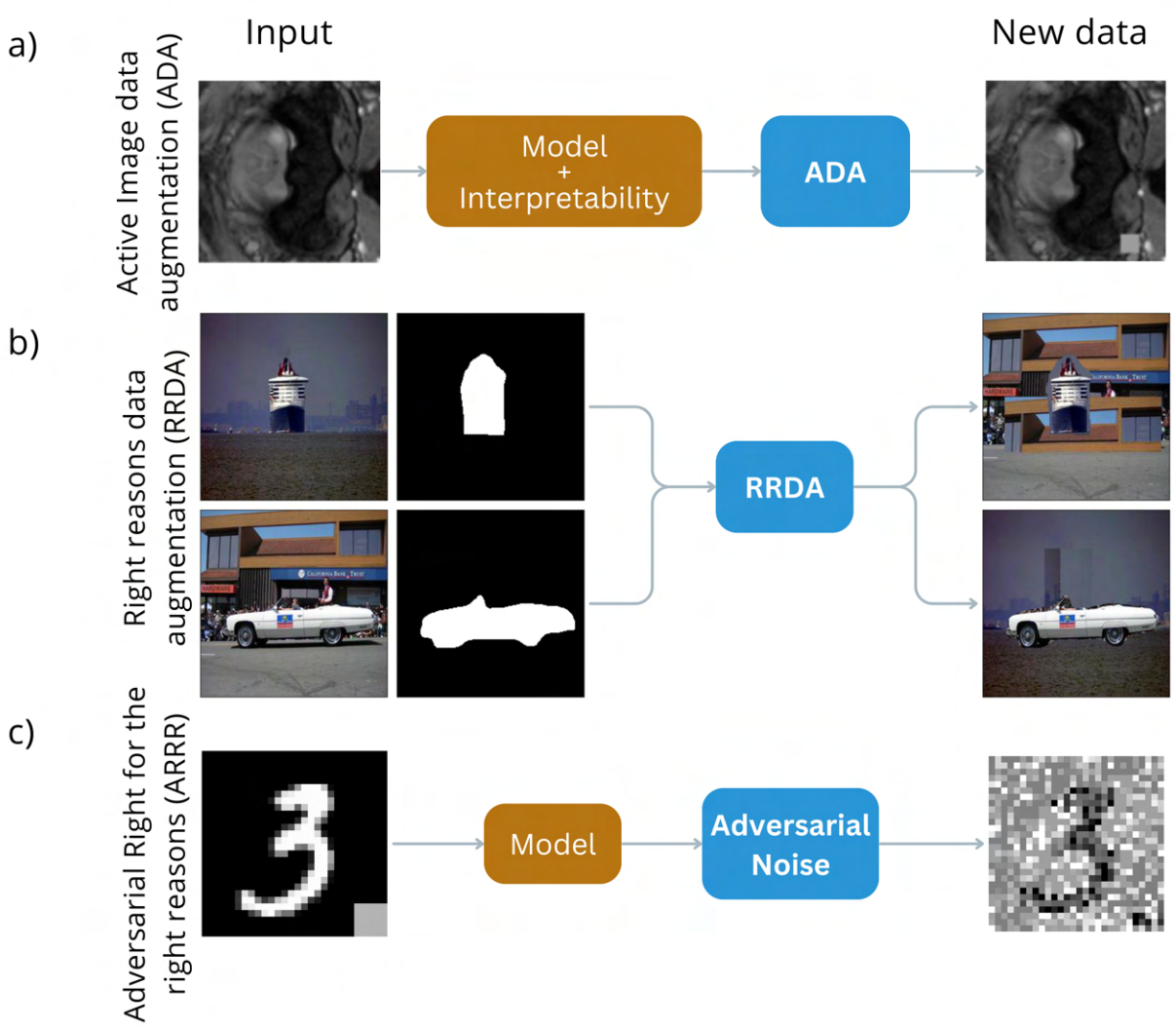

Figure 2 – **Example of the data generation methods.** The three proposed data generation contributions to align the model with human decision-making. a) The active image data augmentation method uses interpretability maps to remove the non-signal region, which is most important to the model but not for the human decision-making process. b) The right reasons data augmentation creates random backgrounds to augment training samples. b) The adversarial right for the right reasons uses adversarial samples during model training.

models trained with traditional approaches. We proposed the Adversarial Right for the Right Reasons (ARRR) method to explore whether adding adversarial samples into RRR training methods will improve the model's robustness and interpretability.

Chapter 6, Background dependence of Visual language models, published in (SANTOS; SOUZA; ZANCHETTIN, 2023a): Contrastive visual language models, such as CLIP and ALIGN, are trained to map input images and input text into the same feature space, minimizing the distance between their embeddings. Typically trained with hundreds of millions of (image, text) pairs, these models enable high-accuracy zero-shot image classification after

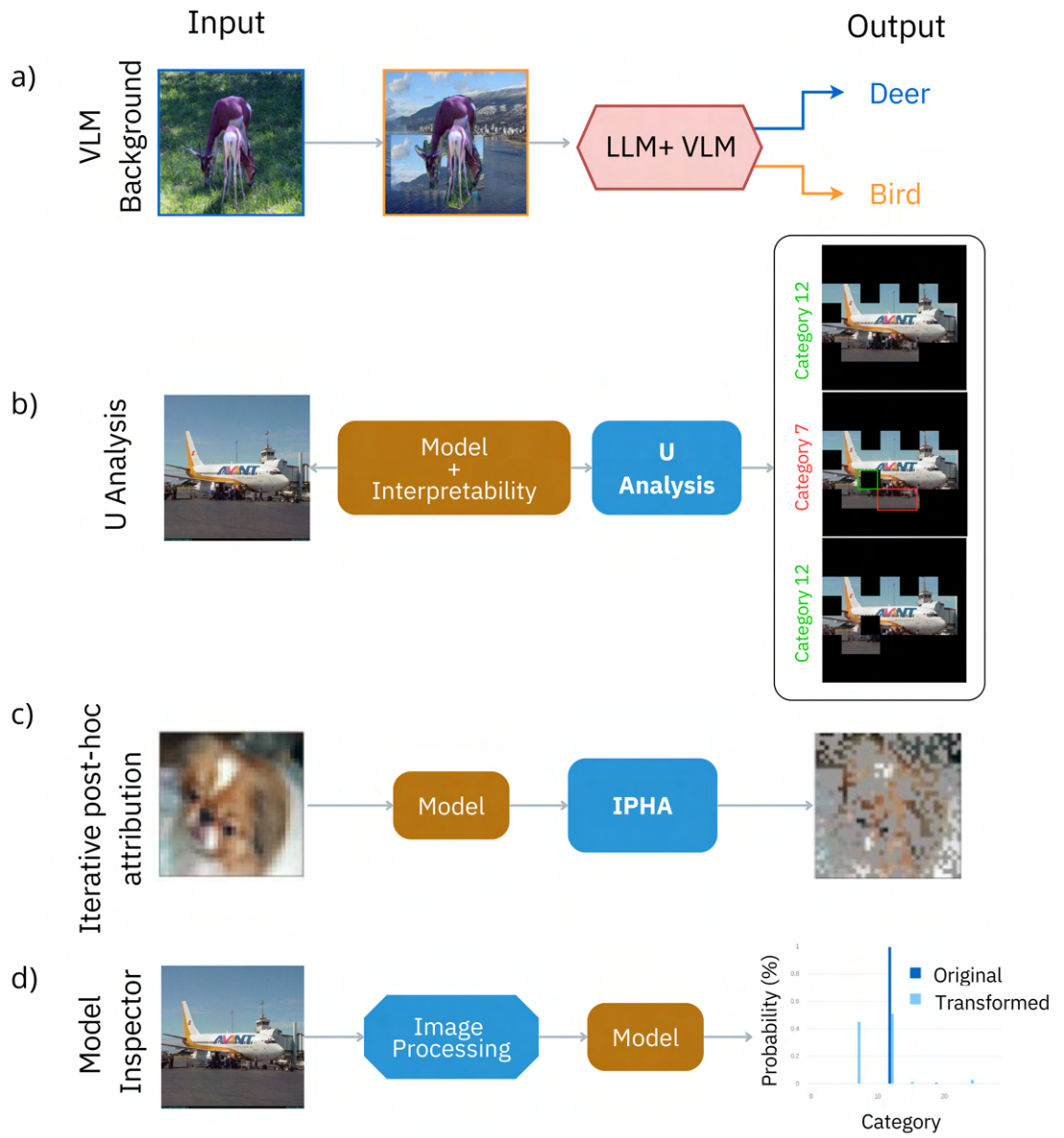

Figure 3 – **Example of the input feature analysis methods.** a) The VLM's background sensitivity analysis computes the difference between its prediction on the original image and the same image with a different background. b) The U Analysis uses interpretability maps to find input feature co-dependence to model prediction; for example, the red and green box is necessary to appear in the input image so the model predicts correctly. c) Iterative post-hoc attribution methods employ an optimization view of interpretability and find a binary mask that points to the important input features for model prediction. d) The Model Inspector tool allows the user to manipulate the input image and verify how it affects the model prediction.

training (MENON; VONDRICK, 2023) and are a new baseline to image models. Recognizing their significance, this chapter conducts an evaluative analysis of zero-shot image classifiers based on CLIP and ALIGN models. It examines their sensitivity to background information, which is

a flaw in traditional approaches.

Chapter 7, Model inspector tool, published in (SANTOS et al., 2023) and (SANTOS et al., 2021): This chapter proposes the Model inspector tool, along with two methods to analyze image classification models, namely U Analysis (UA) and Iterative post-hoc attribution (IPHA) method. The Model Inspector allows users to load an image classification model and input images to evaluate the model performance. Then, the user may perform several image transformations to evaluate how the model deals with those transformations. The U Analysis represents a pipeline of image transformations based on the interpretability maps and highlights feature co-dependence on the model prediction. Finally, the IPHA proposes to map the model interpretability problem as an optimization problem; it can highlight which features are important to model prediction and the non-important ones.

2 STATE-OF-THE-ART

This thesis is closely related to different deep learning topics, such as interpretability, adversarial robustness, and right for the right reasons methods. Thus, aiming to contextualize the reader, we review each topic in this chapter, highlighting the most relevant methods to this work.

2.1 INTERPRETABILITY METHODS

Deep learning (DL) models have achieved state-of-the-art results in several applications across different domains, such as natural language processing (OTTER; MEDINA; KALITA, 2021) and computer vision (DONG; WANG; ABBAS, 2021). This success is not limited to the scientific field; today, DL models have become pervasive through numerous applications[1]. Despite this success, they face challenges in being used in production in certain areas, such as medical systems (MIOTTO et al., 2017) and law systems (DEEKS, 2019). Many of these systems require explaining the decision made by the model, or the user needs at least an indication of why the model suggested a particular inference for an input. The difficulty of using those models in these domains is not unfounded; they are known to be black-box models due to generally working with high-dimensional data and being composed of a high number of layers and nonlinear processing, making it not easy to track their decision process.

In an effort to mitigate the black-box issue of DL models, the scientific community has proposed interpretability methods for explaining the DL models decision. According to Doshi Velez and Been Kim (2017) (DOSHI-VELEZ; KIM, 2017), the authors define interpretability as "the ability to explain or to present in understandable terms to a human" (p.2). By enhancing the interpretability of deep learning models, we not only address concerns surrounding their black-box decision-making processes but also their applicability in critical domains such as healthcare, finance, and autonomous systems. This quest for interpretability reflects a broader commitment to building trust and facilitating the integration of these advanced models into real-world scenarios. Now, let's explore a practical example of the notation, vectors dimensions, input and output vectors when we compute the interpretability of deep learning models.

In practice, let $F : \mathbb{R}^n \rightarrow C$ represent a deep learning model, where $\mathbb{R}^n$ denotes the n-dimensional real space, and C is the set of categories. Consider an input vector x in $\mathbb{R}^n$ associated with the input data. The interpretability of $F^c(x)$ yields a vector a in $\mathbb{R}^n$, where each i-th position in a indicates the importance of the i-th position in x to the output $F^c(x)$. The a vector is also referred to as an attribution map, interpretability map, or the interpretability of the model output $F^c(x)$.

One of the pioneering interpretability methods was proposed by Strumbelj e Kononenko (2010). In this work, the authors proposed an approach to obtain model interpretability by

[1] https://www.technologyreview.com/2015/02/09/169415/deep-learning-squeezed-onto-a-phone/

calculating Shapley Values. The method randomly permutes the features and gradually adds each feature to a chosen reference point. The difference between the outputs obtained after adding each feature corresponds to its importance value. This process is repeated N times, and the final importance of each feature is obtained by averaging all N executions.

In addition, other interpretability methods have emerged over the years (ZHANG et al., 2021). Most methods found in the literature share some common characteristics; for example, they may be based on the gradient of the model's prediction with respect to the input vector, attempt to create a local linear version of the model, and may also construct the model interpretability based on a baseline version of the input data vector. In the following, we present a review of these methods grouped into these categories.

Gradient-Based Models

According to our literature review, *Deconvolution* (ZEILER; FERGUS, 2014) is the pioneering work in obtaining interpretability of Convolutional Neural Network (CNN) models following a top-down approach. The method can be seen as a traditional CNN model as it uses the same components (filters, pooling), but in reverse order. Instead of mapping the input image into the filter space, it maps the filter information into the input image space. Initially, an input image is given to the trained CNN model, and the feature maps of a layer L are obtained. To analyze an activation i of layer L, all activations different from i are set to zero, and the feature map is passed to a DeConvNet. Then, the DeConvNet performs the following operations successively: (i) UnPool, (ii) ReLU, (iii) Transpose Filter. These three operations are repeated until a representation in the pixel space is obtained. In the *Transpose Filter* layer, the same filters as the original CNN are used, but they are transposed to return the feature maps to the dimension of the input image. Since the pooling layer cannot be transposed, DeConvNet defines a hook structure to save the positions of the poolings in the original CNN.

Deconvolution (ZEILER; FERGUS, 2014) is almost equivalent to calculating the gradient of the output of an arbitrary neuron with respect to the input vector (*Vanilla Gradient*). The minor difference is that when the signal is backpropagated through a ReLU function, it changes to zero for each negative value of the previous gradient. Following a more formal approach, the Vanilla Gradient method (SIMON; RODNER; DENZLER, 2014; SIMONYAN; VEDALDI; ZISSERMAN, 2014) was proposed. It obtains an interpretability map containing the degree of importance of each position in the input vector x. Given an output score i from a model f, it calculates the absolute value of the gradient ∇f_x^i to obtain the importance of each position of x for i output of model f.

As discussed earlier, *Vanilla Gradient* (SIMON; RODNER; DENZLER, 2014) does something similar to *Deconvolution*. However, instead of changing the negative values of previous gradients to zero, it changes these values to zero when the values of the respective layer in the forward step are negative (i.e., it follows the derivative of the ReLU function). The *Guided Backpropagation* method (SPRINGENBERG et al., 2015) combines both methods, changing to

zero all values where the gradient values are zero or where the forward step of the respective layer is negative.

The Gradient-Weighted Class Activation Mapping (Grad-Cam) (SELVARAJU et al., 2017) uses gradient information to produce activation maps that highlight the most input's important parts to the model inference. According to the authors, the most significant difference between Grad-Cam and the presented methods, such as Deconvolution and Guided Backpropagation, is that Grad-Cam can highlight the most discriminative information for the predicted class. In contrast, the other methods obtain high-resolution maps highlighting many image details. The Grad-Cam steps are present in equations 2.1-2.2. In general, Grad-Cam computes the gradient of the class c output score y^c with respect to the feature map A ($\frac{\partial y^c}{\partial A}$), then weights ($\alpha_k^c$) the importance of each feature map A_k, and calculates the combination of their respective gradients using all α_k^c as weights. Next, as in equation 2.2, it applies the ReLU function (NAIR; HINTON, 2010) to the result of the linear combination to obtain only the positive contributions. It is important to highlight that the feature map A can be the output of any layer from the deep learning model, but generally the last convolutional layer output is used. Despite Grad-Cam finding regions considered discriminatory, it fails to find important details. Therefore, the authors introduced the Guided Grad-Cam method in the same work, which consists of the multiplication between the outputs of Guided Backpropagation and Grad-Cam.

$$\alpha_k^c = \frac{1}{Z} \sum_i \sum_j \frac{\partial y^c}{\partial A_{ij}^k} \tag{2.1}$$

$$\text{Grad-CAM}^c = \text{ReLU}\left(\sum_k \alpha_k^c A^k\right) \tag{2.2}$$

While presenting interesting results, gradient-based interpretability methods such as Vanilla gradient, GradCam, Deconvolution, and Guided Backpropagation generate interpretability maps with a lot of noise. Therefore, Smilkov et al. (2017) proposed the Smooth Gradient method to obtain interpretability maps with less noise. The method involves adding noise to the original image and obtaining its interpretability for N times. The final interpretability map is obtained from the average of the N interpretability maps.

In addition to producing interpretability maps with noise, gradient-based methods have other limitations. For example, they can not propagate the importance signal in regions where the gradient is zero, and they may also have problems in regions of discontinuities. To mitigate these two issues, Shrikumar, Greenside e Kundaje (2017) proposed the *Deep Lift* method, which uses a reference point to obtain the interpretability map. The reference point must be considered neutral and chosen appropriately for each problem. To obtain the importance of the features, Deep Lift calculates the difference in the activation of the neurons between when the model processes the original input and when it processes the reference input, assigning importance to each feature based on this difference.

Different from the approaches presented so far, Sundararajan, Taly e Yan (2017) developed the interpretability method Integrated Gradients (IG) based on an axiomatic approach. The authors proposed two axioms that interpretability methods must obey. The axioms are *Sensitivity* and *Implementation invariance*. The *Sensitivity* axiom suggests that if two images differ only in one pixel and produce different inferences, then the interpretability method must assign a non-zero importance to this pixel. The *Implementation invariance* axiom states that the interpretability method must be robust to implementation, i.e., different implementations of the same method should return the same interpretability. Given the input vector x, the baseline vector $x^{'}$, and the class c of interest, the IG suggests the interpretability by accumulating the gradients of all points on the straight line between the baseline vector and the vector of interest, as shown in Equation 2.3.

$$IntegratedGrads_i ::= (x_i - x_i^{'}) \times \int_{\alpha=0}^{1} \frac{\partial F^c(x^{'} + \alpha \times (x - x^{'}))}{\partial x_i} d\alpha \tag{2.3}$$

The version of IG presented in Equation 2.3 is computationally expensive, as there are infinite values from $\alpha = 0$ until $\alpha = 1$. However, in practice, an approximate version of IG is calculated as shown in Equation 2.4. In this version, m is the number of points between the baseline vector and the vector of interest, which also represents the number of steps in the Riemann integral approximation.

$$IntegratedGrads_i^{Aprox.} ::= (x_i - x_i^{'}) \times \sum_{k=1}^{m} \frac{\partial F^c(x^{'} + \frac{k}{m} \times (x - x^{'}))}{\partial x_i} \times \frac{1}{m} \tag{2.4}$$

Although the Integrated Gradient method presents interpretability maps with little noise when compared to other purely gradient-based methods, it has two negative points: (i) the choice of the baseline (STURMFELS; LUNDBERG; LEE, 2020), and (ii) the choice of the number of points on the line that it will calculate the gradient to be accumulated (m in Equation 2.4). To mitigate these negative points of the IG method, Xu, Venugopalan e Sundararajan (2020) proposed the Blur Integrated Gradient (BIG) method. BIG eliminates the choice of a baseline vector by using several vectors similar to the original version but with noise. Thus, instead of calculating the gradient with respect to all points on the straight line between the baseline and the original vector, BIG accumulates the gradients between the original vector and those with noise.

Inheriting ideas from the various approaches discussed so far, Sattarzadeh et al. (2020) proposed the Semantic Input Sampling for Explanation (SISE) method. SISE consists of 4 steps: (1) *Feature map extraction*, (2) *Feature map selection*, (3) *Attribution mask*, and (4) *Visualization map*. Figure 2.1 illustrates the four steps of the SISE method. In the first step, it is necessary to choose the layers of interest, and then an input vector feeds the model to obtain the feature maps produced by the layers. In the second phase, a subset of feature maps is selected from all obtained features. Next, the selected feature maps are post-processed in the

third step to produce an *Attribution mask*. Finally, in the fourth step, all produced *Attribution masks* are combined, and a single interpretability map of the model is produced.

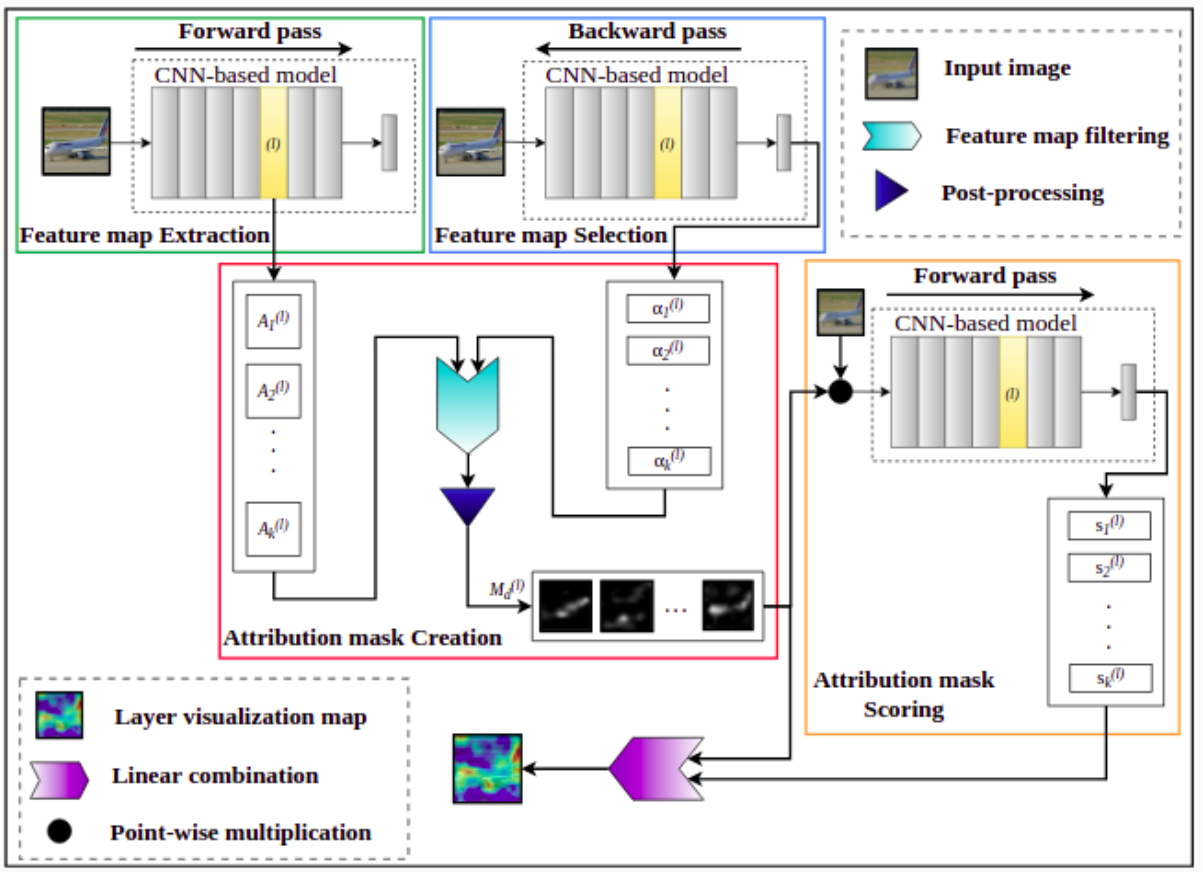

Figure 4 – Graphic representation of the steps of the SISE method. The *Forward pass* step feeds the model with the input image and performs the inference, while in the *Backward pass*, a quantity of *feature maps* from the intermediate layers is chosen. During the last two steps, *Attribution mask* and *Visualization mask*, attribution maps are created for each *feature map*, and then all these maps are grouped to form the final interpretability of the model. Source: (SATTARZADEH et al., 2020)

Adaptive Semantic Input Sampling for Efficient Explanation of Convolutional Neural Networks (Ada-SISE) (SUDHAKAR et al., 2021) is an extension of the SISE method. The authors of Ada-SISE noticed that many feature maps with redundant information were chosen in the second step of the SISE method, making it computationally expensive. To mitigate this issue, the Ada-SISE method selects the feature maps adaptively and has a lower computational cost. The results obtained showed that the feature maps obtained by Ada-SISE were not qualitatively compromised when compared to SISE and even, in some cases, showed the information more clearly.

Linearization/Decomposition Methods

The interpretability methods discussed earlier are based on the gradient of the neural network. Following an approach that does not require directly calculating the gradient, Bach et al. (2015) proposed the Layer-wise relevance propagation (LRP) method. LRP is a conservative method whose purpose is to understand the decision of a deep network by decomposing the output $f(x)$ into the input vector x, thus obtaining the degree of relevance of each dimension of x. The method is conservative because the sum of the relevance of each pixel is equal to the output value of the mode that we want to interpret. After feeding the model with the input vector x and obtaining the output $f(x)$, LRP defines a set of rules (e.g., the Basic Rule, Epsilon Rule, and Gamma Rule) for backpropagate the value $f(x)$ to the input vector x and thus obtain the relevance of each x feature. Figure 2.1 illustrates the execution of the LRP method.

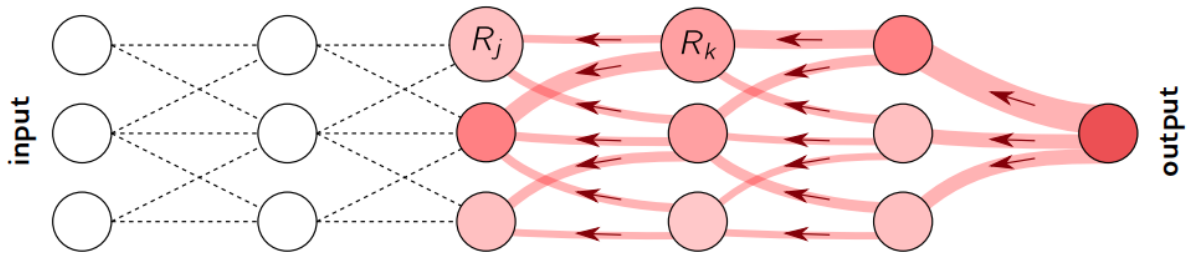

Figure 5 – Representation of the *Layer-wise relevance propagation* method. The arrows in the red-shaded region, starting from the output, represent the execution of backpropagation of the importance of each processing unit. Source: (MONTAVON et al., 2019)

Deep Taylor Decomposition (DTD) (MONTAVON et al., 2017) is an evolution of the LRP method. Its main proposal is to decompose the activation of a neuron through the contribution of each input pixel. It calculates this relevance using a first-order Taylor series expansion through a reference point x_0 where $w^T x_0 = 0$ (for the case of linear regression). The relevance of the chosen initial neuron is equal to the sign obtained in the *forward* phase.

Taking a different approach from the approaches presented earlier, LIME (RIBEIRO; SINGH; GUESTRIN, 2016) obtains the interpretability of *any* model by linearizing the classifier decisions at the point of interest. To build this linearization, LIME creates several points in the vicinity of the point of interest and trains a simple (linear) model using these new points. Kernel Shap (LUNDBERG; LEE, 2017) is an extension of the LIME method, whose goal is to obtain the *Shapley Values* through formulating the LIME method. It achieves this goal by changing parameters such as error function, kernel, and regularization terms chosen heuristically.

All methods discussed so far assume the input vector x as important and try to create a contribution map to obtain the output $f(x)$. Taking a different perspective, PatternNet (PN) and PatternAttribution (PA) (DUMITRU et al., 2018) assume that an input vector is

composed as follows: $x = s + d$; $input = signal + distractor$. Where s is the signal part containing the relevant information for the model's inference, while d is a distractor composed of obscuring information that makes the task more challenging. In addition, the authors argue that visualizing the output of the function as a whole is simple, as seen in the methods presented so far. Thus, (i) visualizing the signal and (ii) the relevance of each signal feature is more complicated. So, they proposed the PatternNet (PN) method to find the signal s and PatternAttribution (PA) to find the relevance of each region of s.

2.2 ADVERSARIAL ATTACKS AND TRAINING

Adversarial training boosts the robustness of deep learning models by intentionally exposing them to adversarial examples during training. This process not only teaches the model to identify and correct vulnerabilities but also promotes the learning of more efficient and robust features. As a result, models become less sensitive to minor variations and more capable of generalizing across the training data set. As our main goal is not to propose a new adversarial attack method but to use its adversarial noise, we will focus on the most standard adversarial approaches in this section.

As far as we know, Fast Gradient Sign Method (FGSM) is the first adversarial attack method. Goodfellow, Shlens e Szegedy (2015) proposed the FGSM to generate adversarial attacks, where it involves calculating the gradient of the error function with respect to the input vector and obtaining the signs (direction) of each dimension of the gradient vector. The authors argue that the gradient direction is more important than the specific point of the gradient because the space in which the input vector is contained is not composed of sub-regions of adversarial attacks. Other variations of FGSM are also present in the literature, such as R-FGSM (TRAMÈR et al., 2018) and Step-LL (KURAKIN; GOODFELLOW; BENGIO, 2017).

An alternative to make the model robust to adversarial attacks is training it with adversarial samples. In Equation 2.5, we present a cost function for adversarial training based on FGSM. Given a standard error function (J) and the input vector x, it obtains the final error based on the sum of two steps: (1) Computes the error based on the original input vector ($J(\theta, x, y)$), and (2) Computes the error based on the FGSM adversarial attack ($J(\theta, x + \epsilon\text{sign}(\nabla_x J(\theta, x, y)))$).

$$J'(\theta, x, y) = \alpha J(\theta, x, y) + (1 - \alpha) J(\theta, x + \epsilon\text{sign}(\nabla_x J(\theta, x, y))) \tag{2.5}$$

As FGSM only generates the adversarial noise based on gradient directions. Madry et al. (2018) conducted a study on adversarial attacks from a min-max perspective, to be precise about the class of attack they try to recognize and consequently defend against. Equation 2.6 presents this min-max formulation. The max part of this formulation aims to find an adversarial noise that produces the highest value of the error function L when added to the input vector. The min term aims to find the model parameters that minimize the error function L, making the model robust to the max-attack. From this analysis, the authors proposed the

Projected Gradient Descent (PGD) method, which they call a first-order universal attack, i.e., the most difficult attack using only first-order information.

$$\min_{\theta} \; p(\theta), \;\; \text{where } p(\theta) = \mathop{\mathbb{E}}_{(x,y)\in D} [\max_{\delta \in S} \; L(\theta, x+\delta, y)] \tag{2.6}$$

Despite the PGD method achieving good results, it is computationally expensive because it needs to calculate the gradient of the function several times. To mitigate this restriction, Shafahi et al. (2019) proposed the Free Adversarial Training (FAT) method. The main contribution of the FAT method is to reuse the gradient of the function computed at the moment when the model performs the gradient descent in the optimization step. However, with only one gradient calculation step, they cannot construct an attack that causes as much error as in PGD. To minimize this drawback, the authors propose training the same input batch for m times, so the model will be robust to more than one version of an attack for the same input vector.

2.3 INTEGRATING PRIOR KNOWLEDGE INTO MODEL INTERPRETABILITY

DL models can identify patterns even when we shuffle the data classes (ZHANG et al., 2017). They are characterized by a high number of layers (HE et al., 2015) and the use of nonlinear functions, making it difficult to interpret their decisions. In section 2.1, we discussed a set of methods proposed to interpret their decisions. This interpretability can indicate features that may be considered important or unimportant for model prediction. After obtaining it, we can assess whether the model uses the same features humans use for decision-making in the same problem. From this analysis, we may raise a question: what to do when the model is making the correct decision for reasons the domain expert considers incorrect? Or, how do we ensure that the model uses the features considered correct by the domain expert for decision-making? The interpretability methods presented earlier only return the degree of importance of each feature; thus, they do not solve this problem and only help to find it. In addressing this issue, a class of methods has been proposed to make the model able to infer correctly for the right reasons (used by the domain expert for decision-making). This section aims to present and discuss some of these methods.

Understanding the importance of the gradient vector of the model's output with respect to the input vector ($\frac{\partial f_i}{\partial x}$) is crucial. This gradient provides the direction and magnitude needed to adjust each input vector dimension to maximize the model's output. The gradient vector has been used as one form of interpretability, where dimensions with a high absolute value are considered more important for the model during the inference process. From this analysis of gradient usage, Ross, Hughes e Doshi-Velez (2017) proposed the Right for the right reasons (RRR) method by introducing the *Vanilla Gradient* during training and regularizing the model to use only the features that are genuinely important for the problem in the domain expert's view. This regularization is performed by penalizing the gradients of features con-

sidered unimportant by the domain expert, forcing them to have a zero value. Equation 2.7 shows the loss function proposed by RRR authors, it has two main components, namely Right answer and Right for the reasons. The right answer is only the cross entropy loss function to force the model to answer correctly. In contrast, the Right for the right reasons performs a point-wise multiplication between the domain expert binary mask (i.e., a variable with 1 in the non-important features and 0 on the signal) by the gradient of the model output with relation to the input, then sums its values. Thus, the Right for the right reasons factor computes how much the model uses non-important features.

$$L(\theta, X, y, A) = \underbrace{\sum_{n=1}^{N}\sum_{k=1}^{K} -y_{nk}\log(y'_{nk})}_{Right\ Answer} + \lambda_1 \underbrace{\sum_{n=1}^{N}\sum_{d=1}^{D}(A_{nd}\frac{\partial}{\partial x_{nd}}\sum_{k=1}^{K}\log(y'_{nk}))^2}_{Right\ for\ the\ right\ reasons} + \lambda_2 \underbrace{\sum_{i}\theta_i^2}_{Regular} \quad (2.7)$$

Also, using interpretability maps, Liu e Avci (2019) propose a way to introduce prior information into models, forcing them to use specific features of the input vector. Their method involves introducing the L2 error between the output of an interpretability method and the desired importance value (prior information). The experiments conducted by the authors showed that after applying this method, the models did not experience a performance loss according to traditional metrics, and the model became fairer considering the domain expert's opinion.

Continuing in this direction of making the model consistent with a domain expert, Erion et al. (2019) proposes a new and flexible interpretability method called Expected Gradient (EG). Also, like Liu e Avci (2019), they incorporated prior information into the model using EG as an interpretability method. From the experiments conducted, they show that to find the features that the model actually uses for inference, the EG method has more reliable results than its predecessor *Integrated Gradients*.

Another approach in this context, Du et al. (2019) proposed the CREX method, which forces the model's interpretability to be equal to an interpretability obtained by a domain expert. Additionally, it also enforces that the model's interpretability is sparse. This second point is essential when the model does not have domain expert annotations during training or has them partially.

Rieger et al. (2020) proposed the Contextual Decomposition Explanation Penalization (CDEP) method, aiming to enable the addition of domain knowledge to the models so that they can penalize unimportant features and only use those that are genuinely important. To achieve this goal, CDEP uses the Contextual Decomposition (CD) method (MURDOCH; LIU; YU, 2018) as a base. CD decomposes the input vector into regions of interest and obtains an output score from the network for each of these regions. Thus, CDEP uses the scores obtained by CD and forces the score of the region with unimportant features to be zero, making the region of features considered important by the domain expert responsible for the model's inference. Equation 2.8 represents the modeling of the authors of CDEP, where the function

L_{expl} is responsible for obtaining the model's interpretability and forcing it to be equal to an interpretability $expl_X$.

$$\hat{\theta} = \underset{\theta}{argmin} \underbrace{L(f_\theta(X), y)}_{Prediction\ error} + \lambda \underbrace{L_{expl}(expl_\theta(X), expl_X)}_{Interpretability\ error} \tag{2.8}$$

Unlike the methods previously discussed, Schramowski et al. (2020) proposed the Explanatory Interactive Learning (XIL) method to involve domain experts in the model training process. The role of the expert in XIL is to provide feedback on the model's interpretability so that the model adjusts to use the features considered relevant by the expert. In addition to the interactive use of domain experts, XIL also uses the RRR method during model training.

Following a different approach from using interpretability methods, Bao et al. (2018) proposes introducing human knowledge a priori into attention models, allowing the model to produce attention weights of better quality and, consequently, achieve better results in the given problem.

2.4 CONCLUSION

In this chapter, the state-of-the-art methods related to the proposed thesis were presented. We began the discussion with a general overview of interpretability methods in the literature. Next, we introduced adversarial training methods and concluded by discussing training approaches that make the model consistent with the opinion of a domain expert.

After describing all these works, a noteworthy point for the reader is the importance of the gradient vector of the model's output ($f(x)_c$) with relation to the input vector (x). This gradient vector is used in different methods across all three topics discussed. For instance, interpretability methods utilize the information from this vector or a variation of it to highlight the important input features for the model inference. Adversarial training methods use it to determine how feature values should be altered to drive inference errors. Finally, RRR methods use it to identify if models are using relevant features for the inference.

3 ACTIVE IMAGE DATA AUGMENTATION

One of the questions discussed in this thesis is whether interpretability methods can help guide the model to use important features for the problem, thus making the model's decisions consistent with a knowledge specialist. The RRR methods presented in the theoretical background chapter, such as *Right for the right reasons* (ROSS; HUGHES; DOSHI-VELEZ, 2017), *Expected Gradient* (ERION et al., 2019), and *Contextual Decomposition Explanation Penalization* (RIEGER et al., 2020), have shown that introducing the interpretability information into the cost/error function may guide the training process to learn how to use features that are relevant to the problem. However, some interpretability methods (i.e., Vanilla (SIMON; RODNER; DENZLER, 2014; SIMONYAN; VEDALDI; ZISSERMAN, 2014)) already compute the gradient of model output with relation to the input; adding them into the loss function will be an additional computational cost because the final loss function will compute second-order derivatives. Besides, some methods depend on a specific interpretability method to work, and identifying the best interpretability method is difficult.

Instead of optimizing the interpretability output to align with the knowledge specialist mask, let's pose a question: if interpretability highlights important features for model inference, what happens if we generate a new input sample by removing features that the model deems highly important but the knowledge specialist does not consider essential for the problem? Figure 6 illustrates this question with a motivation example. For this example, we trained a U-Net model (RONNEBERGER; FISCHER; BROX, 2015) to perform gray matter segmentation (PRADOS et al., 2017), and it achieved a Dice score (DICE, 1945a) of 0.91 on the validation data, which is an excellent result. However, suppose we exclude some pixels of the input image (not related to gray matter pixels), even keeping all the gray matter pixels in the image. In that case, the U-Net model is no longer able to segment the input image correctly (Figure 6 b)). This behavior can be a problem because the model is not using the gray matter pixels to segment the image; it is using contextual pixels of the gray matter or even shortcut learning pixels.

This motivation example helps us to raise another question: What if we train the model with this modified data? Will the model learn to focus on the right features? We explore this question by introducing a method called Active Image Data Augmentation (ADA)[1]. The fundamental idea behind ADA is to remove information from the input vector that the model attributes high importance, but a domain export does not. ADA dynamically adjusts training data based on

[1] This chapter is based on these two works: 1) Santos, Flávio Arthur Oliveira, Cleber Zanchettin, Leonardo Nogueira Matos, and Paulo Novais. "Active image data augmentation." In Hybrid Artificial Intelligent Systems: 14th International Conference, HAIS 2019, León, Spain, September 4–6, 2019, Proceedings 14, pp. 310-321. Springer International Publishing, 2019. URL <https://link.springer.com/chapter/10.1007/978-3-030-29859-3_27> and 2) Arthur Oliveira Santos, Flávio, Cleber Zanchettin, Leonardo Nogueira Matos, and Paulo Novais. "On the Impact of Interpretability Methods in Active Image Augmentation Method." Logic Journal of the IGPL 30, no. 4 (2022): 611-621. URL <https://academic.oup.com/jigpal/article-abstract/30/4/611/6123345>.

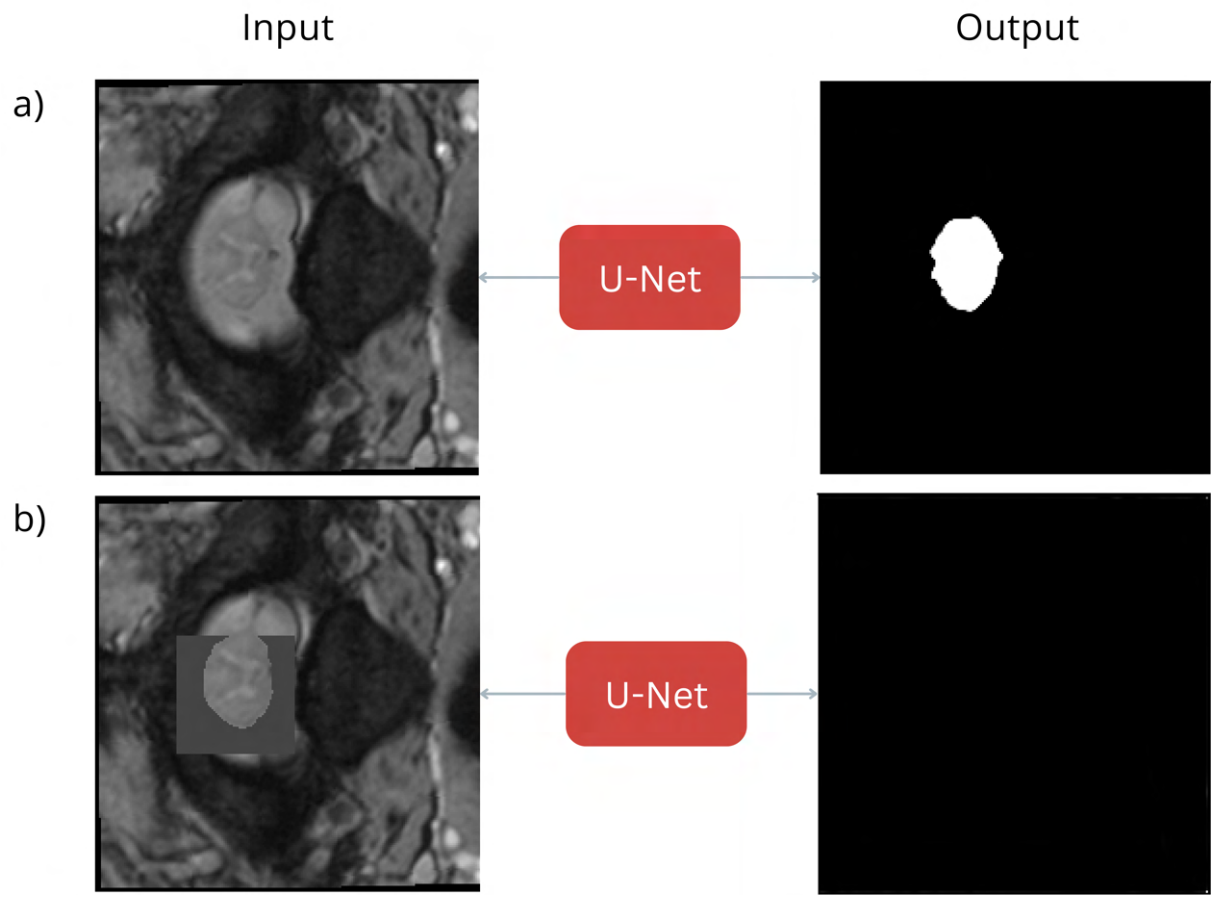

Figure 6 – **Motivation problem.** a) Given the original input image, the U-Net model correctly segments its spinal cord grey matter region. b) After we compute the model interpretability from the output a) and erase the most important region unrelated to the grey matter region (signal), the model can no longer segment the spinal cord grey matter region, even if it is on the input image.

the interpretability of its inferences, and it does not rely on a specific interpretability method or use the second-order derivative in the cost function.

3.1 METHOD

ADA is an approach we proposed to improve the robustness of models and guide the training process to focus on signal features for the given task. In summary, ADA involves generating new training samples in each training cycle. First, ADA uses an interpretability method to identify how important is each input feature for model inference. Then, it selects the most important region of size $N \times N$, which is not in the signal, and all pixel values in this region are set to 0 (removing its information). After applying ADA systematically, we hypothesize that the model will learn to focus on regions with features related to the target task. In the following, the equations 3.1-3.4 illustrate the step-by-step process of the ADA method.

$$y = f(x; \theta) \tag{3.1}$$

The Equation 3.1 illustrates the inference process of a model. In this equation, the function f represents the model, x is the input vector, θ is the model's parameters f, and y is the result obtained from the inference $f(x)$.

$$maps(c, x) = \left\| \frac{\partial y_c}{\partial x} \right\| \tag{3.2}$$

The Equation 3.2 represents the step of constructing the interpretability map of the model's prediction with respect to the input vector. The variable c represents the category/class for which we will obtain the interpretability maps. The equation is an example of the *Vanilla Gradient* interpretability method (SIMON; RODNER; DENZLER, 2014), but different ones can be used in this context.

Next, the function $build\_mask$ (Equation 3.3) represents the step where a binary mask is constructed to erase information important to the model but irrelevant to the task.

$$mask = build\_mask(maps(c, x), ground\_truth, n, z) \tag{3.3}$$

In equation 3.3, a binary mask of dimensions $N \times N$ is constructed, but only a contiguous region of dimensions $Z \times Z$ has a value of 0; all other values are changed to 1. The $Z \times Z$ region is the most important for the model's inference according to $maps(c, x)$. The function $build\_mask$ can be implemented in a computationally efficient manner through the use of dynamic programming.

Next, equation 3.4 uses the mask returned by $build\_mask$ to create the new training data. The operation $*$ denotes element-wise multiplication between two vectors. The resulting vector x_{new} represents the new input data close to x but with removed information from the most important region ($Z \times Z$).

$$x_{new} = x * mask \tag{3.4}$$

It is important to note that the ADA method differs from randomly removing a region from the input data because it chooses what to remove based on interpretability methods. Thus, we can consider it as a form of interpretability-guided data augmentation.

3.1.1 Training method using Active image data augmentation

Algorithm 1 presents the ADA method, and Figure 7 graphically represents its training pipeline. ADA consists of two main steps: (1) training the model for $standard\_epochs$ using the original data (Lines 2-3), thus it can learn the original data distribution; and (2) executing several cycles in which each generates new data using the ADA method (Lines 4-7) and trains the model for ada_epochs using this new data (Lines 6-7).

The ADA training method runs $(standard\_epochs + cycles * ada\_epochs)$ training epochs. This number is an important choice as it is directly related to the computational cost of the ADA.

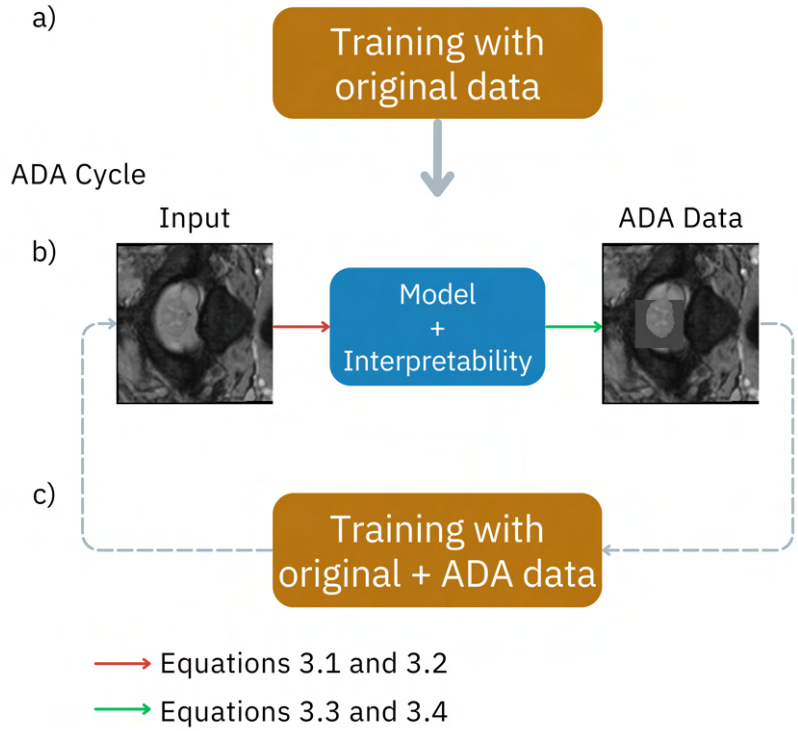

Figure 7 – Graphical representation of the ADA training method. a) first, the model is trained during *standard_epochs* epochs using only original data; b) second, for each image in the training set, we generate the ADA data, and in step c) we train the model during *ada_epochs* using original and ADA data. The steps b) and c) are executed *cycles* times.

Algoritmo 1: ADA's training method.

Input: $Ada\_Training, model, data, cycles, ada\_epochs$

$conventional\_data \leftarrow data + classic\_augmentations(data)$;
for $i \leftarrow 0; i < standard\_epochs$ **do**
    $model.train(conventional\_data)$;
for $i \leftarrow 0; i < cycles$ **do**
    $new\_data \leftarrow conventional\_data + Ada(data, model)$;
    for $j \leftarrow 0; j < ada\_epochs$ **do**
        $model.train(new\_data)$;
return $model$

3.2 EXPERIMENTS

In the experiments, we used five interpretability methods in ADA and evaluated their impact on the robustness of the trained models. These methods include: *Vanilla Gradient* (SIMON; RODNER; DENZLER, 2014), *Guided Backpropagation* (SPRINGENBERG et al., 2015), *Grad-Cam* (SELVARAJU et al., 2017), *Guided Grad-Cam*, and *Input x Gradient*. The Spinal Cord Grey Matter Segmentation (SCGM) dataset (PRADOS et al., 2017) was used in this experiment. The *DeepSeg* model (PORISKY et al., 2017) achieved good results in SCGM. As it is based

on a U-Net architecture (RONNEBERGER; FISCHER; BROX, 2015), we decided to use the U-Net architecture to execute these experiments. The following section describes the SCGM dataset, evaluation metrics, U-Net architecture, and details about the experiments.

3.2.1 *Spinal Cord Grey Matter Segmentation* - SCGM

The SCGM dataset consists of Magnetic Resonance Imaging (MRI) images from different patients. It comprises 80 sub-datasets, with 40 designated for training and 40 for testing. According to SCGM (PRADOS et al., 2017), each group of 20 datasets was acquired from the following institutions: University College London, Polytechnique Montreal, University of Zurich, and Vanderbilt University. More specific details about how the data were acquired can be found in (PRADOS et al., 2017). As SCGM does not have a defined validation dataset, we used 20% of the original training data to produce the validation data. Our experiment also includes a robustness scenario, which is crucial for evaluating the efficiency of the ADA method. We used the validation data to construct this scenario since we do not have access to the test data of SCGM. The evaluation process for SCGM is conducted through an online system [2].

3.2.2 Metrics

The evaluation process of the proposed method uses five different evaluation metrics. They measure different types of information between the model's output and the labels of the ground truth data. They are divided into three categories: overlap, distance, and statistical. Table 1 summarizes the metrics, including their names, abbreviations, scale ranges, and categories.

Table 1 – Summary of evaluation metrics. Adapted from (PRADOS et al., 2017)

Metric Name	**Abbreviation**	**Range**	**Category**
Dice Similarity Coefficient (DICE, 1945b)	DSC	$0\lor 100$	Intersection
Hausdorff Surface Distance (TAHA; HANBURY, 2015)	HSD	> 0	Distance
Sensitivity (TP)	TPR	$0\lor 100$	Statistical
Specificity (TN)	TNR	$0\lor 100$	Statistical
Precision	PPV	$0\lor 100$	Statistical

In the distance category metric, the lower the obtained value, the better the result. However, in the metrics of the Intersection and Statistical categories, the higher the values obtained, the better the results. It is important to note that the metrics *Sensitivity*, *Specificity*, and *Precision* represent classical metrics such as true positive rate, true negative rate, and precision, respectively.

[2] http://niftyweb.cs.ucl.ac.uk/program.php?p=CHALLENGE

3.2.3 Models

The U-Net (RONNEBERGER; FISCHER; BROX, 2015) used in this work comprises three downsampling layers, three upsampling layers, and an output layer with a logistic sigmoid activation function. It is important to note that after each convolutional layer, a dropout regularization layer (SRIVASTAVA et al., 2014) is used, followed by batch normalization (IOFFE; SZEGEDY, 2015). During the first 100 training epochs, the following traditional data augmentation methods were used: Rotation, Shift, Scale, Channel Shift, and Elastic Deformation. Table 2 presents all the parameters defined based on previous works (PERONE; CALABRESE; COHEN-ADAD, 2018).

The U-Net model was trained for 100 epochs using only the original data. Then, five U-Net models were created and initialized with the same state as this first U-Net at epoch 100. Each of these five instances of U-Net was trained using the ADA method with a specific interpretability method, namely: (i) Vanilla Backprop, (ii) Input X Gradient, (iii) Grad-Cam, (iv) Guided Backprop, and (v) Guided Grad-Cam. These interpretability methods were chosen because they share common characteristics. For example, all of them are based on the gradient vector and do not require a reference input vector. The dropout rate used by U-Net was 0.5, and the batch normalization momentum was 0.4. The optimization method used was Adam (KINGMA; BA, 2014) with an initial learning rate of 0.001 and a batch size of 16. Each instance of the U-Net models was trained for 31 ADA cycles, with each cycle training the instance for 30 epochs using data obtained from ADA with an occlusion region of size 20×20. We chose the epoch of each model with the best validation results to produce the robustness results.

Table 2 – Parameters used in the *data augmentation*.

Method	**Parameters value**
Rotation (degrees)	$[-4.6, 4.6]$
Shift (%)	$[-0.03, 0.03]$
Scaling	$[0.98, 1.02]$
Channel Shift	$[-0.17, 0.17]$
Elastic Deformation	$\alpha = 30.0, \sigma = 4.0$

3.2.4 Robustness Database

We used the validation data from SCGM to construct the robustness dataset. New images were generated with a 20x20 region occluded. Thus, we evaluated the model's performance in a scenario with images containing incomplete information. Algorithm 2 outlines how these new evaluation data, which we refer to as the robustness dataset, were constructed.

The function $erase\_region$ in Algorithm 2 takes the image as input and sets to 0 all pixels within the square of area $Z \times Z$ that starts at position (i, j) and ends at $(i + z, j + z)$.v

Algoritmo 2: Function to build the robustness dataset.

```
Input: Build_Robustness_Data
data, n, z robustness_data ← [] ;
for x ∈ data do
    for i ← 0; i < n; i+ = z do
        for j ← 0; j < n; j+ = z do
            x_occluded ← erase_region(clone(x), i, j, z) ;
            robustness_data.append(x_occluded) ;
return robustness_data
```

3.3 RESULTS AND DISCUSSIONS

Figure 8 presents the convergence curve of the five U-Net models, each using a different interpretability method. This convergence curve was generated by evaluating the U-Net models with the original validation data at each epoch. This analysis reveals that the model using Grad-Cam achieves better validation results in the early epochs, surpassing a *dice score* of 80. In contrast, the best among the other methods achieves a *dice score* of 79. This result indicates that the model did not generalize enough before applying the ADA method. However, after the initial epochs of ADA with Grad-Cam, the model improved validation performance after a few epochs. In addition to the promising early results of U-Net with Grad-Cam, the curve shows that after epoch 123, all methods perform similarly, suggesting they converge to almost the same point.

After analyzing Figure 8, Table 3 highlights the results of each model on the validation data. Each model achieves a similar best Dice score (DSC) to the others, but the U-Net with Guided Gradient exhibits the highest DSC, followed by the U-Net using Grad-Cam. It is important to note the difference in precision and recall achieved by Grad-Cam compared to the other methods.

Table 3 – Results obtained using the validation data from SCGM.

Models	**DSC**	**HSD**	**TPR**	**TNR**	**PPV**
Vanilla Backprop	86.56	1.72	87.72	99.95	86.21
InputXGradient	86.45	**1.70**	87.33	99.95	86.38
Guided Gradient	**86.72**	1.71	87.57	99.95	**86.70**
Grad-Cam	86.61	1.72	**89.53**	99.94	84.61
Guided Grad-Cam	86.56	**1.70**	87.65	99.95	86.34
U-Net	87.00	4.79	96.92	99.07	78.95

The main objective of ADA is to improve the model's robustness. Therefore, creating a dataset to evaluate robustness in the proposed scenario was necessary. This dataset was built from the validation data. Section 3.2.4 presents the details related to the construction of this

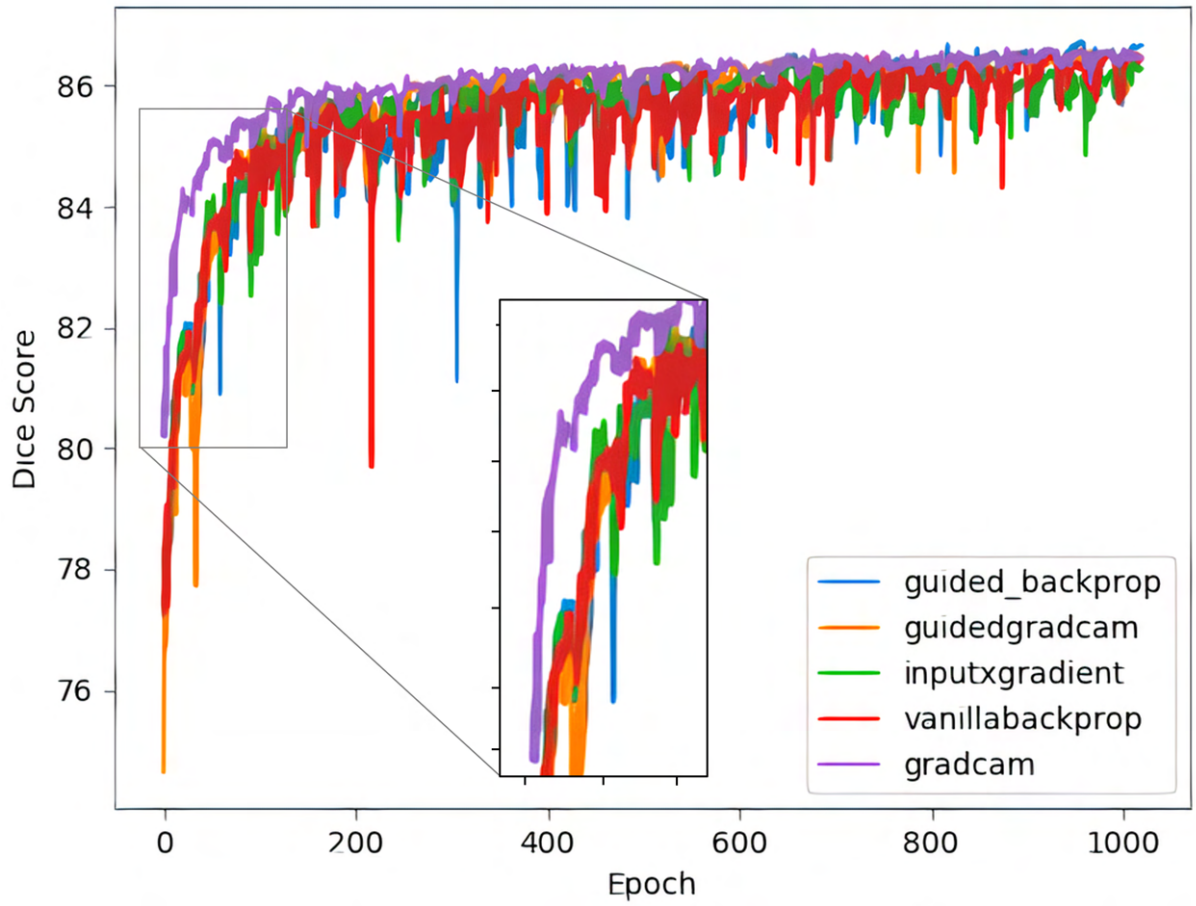

Figure 8 – Analysis of the convergence of the model. The x-axis means the epoch and the y-axis means the Dice score after trained in the respective epoch. The zoomed-in region shows that Grad-Cam performs better than other interpretability methods in the early training epochs

database. All results presented so far using the original validation data were important only to show that the models trained using ADA maintain the original quality of the U-Net without ADA.

Table 4 presents the results obtained from the database built to analyze robustness. They suggest that the U-Net with Grad-Cam performs better than all other models. Additionally, the precision-recall curve behavior of the U-Net model with Grad-Cam is different from all others, as the precision is relatively lower than the others, and the recall is higher. These results suggest that the ADA method yields better results in the robustness scenario. Moreover, they provide strong evidence that the Grad-Cam method can guide the U-Net model to focus on information that is truly relevant to solving the problem.

3.3.1 Analysis of the Initial Training Steps with ADA

The results obtained showed that the impact of interpretability methods on ADA is primarily linked to the early epochs of its execution, which is an interesting behavior as it reduces the

Table 4 – Results obtained using the robustness dataset.

Models	**DSC**	**HSD**	**TPR**	**TNR**	**PPV**
Vanilla Backprop	85.86	1.72	86.73	99.94	85.99
InputXGradient	85.76	1.71	86.36	99.95	86.15
Guided Gradient	86.03	1.71	86.57	99.95	**86.50**
Grad-Cam	**86.15**	1.73	**88.84**	99.94	84.45
Guided Grad-Cam	85.90	**1.70**	86.68	99.95	86.12

need for many iterations as well as computational costs. To analyze this phenomenon in detail, this section analyzes the interpretability maps obtained by different methods in the initial steps of ADA.

The interpretability methods used in ADA suggest which region of the input image should be occluded to construct the new training image. Thus, given a new image, we calculate the IoU index between the regions that the method deems important to assess their similarity. Figure 9 presents an example of the original input image and the image obtained after applying ADA. The IoU is calculated based on the mask used to occlude each region of the input image.

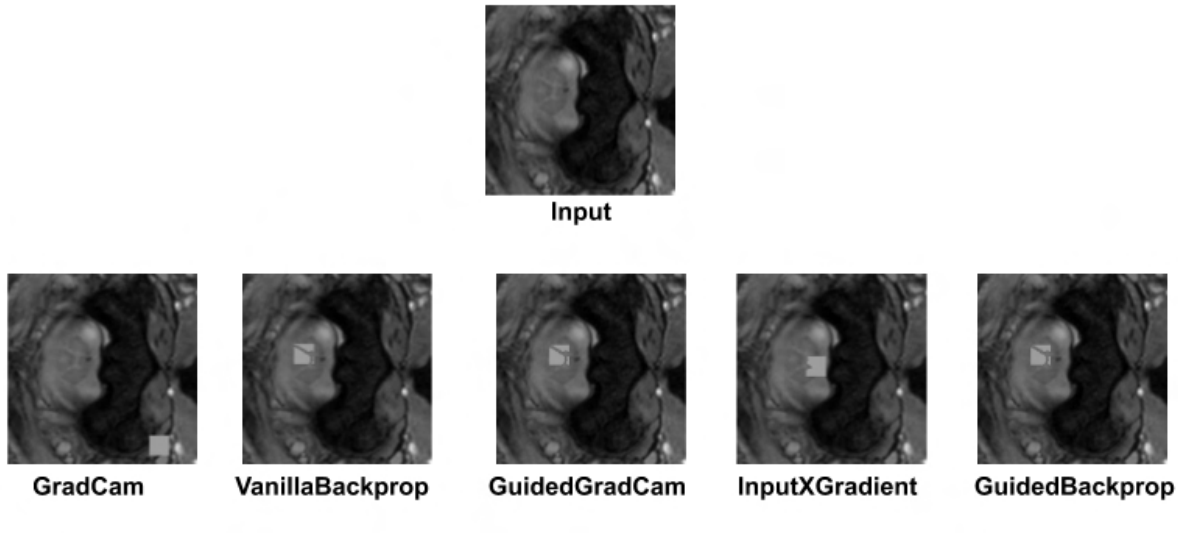

Figure 9 – Data augmentation samples obtained from the first ADA cycle. Given an input image, we compute the most important region for each interpretability method, and then we compute the IoU metric between their masks.

The IoU matrix in Figure 10 presents the average of all IoU matrices calculated from all images in the training data. Since the IoU matrix is obtained in the early steps of ADA, and Grad-Cam showed better results in the initial epochs, this experiment demonstrates that Grad-Cam produces new training images different from those produced by other methods.

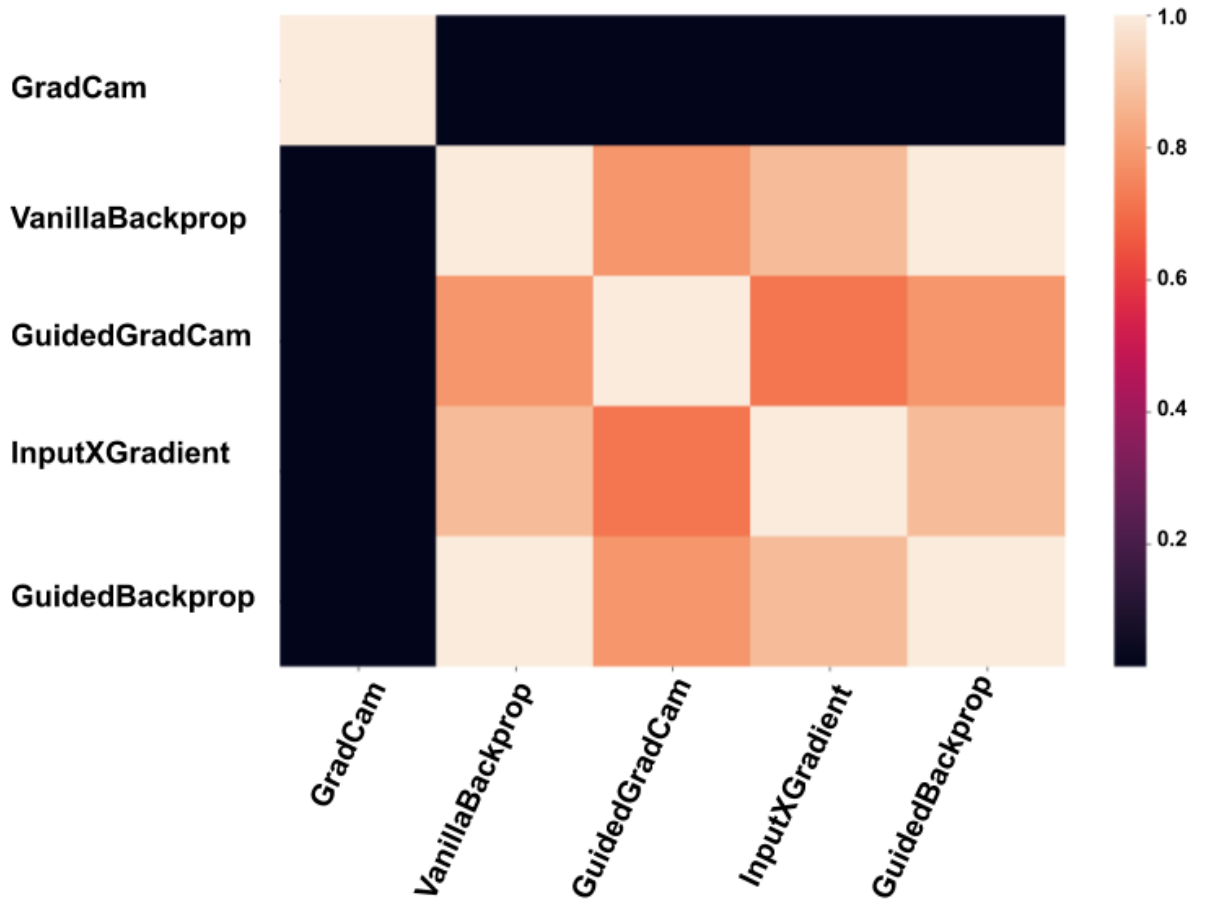

Figure 10 – IoU matrix between all methods. Grad-Cam presents a lower IoU when compared to all other methods. This implies that the regions occluded by it are very different from those occluded by the others, as shown in the figure 9.

3.4 CONCLUSION

Active Image Data Augmentation aims to improve the model's robustness by helping them focus on important information in the input vector. ADA uses interpretability methods in the process, so we evaluated it using different interpretability methods to identify their influence on the final model performance.

The results obtained showed that all models using different interpretability methods achieved competitive results. Still, they suffered a tiny performance drop on the original test set compared to a model trained only with original data. All models achieved closed dice score values on original test set, which suggest that in future work is necessary to apply statistical test in a more robust experiment to verify if there is statistical significance in the results. Additionally, in the robustness evaluation, all models showed similar results, but the approach using Grad-Cam yielded better results in the early epochs. This suggests that Grad-Cam may guide ADA to help the model focus on the signal information from the input vector in the early epochs.

4 RIGHT REASONS DATA AUGMENTATION

Several recent works have proposed new loss functions to guide the model to focus on signal features (ROSS; HUGHES; DOSHI-VELEZ, 2017; SCHRAMOWSKI et al., 2020; VIVIANO et al., 2021; SIMPSON et al., 2019; RIEGER et al., 2020; ERION et al., 2019), thereby using signal information instead of contextual information in the inference process. These methods are referred to in the literature as Right for the Right Reasons (RRR) approaches. These loss functions generally use second-order gradient optimization (DRUCKER; CUN, 1992) and incorporate a right reasons factor into the loss function. The right reasons factor is responsible for encouraging the model to use the signal information in decision-making. Equation 4.1 presents a generic loss function for RRR training. This generic loss function consists of a $Right\_answer$ factor to guide the model to decide correctly and a $Right\_reason$ factor to instruct the model to focus on signal information. The terms λ_1 and λ_2 are parameters to weigh the contributions of both factors.

$$\begin{aligned} L(\theta, X, y, A) = \lambda_1 Right\_answer(\theta, X, y) \\ +\lambda_2 Right\_reason(\theta, X, y, A) \end{aligned} \tag{4.1}$$

In chapter 3 we argue that the RRR methods have an additional computational cost and some of them need a specific interpretability method. Thus, we propose Active Image Data Augmentation (ADA) to mitigate these issues. However, the ADA method also has weaknesses. For example, we need to evaluate what is the best interpretability method for it, the number of standard training epochs, the number of ADA cycles, the number of ADA epochs, the size of the patch that will be removed, and which information we will insert in the removed region. In addition, we would like to highlight that all of these RRR methods (including ADA) depend on the fairness of interpretability methods.

All the RRR methods try to adjust the model during the training and ignore that context/background dependence may be a data issue instead of a model issue. Rather than explicitly optimizing the model to focus on signal features, we would like to wonder: what happens if we have a batch of inputs of different categories with the same background (i.e., Figure 11)? We assume the model will try to use the signal features as this is the only difference between the inputs. Based on this motivation, in this chapter, we propose the Right reason data augmentation (RRDA)[1] to learn robust and fair model.

[1] This chapter is based on Santos, Flávio Arthur Oliveira, and Cleber Zanchettin. "Exploring Image Classification Robustness and Interpretability with Right for the Right Reasons Data Augmentation." In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops, pp. 4147-4156. 2023. URL <https://openaccess.thecvf.com/content/ICCV2023W/LXCV/html/Santos_Exploring_Image_Classification_Robustness_and_Interpretability_with_Right_for_the_ICCVW_2023_paper.html>

Figure 11 – **Right reasons data augmentation - Motivation.** This sample illustrates a batch of three images of three different categories, but they have the same image. We argue that if the discriminative information is only the signal features, the model will learn to focus on this information during the inference step.

4.1 RIGHT FOR THE RIGHT REASONS - A DATA-CENTRIC PERSPECTIVE

Right for the Right Reasons (RRR) is a property of models relating to their robustness, fairness, and reliability. RRR models are trained to extract pertinent patterns from the input signal and make inferences based on meaningful signals rather than spurious correlations. According to the Cambridge Dictionary, context is the situation within which something exists or happens, and that can help explain it[2]. Thus, context should not be the primary focus, but rather, it should assist in understanding the primary focus. Issues with data can lead models to learn shortcuts from context information (GEIRHOS et al., 2020), correlating context information with the input label and resulting in an unfair model. Several works (ROSS; HUGHES; DOSHI-VELEZ, 2017; SCHRAMOWSKI et al., 2020; SIMPSON et al., 2019; RIEGER et al., 2020; ERION et al., 2019; VIVIANO et al., 2021) propose new optimization loss functions for the model to learn to ignore non-signal information.

Consequently, after the training process, models will learn to extract patterns related to the signal. In this work, we take a different approach by proposing a data-centric perspective to achieve RRR. We argue that if a model is trained on the Right Reasons Data (RRD), it will inherently be RRR. In the following sections, we present the Right for the Right Reasons Data (RRRD) concept and discuss how to transform raw data into right reasons data.

RRRD assumes an input data vector $x = [x_1, x_2, \ldots, x_n]$ that comprises both class-informative (signal) and context-informative features. *We argue that if, after training a model f with a dataset D, it correlates a set of context-informative features C with label y, this is likely because D is context-biased, and the context C only appears in input samples with label*

[2] https://dictionary.cambridge.org/us/dictionary/english/context

y. Therefore, D cannot be considered an RRRD dataset because its context information alone is enough for the model to discriminate between samples. Next, we present the definitions necessary to understand this concept. These definitions assume the existence of an oracle O that is robust, fair, and trustworthy.

Definition 4.1.1 *Given an input vector I of category c, a subset of features, denoted as IC, is defined as 'class-informative' if it is sufficient for the model O to classify I as category c.*

Definition 4.1.2 *Given an input vector I of category c, a subset of features, denoted as C, is defined as 'context-informative' if its intersection with IC is empty, and it is insufficient on its own for the model O to classify I as category c.*

Definitions 4.1.1 and 4.1.2 provide clarity on what we consider class-informative and context-informative features. Moreover, these definitions imply that IC and C are disjoint sets, and their union constitutes the complete input vector.

4.1.1 Right Reasons Data Augmentation

This section discusses the issue of models learning patterns from context rather than signal when the data correlates context and label. We propose a solution through a data augmentation method named Right Reasons Data Augmentation (RRDA). It aims to transfer context information between data samples, encouraging the model to utilize signal information for discrimination, thereby enhancing fairness and robustness. Figure 12 presents a sample of RRDA performed on a batch of images. Algorithm 3 provides a pseudo-code illustrating its generic implementation.

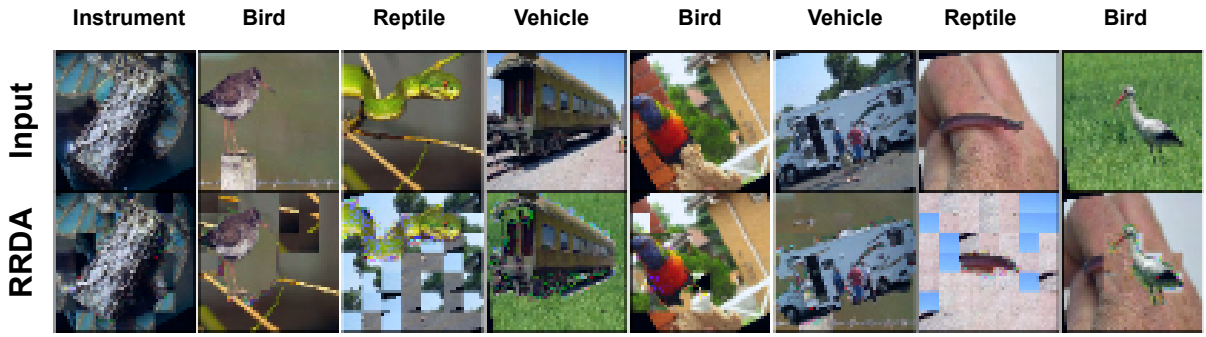

Figure 12 – **Example of RRDA performing data augmentation on a batch of 8 input images.** Each column represents an input image. The first row shows the original input batch, and the second row shows the output obtained from the RRDA algorithm.

Given a batch of samples X, the labels y, and a binary context information mask CI, the RRDA algorithm 3 iterates over each batch sample (Line 4) and selects a random one to replace its context information (Lines 5-7). It then adds the new samples and their respective

labels to a new batch list (Lines 8-11). The primary objective of the RRDA algorithm is to embody the idea of context shift in a generic manner. For simplicity, it presumes the data is structured and all context and class-informative features are in the same position for all samples. Therefore, when computing new_left (Line 6), we insert zeros in $X[left]$ context and add the context information from $X[right]$. This process may not apply to unstructured data (e.g., images and text) because context and signal positions often vary for each sample in the dataset. Consequently, specific implementation for each domain must address this issue.

Algoritmo 3: RRDA algorithm

1: **procedure** RRDA(X, y, CI) ▷ Compute RRDA for a batch X
2: $rrda\_batch \leftarrow []$ **for** $left \leq len(X)$ **do**
3: $right \leftarrow random(len(X))$
4: $new\_left \leftarrow (1 - CI[left]) \odot X[left] + X[right] \odot CI[right]$
5: $new\_right \leftarrow (1 - CI[right]) \odot X[right] + X[left] \odot CI[left]$ ▷ Replace context between two samples.
6: $rrda\_batch.append((new\_left, y[left]))$
7: $rrda\_batch.append((new\_right, y[right]))$
8:
9: **return** $rrda\_batch$ ▷ The RRDA new batch and labels
10: **end procedure**

4.2 EXPERIMENTS AND RESULTS

Evaluating RRR is challenging as it requires a task with both signal and context information. Additionally, data manipulation and the creation of new data are necessary for assessing the model's robustness in the face of context shifts, thereby extending beyond the typical test accuracy evaluation. Background sensitivity serves as a task for evaluating the impact of image backgrounds on object recognition models. We used this task to evaluate RRR methods and the proposed RRDA. If the model can ignore the background information and exhibit robustness to context shifts, it indicates a focus on the signal information. This aligns with the requirements and scope of this work.

4.2.1 Datasets

To evaluate background sensitivity, we need datasets with image labels and object segmentation. Therefore, we utilize the ImageNet-9 (XIAO et al., 2021) challenge, which is specific to background robustness, and construct a similar background challenge with RIVAL10 (MOAYERI et al., 2022).

ImageNet-9 (IN-9) (XIAO et al., 2021) is a dataset designed for evaluating background sensitivity in object recognition. It is a subset of ImageNet (DENG et al., 2009b) and consists of nine classes, each containing 5.045 training and 450 testing images. The image bounding box

annotations, crucial for evaluating background sensitivity, are not abundant for each category in the original ImageNet split. Consequently, the authors of IN-9 grouped the ImageNet categories according to their ancestors in the WordNet (MILLER, 1995) hierarchy. In addition to the raw images from ImageNet, IN-9 includes seven synthetic dataset variations intended to assess the background sensitivity of image classification models. These variations result from the processing of foreground or background elements in the original dataset. Figure 13 provides a visual example of each dataset variation.

RIVAL10 (MOAYERI et al., 2022), a subset of ImageNet aligning with the CIFAR10 dataset classes and comprising roughly 26k images, also offers object segmentation for each image and comprehensive attribution annotation for each object. To verify the generalization of our proposal, we employ the full object segmentation from RIVAL10 to generate the mixed-same, mixed-rand, mixed-next, and only-fg variations.

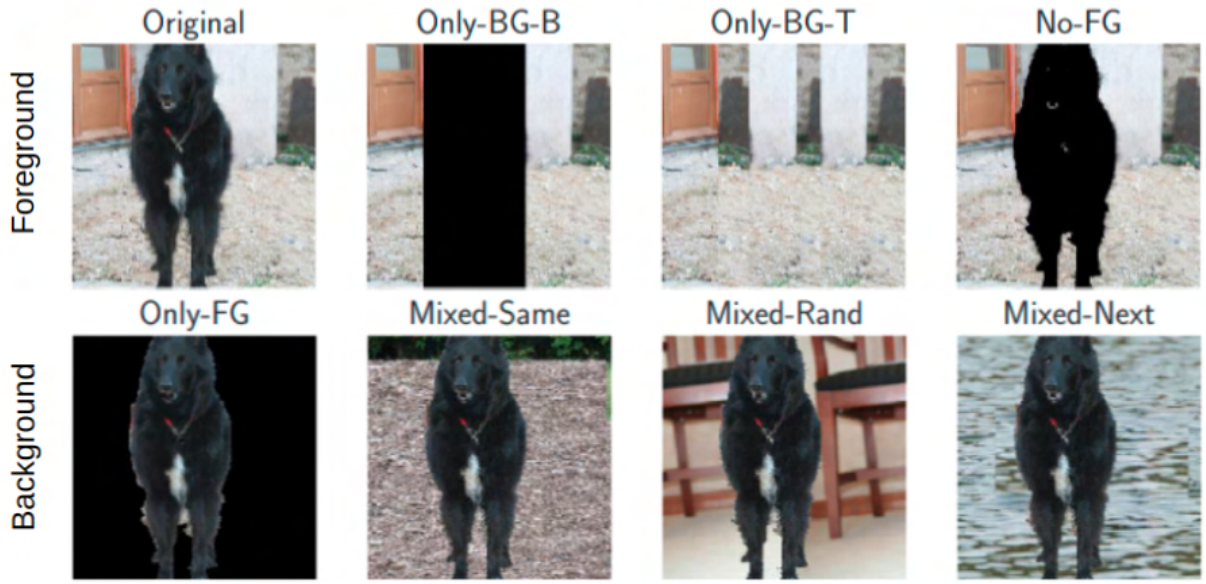

Figure 13 – **ImageNet-9 challenges.** The top row displays samples of challenges that alter the foreground information, while the bottom row introduces the challenges that modify the background information. The original challenge includes images with neither foreground nor background information changes. In the 'Original' scenario, the original background of the image is used. 'BG' refers to the background, and FG' to the image foreground. In the 'Mixed Same' scenario, the background is swapped with the background of another image belonging to the same class. In the Mixed Rand scenario, the background is swapped with the background of another image from a different random class. In the 'Mixed Next' scenario, the background is swapped with one of another image belonging to the next class, i.e., if the class index for the image is 2, then we swap backgrounds with an image from class 7.

Table 5 – **Challenge results for ImageNet9 dataset.** The table organizes the results by dataset, with each row representing an evaluation. The columns 'Architecture' and 'Method' represent the architecture and training method used. The 'ImageNet-9' column represents the results for the model trained with each respective dataset. The 'Original', 'Mixed same', 'Mixed rand', and 'Mixed next' columns represent the accuracy results for each challenge, while the 'BG-Gap' column represents the difference between the 'Mixed rand' and 'Mixed same' results.

		ImageNet-9				
Architecture	Method	Mixed same	Mixed rand	Mixed next	BG Gap	Original
ResNet-18	Standard	92.67	82.99	80.22	9.68	96.15
ResNet-18	ActDiff	90.27	84.47	83.26	5.80	93.46
ResNet-18	GradMask	86.77	76.34	73.43	10.42	90.79
ResNet-18	ADA	92.20	81.80	79.28	10.40	96.05
ResNet-18	RRR	91.90	82.12	78.77	9.78	95.31
ResNet-18	ActDiff + RRDA	89.56	85.90	84.89	3.65	92.49
ResNet-18	Standard + RRDA	88.30	83.41	82.37	4.88	90.62
ViT	Standard	94.15	86.84	84.69	7.3	98.35
ViT	ActDiff	95.98	90.27	89.46	5.7	98.99
ViT	GradMask	93.38	86.52	84.77	6.7	97.04
ViT	ADA	91.73	81.98	80.12	9.7	97.24
ViT	RRR	90.42	80.04	78.54	10.4	96.74
ViT	Standard + RRDA	**97.28**	**96.00**	**95.88**	**1.28**	**99.06**
ViT	ActDiff + RRDA	96.12	93.26	93.04	2.86	98.79

4.2.2 Implementation details

We used two pretrained models in the experiments, specifically the ResNet-18 from Torchvision[3] (HE et al., 2015) and ViT (DOSOVITSKIY et al., 2020) from timm[4]. Both models were end-to-end fine-tuned with Stochastic gradient descent (SGD) by 50 epochs using a learning rate equal to 0.001 and batch size 32. In addition, we ablate the weight contribution of the Right answer factor (i.e., $\lambda 2$ in equation 4.1) for all non-standard methods and present the best result in the paper. Table 7 presents the $\lambda 2$ ranges used in the experiments.

[3] ResNet18 model from <https://pytorch.org/vision/main/models/resnet.html>

[4] vit_base_patch16_224_in21k model from <https://github.com/huggingface/pytorch-image-models>

Table 6 – **Challenge results for RIVAL10 dataset.** This table is structured in the same way as table 5.

Architecture	Method	RIVAL10				
		Mixed same	Mixed rand	Mixed next	BG Gap	Original
ResNet-18	Standard	95.01	87.82	88.65	7.19	99.19
ResNet-18	ActDiff	94.91	86.55	87.16	8.36	98.77
ResNet-18	GradMask	90.65	83.96	84.34	6.69	96.61
ResNet-18	ADA	95.20	88.64	89.35	6.55	99.07
ResNet-18	RRR	94.82	87.89	88.67	6.92	98.90
ResNet-18	ActDiff + RRDA	96.25	94.57	94.21	1.68	98.52
ResNet-18	Standard + RRDA	95.38	93.93	93.89	1.46	96.80
ViT	Standard	95.31	87.99	88.61	7.32	99.24
ViT	ActDiff	96.92	92.26	91.47	4.65	99.62
ViT	GradMask	96.52	90.81	91.09	5.71	99.49
ViT	ADA	96.27	88.84	90.09	7.45	99.69
ViT	RRR	53.01	34.09	35.19	18.94	64.76
ViT	Standard + RRDA	96.67	**96.44**	**96.48**	**0.23**	97.81
ViT	ActDiff + RRDA	**97.69**	94.45	94.09	3.24	**99.75**

Table 7 – **Regularizer rate values used during training for the right for the right reasons methods.**

Method	$\lambda 2$
ActDiff	$[2^0, 2^3]$
GradMask	$[10^{-5}, 5 * 10^{-3}]$
RRR	$[10^1, 10^3]$

4.2.3 Background Challenge Results

Table 5 presents the results from the Background challenge. The $orig.$ column displays the results obtained from the original test split. All models achieve an accuracy above 90%, except for ViT with RRR, which does not generalize well on the RIVAL10 dataset. The BG-Gap column represents the difference between the Mixed Rand and Mixed Same results, indicating the variation in model accuracy when evaluated with biased (i.e., a background of the same class) and unbiased backgrounds. Therefore, a lower BG-Gap reflects a more robust model capable of handling backgrounds from different categories. It is important to highlight that

the BG-Gap should be analyzed jointly with original accuracy because a perfect model (i.e., 100% on mixed rand and same) and a random model (i.e., 10% accuracy on mixed rand and same) will have BG-Gap equal to zero. The results suggest that not all RRR methods are robust to background sensitivity, as evidenced by the BG-Gap from GradMask, ADA, and RRR being worse than Standard training with the ResNet architecture on IN-9. Furthermore, the best BG-Gap on both datasets was achieved by the ViT trained with Standard + RRDA.

Does Robustness Depend on Dataset Characteristics? The results reveal a significant difference between the accuracies on IN-9 and RIVAL, suggesting that specific dataset features, such as categories, number of classes, image distribution, and the relationship between signal and background, may considerably impact model robustness.

On the connection between the challenges. Figure 14 presents the correlation between the results of the challenges for each dataset. These results indicate a strong correlation between the Mixed same and Original, as well as between the Mixed rand and Mixed next scenarios. This result suggests how the models correlate signal with background information because mixed same have background information from the same categories, and mixed next as well mixed rand have the background from different classes.

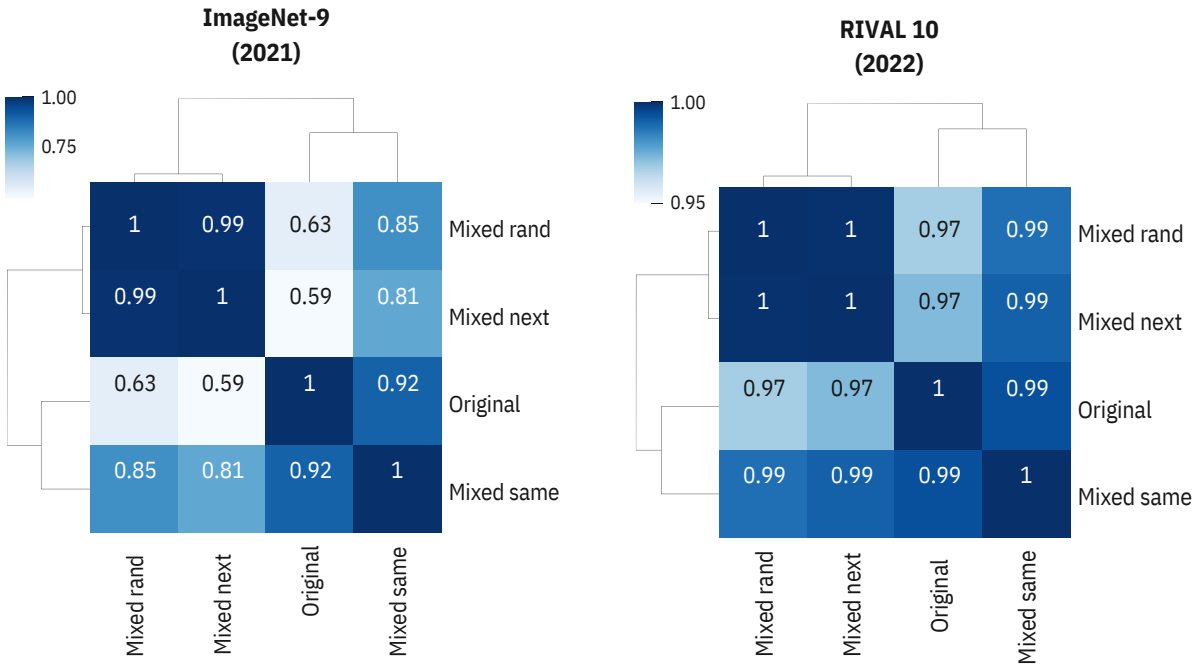

Figure 14 – **Correlation between the challenges.** We compute the Person correlation between the challenge results for each dataset. It shows a positive correlation between all pairs, but the correlation between Mixed same and Original, and Mixed rand and Mixed next are the higher values in both datasets.

4.2.4 Analysis of BG-Gap distributions

Is background robustness architecture dependent? Supervised learning design encompasses three major components: the model, the data, and the optimization loss. Thus far, we

have primarily discussed different data and optimization loss functions to guide the model to adhere to the RRR principle. We aim to evaluate the impact of the architectural choice on robustness. We specifically highlight the difference between the results of the ResNet and ViT architectures, as shown in Figure 15a. The figure illustrates that the ViT architecture is more robust than ResNet, achieving a background gap minimum that is at least twice as low as that of ResNet on both datasets. Furthermore, both the maximum and median background gaps of ViT are lower than those of ResNet (PAUL; CHEN, 2022). These findings are in line with existing literature, where authors have argued that ViT exhibits greater robustness than ResNet in terms of image transformations (PAUL; CHEN, 2022).

Does RRDA impact background robustness? Figure 15b compares the BG-Gap distributions with and without RRDA. It demonstrates a substantial impact of RRDA on BG-Gap, with high-density values for low BG-Gap approaching 0. In contrast, the BG-Gap for models without RRDA is close to 10 for both datasets. Additionally, the median values exhibit stark differences between the two situations. These results indicate that architecture design is crucial in model fairness and robustness. This suggests a new direction for research, focusing on the development of "right for the right reasons" architectures rather than solely on data and optimization loss functions. Additionally, the analysis of BG-Gap distributions suggests that RRDA significantly impacts model background robustness.

Is BG-Gap dependent on original accuracy? Figure 16 presents the correlation between the BG-Gap metric and original accuracies. The correlation on RIVAL10 results is positive, while in ImageNet-9, it is negative. Nevertheless, although in both cases, the correlation is not strong enough, these insights could have significant implications for the training of deep learning models for image classification, suggesting that striving only for high accuracy might inadvertently lead to models that overfit the background of images and the more robust model is not necessarily the best in test accuracy.

4.2.5 Model Dependence on Edge Information

Edge information is vital for image recognition as it represents boundaries between different pieces of information, such as the foreground and background. In this section, we question whether this information is necessary for models to make correct inferences and how robust they are to changes in edge information. We create new variations of the original IN-9 and RIVAL-10 datasets to perform the edge analysis by removing edge information with a fixed width W. As the edge represents the transition between object information and background information, we occlude parts of the signal and background by replacing it for black pixels, thereby eliminating this transition. Figure 17a) presents an example of the model dependence on the edge information pipeline, using an image from the original set and its version with the edge removed.

We applied edge removal to the IN-9 and RIVAL-10 original sets and all background variations, namely Mixed-same, Mixed-rand, and Mixed-next, with edge sizes varying from 5 to

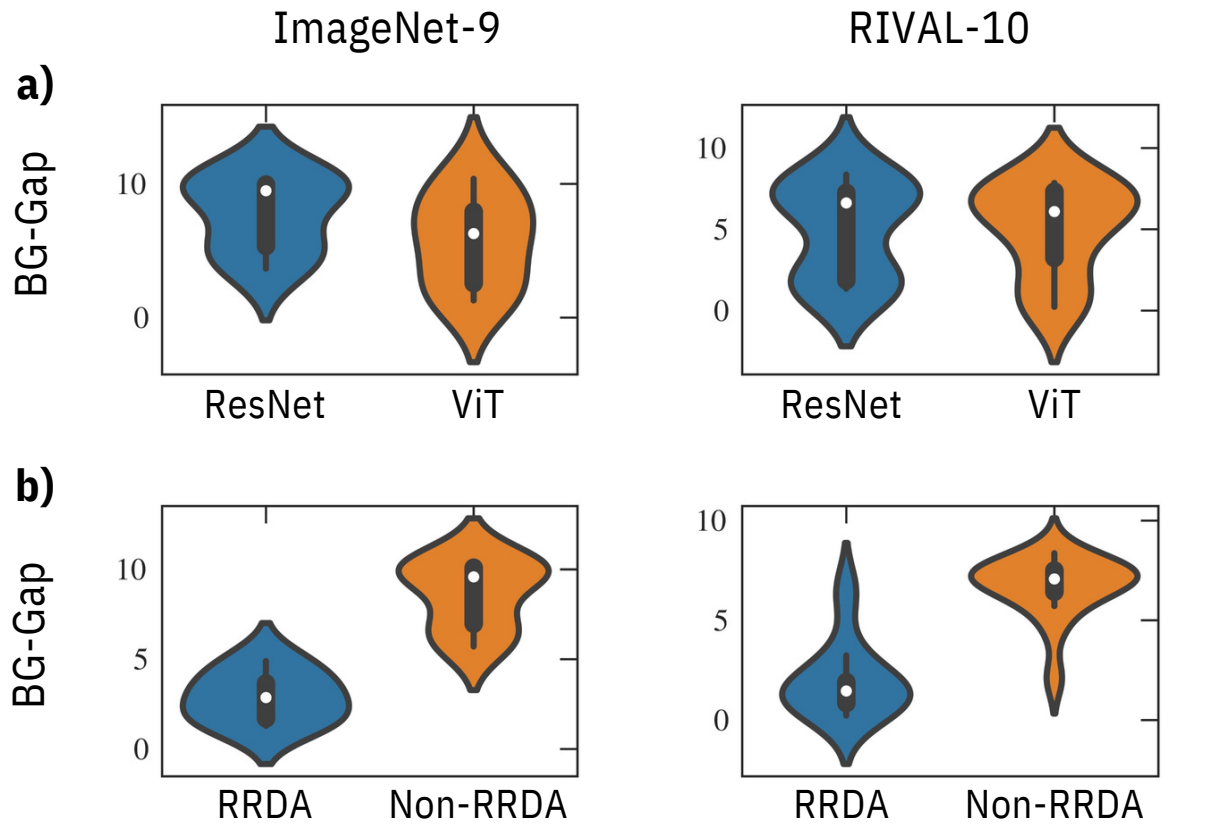

Figure 15 – BG-Gap distributions for different configurations. The BG-Gap distribution is built from the BG Gap column in Table 5. The a) plot shows the comparison between ResNet and ViT architectures based on BG-Gap distribution for ImageNet-9 and RIVAL-10 datasets, while the b) plot compares the BG-Gap when we use RRDA with when we do not use (i.e., Non-RRDA).

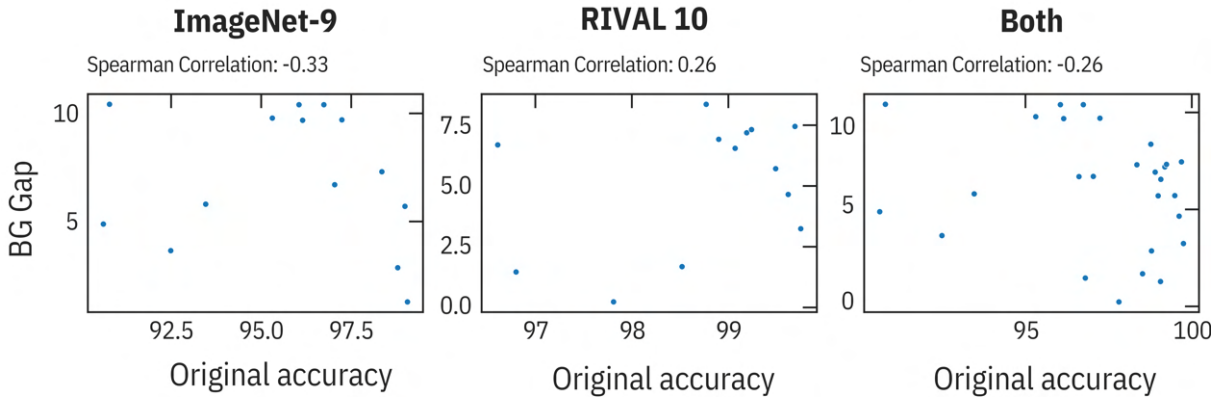

Figure 16 – **Correlation between BG-Gap and Original accuracy.** We compute the Spearman correlation between all original accuracies higher or equal to 80% and BG-Gaps for each dataset scenario. In addition, we concatenate them both and compute the Spearman correlation (i.e., 'Both' scenario). The results do not present a strong positive or negative correlation between the values in all scenarios.

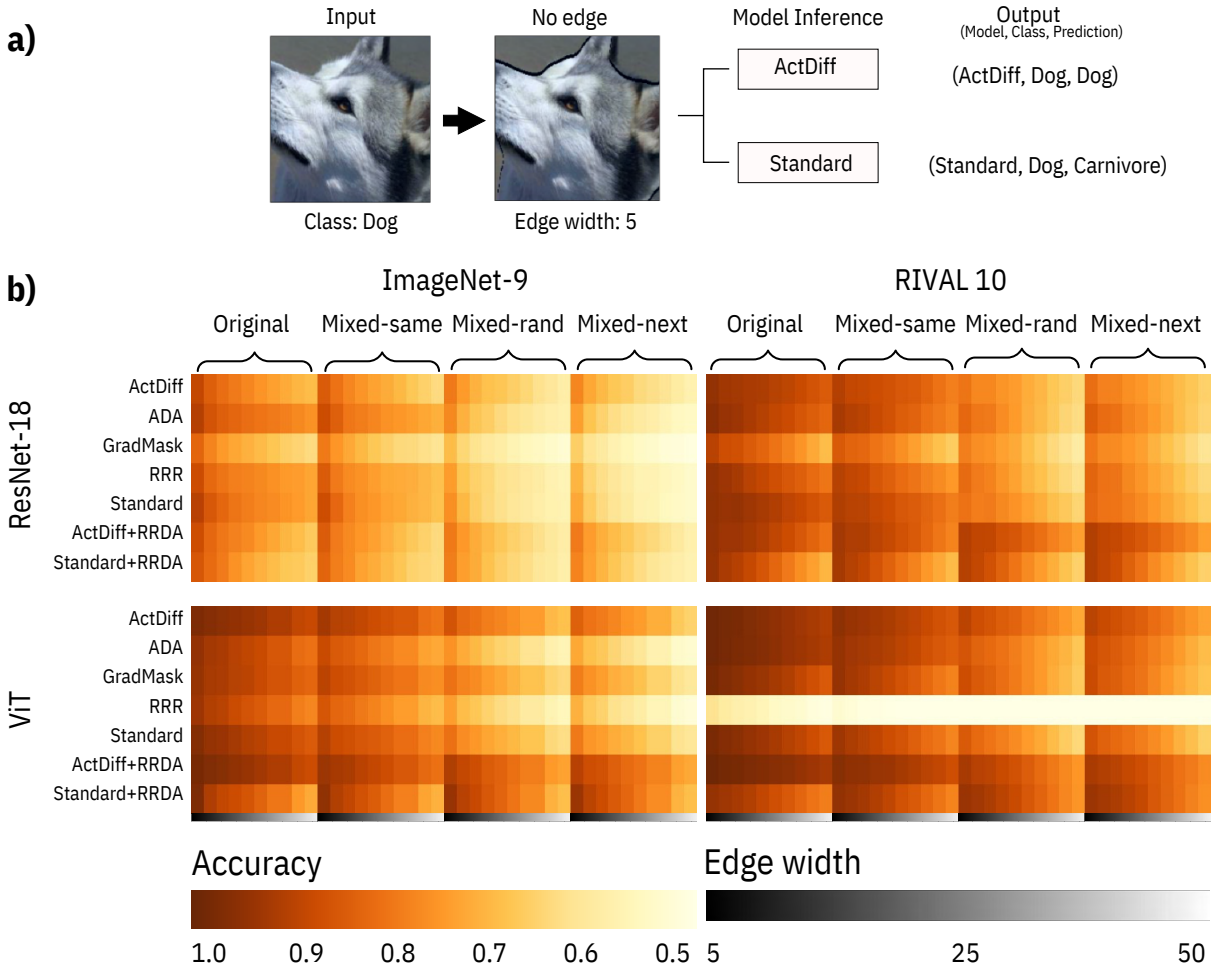

Figure 17 – **a) Pipeline of Edge Analysis.** Given an input image of class y, we first conduct an edge analysis that erases the image edge of width W. We then compute the model's inference to produce a triplet consisting of the model name, class, and predicted class. This edge analysis is computed for all seven models, using an edge width ranging from 5 to 50 pixels. **b) Results Obtained from Edge Dependence Analysis.** The results are grouped by dataset, namely ImageNet9 and Rival10. For each dataset, each column represents a different challenge arranged in a sequence of increasing difficulty, starting with the original data and ending with the original data whose background is from the next class. Within each column, each cell represents the accuracy obtained for a specific edge size, starting from 5 and ending at 50.

50. It is important to highlight that images with high-edge sizes almost do not have information, but these scenarios are important to visualize the tendency of the results. The results of this analysis are presented in Figure 17b). These results indicate that edges are essential for all models across all challenges, as an increase in edge size corresponds to a decrease in accuracy. Another notable observation is the relationship between challenge difficulty and edge dependency. As the difficulty of the challenge increases, the models become more dependent on edge information, as indicated by lower accuracy scores. For instance, focusing on an edge size of five, accuracy decreases in line with the difficulty level of the challenge.

A significant finding is that models using the RRDA augmentation method exhibit greater edge information robustness than the standard and raw RRR methods. While the standard method with RRDA maintains similar performance across all challenges, the raw standard method demonstrates greater robustness when evaluated on the original challenge. This suggests that the standard method is tailored to the original distribution and depends on the background. In general, ActDiff with RRDA augmentation outperformed other methods, demonstrating consistent accuracy across all challenge variations.

Table 8 – **Signal Information Results.** Comparison of model performance when trained with images containing only foreground (FG) or background (BG) information.

Arch.	Method	ImageNet-9		RIVAL10	
		Only FG	Only BG	Only FG	Only BG
RN-18	Standard	85.01	32.52	89.50	41.30
RN-18	ActDiff	86.96	16.20	91.01	40.50
RN-18	GradMask	77.24	23.98	85.41	32.50
RN-18	ADA	86.72	31.78	91.01	41.32
RN-18	RRR	86.10	30.32	88.47	39.69
RN-18	ActDiff + RRDA	85.43	20.79	94.42	29.99
RN-18	Standard + RRDA	85.14	22.37	93.97	24.35
ViT	Standard	91.80	42.35	91.43	42.36
ViT	ActDiff	95.31	41.04	95.04	44.91
ViT	GradMask	90.57	33.63	92.00	45.03
ViT	ADA	87.70	36.35	93.95	47.94
ViT	RRR	86.12	33.24	37.10	27.87
ViT	Standard + RRDA	**97.30**	32.74	**96.94**	16.35
ViT	ActDiff + RRDA	95.68	44.17	95.65	48.58

4.2.6 Models Dependence on Signal Information

Definitions 4.1.1 and 4.1.2 clarify what we consider as class and context informative features. In the context of an image classification task, with humans acting as a fair oracle, the object signal is sufficient for us to perform the classification. This section analyzes model accuracy when presented with only object signals or background information. Table 8 presents the results.

The results demonstrate that high test set accuracy does not necessarily translate to high accuracy when faced with only foreground information. All models trained without RRDA experience a decrease in accuracy of at least 5% (i.e., Orig. - Only FG accuracy). However, the ViT model trained with Standard + RRDA is the least affected, achieving almost the same accuracy with Only FG as with the Original test, with this difference being less than 1% on the RIVAL10 dataset.

Comparing the results of all ViT models with those of ResNet models trained with the same method, it is evident that ViT consistently achieves higher Only FG accuracy. This reinforces the claim that the choice of architecture is a fundamental building block in achieving robust models.

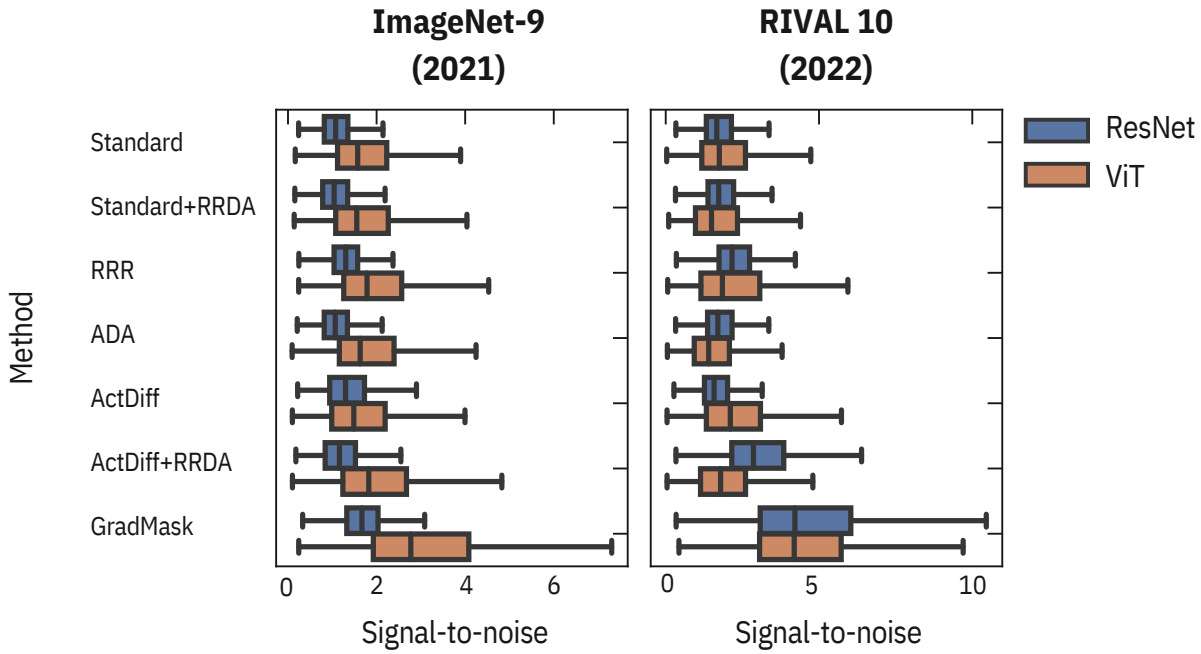

Figure 18 – **Analysis of Signal-to-Noise Ratio for Saliency.** For each model and dataset, we compute the signal-to-noise ratio for each image using the Saliency interpretability method. We then create a box plot to display the distribution of these ratios. The left panel presents the signal-to-noise ratio distributions for the model trained with IN-9, while the right panel illustrates the scenario with RIVAL 10.

4.2.7 Interpretability Methods are Fragile

Interpretability methods generate an attribution matrix, where each input dimension indicates the importance of the corresponding input feature dimension for the model's output prediction. These methods enable us to analyze the difference in feature attribution between a model robust to background changes and one that is not. To carry out this analysis, we compute the signal-to-noise ratio (i.e., the ratio between the mean importance of the signal and background) for each input image in each model and construct a box plot to analyze the differences

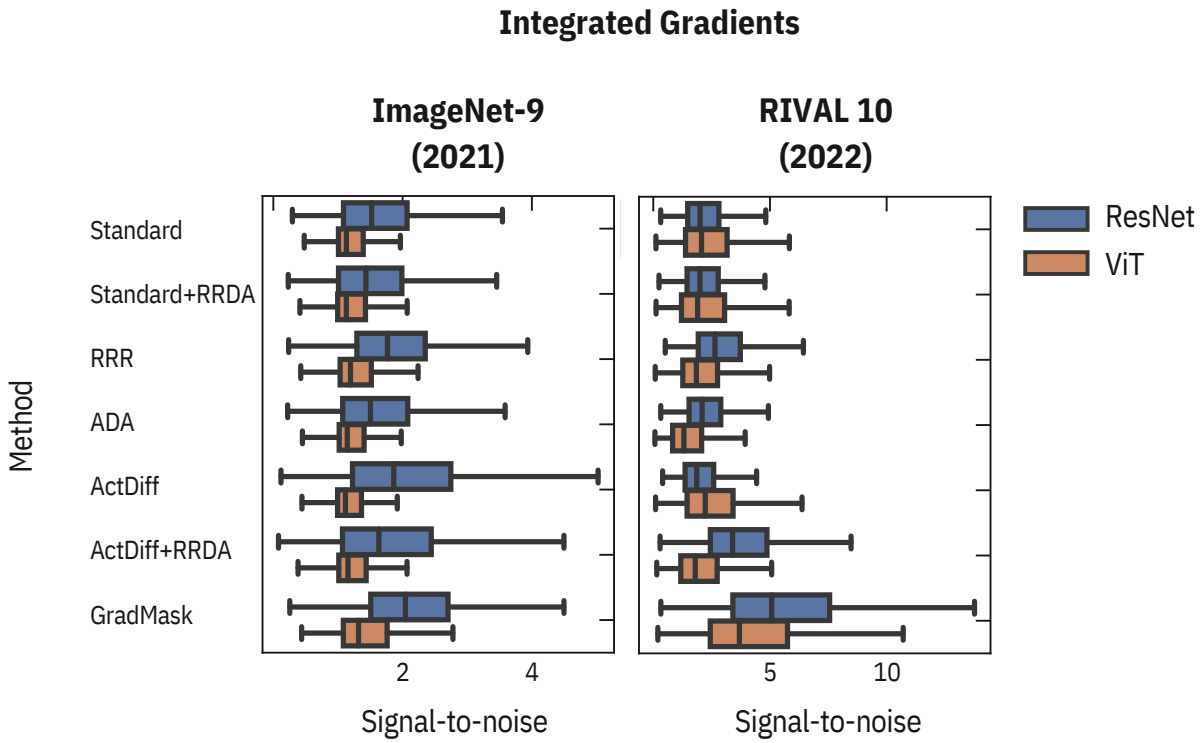

Figure 19 – **Analysis of Signal-to-Noise Ratio for Integrated Gradients.** This pipeline follows the same steps as in Figure 18. However, this scenario uses the Integrated gradients interpretability method instead of Saliency.

between models. Figures 18 and 19 present the results for the Saliency and Integrated gradients interpretability method (SIMON; RODNER; DENZLER, 2014), respectively.

The results show that all ViT models exhibit a higher signal-to-noise ratio than the corresponding ResNet-18 models trained with the same method on IN-9 for the Saliency method. However, this pattern does not hold for the RIVAL 10 dataset, again underscoring that dataset characteristics are crucial in these analyses.

Does high background robustness imply high signal importance? When analyzing robust models that achieve high accuracy on only-FG and all mixed challenges, it might seem natural to expect these models to attribute high importance to the signal and low importance to the background (i.e., have a high signal-to-noise ratio). However, our analysis reveals that this assumption does not always hold. For example, even one of the less robust models, ResNet-18+GradMask, exhibits a high signal-to-noise ratio in RIVAL10. Furthermore, while RRR and GradMask have higher signal-to-noise ratios than standard+RRDA, they are less robust. This suggests that methods that learn to attribute low importance to the background (i.e., those with a high signal-to-noise ratio) are not necessarily the most robust. These counter-intuitive findings warrant further investigation, as they raise several research questions regarding the accuracy of interpretability methods and whether high signal importance is a cause or a consequence of model robustness.

4.3 CONCLUSION

This chapter evaluates methods such as RRR, GradMask, ActDiff, and ADA for their robustness to image background sensitivity using the ImageNet-9 and RIVAL10 datasets. The results indicate that these methods struggle to create a robust model that focuses on signal information rather than context information. In response, we propose the Right Reasons Data Augmentation (RRDA) method to guide the training process and create robust models that prioritize signal over context information. Remarkably, our results show that RRDA improves the model performance upon the standard and ActDiff outcomes.

The vulnerability of RRR, GradMask, and ADA to background sensitivity is intriguing. To obtain deeper insights, we conducted an interpretability analysis to understand how these models attribute importance to different features. We computed the signal-to-noise ratio to quantify the importance of the relationship between signal and context. The results from this analysis, along with the challenges presented by the ImageNet-9 and RIVAL10 datasets, suggest that having a high signal-to-noise ratio (i.e., signal features having high importance) is not necessarily an indicator of model robustness. This helps clarify why RRR and GradMask did not improve in terms of background sensitivity. Besides, this raises questions about the fairness of interpretability methods.

5 ADVERSARIAL RIGHT FOR THE RIGHT REASONS

Deep neural networks are robust to classify complex and high-dimensional data accurately. These models automatically select important *features* or learn new data representations based on what was learned during the training step. Thus, there is no guarantee that the input vectors with correct information (domain expert discriminatory information) guide the decision-making process. The *Right for the right reasons* (RRR) methods try to mitigate this problem by directing the models during training and making them use important information when performing the inference. Following other research directions, (ZHANG; ZHU, 2019) evaluated how Adversarial Trained Convolutional Neural Networks (ATCNN) process the input information and showed that it learn to extract features related to the structure of the objects and are less biased to object texture than standard models. These properties from ATCNN models are important for a fair and robust model. Therefore, we ask, may adversarial training improve the RRR methods? [1] Before answering this question, we perform an analysis with a Toy problem to compare the interpretability maps of adversarial trained models with standard and RRR trained models.

Toy problem motivation

This setup uses a toy problem described in (ROSS; HUGHES; DOSHI-VELEZ, 2017). It comprises a two-class color dataset, as in figure 20. The first class is composed of images whose corners have the same color, and the three top-middle blocks have different colors. The images from class 2 are the ones in which none of the two class-1 conditions are satisfied.

The evaluation setup consists of training a Multilayer-Perceptron (MLP) architecture with three different approaches: (i) standard, (ii) adversarial training, and (iii) right for the right reasons. The MLP model comprises two hidden layers, the first with 50 units and the last with 30 units. Both layers use the ReLU activation function. After training, we compute the test accuracy and adversarial robustness accuracy and perform a qualitative analysis of the interpretability of each model decision. Thus, we can compare how the models give importance to each input.

Table 9 presents the results obtained from this analysis. All models have competitive accuracy on the test set, and the model trained using the adversarial approach has better accuracy on the adversarial evaluation scenario, as expected.

From Figure 21, we can see the qualitative analysis of the interpretability obtained from each trained model. This analysis shows that the model trained with the standard approach uses

[1] This chapter is based on Santos, Flávio Arthur O., Maynara Donato de Souza, and Cleber Zanchettin. "Towards Background and Foreground Color Robustness with Adversarial Right for the Right Reasons." In International Conference on Artificial Neural Networks (ICANN 2023), pp. 169-180. Cham: Springer Nature Switzerland, 2023. URL <https://link.springer.com/chapter/10.1007/978-3-031-44192-9_14>

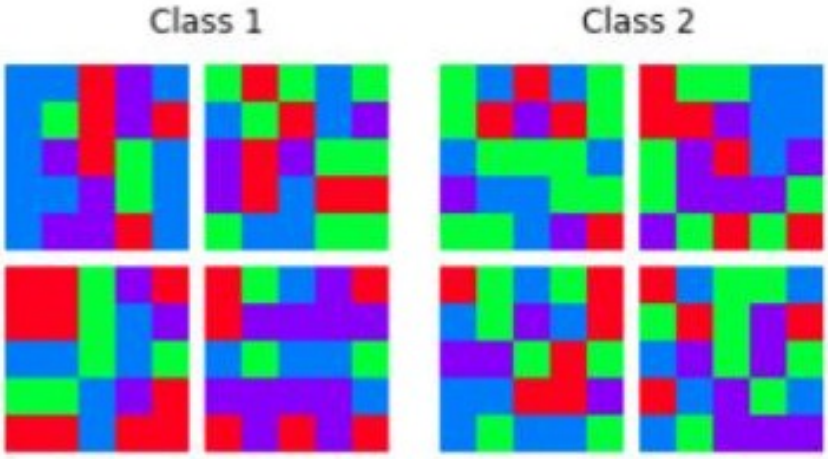

Figure 20 – **Input examples of the Toy problem dataset.** The toy problem has two different classes defined by two well-defined rules. Thus, we can use it to evaluate if the model inference uses features related to the rules.

Table 9 – Results of the toy problem analysis.

Model	Test Accuracy	Adversarial Accuracy
Standard Training	99.0	45.0
Adv. Training	99.0	92.0
RRR	100.0	87.0

many unimportant features, while the models trained with adversarial and RRR approaches use the important ones. These results positively strengthen our question, indicating that adversarial training can help the model to be right for the right reasons. Therefore, we propose a new method to incorporate adversarial samples in the RRR methods, which we named Adversarial Right for the Right Reasons (ARRR).

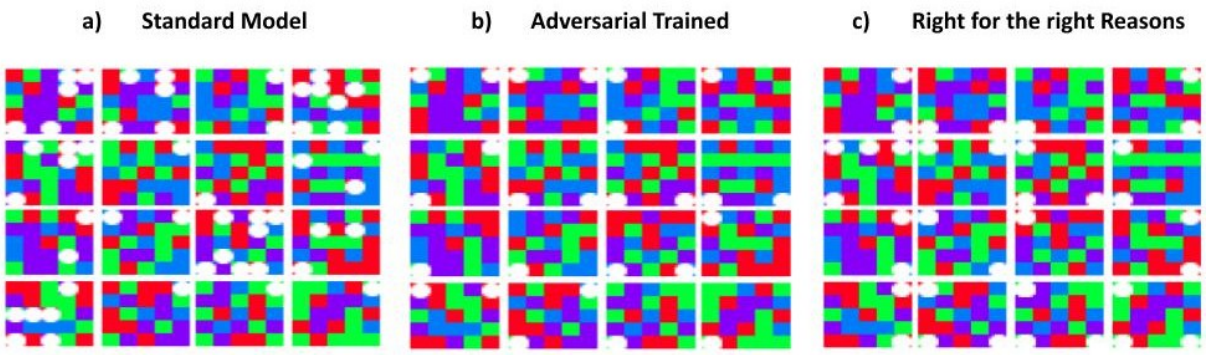

Figure 21 – Qualitative analysis of the input interpretability of all trained models. Figures a, b, and c represent the interpretability obtained from the standard training, adversarial training, and RRR model, respectively. The white dot highlights the most important features of each model inference.

5.1 ADVERSARIAL RIGHT FOR THE RIGHT REASONS

We propose introducing adversarial training in the right for the right reasons methods, generating a new approach we named Adversarial Right for Right Reasons (ARRR). Figure 22 presents the ARRR method pipeline.

$$Loss(X, y, I, rr) = \lambda_1 \underbrace{L_{pe}(f_\theta(X), y)}_{Prediction\ error} + \lambda_2 \underbrace{L_{rrr}(I(f_\theta(X)), rr)}_{RRR\ error} \tag{5.1}$$

The equation 5.1 represents the general structure of the right for the right reasons loss functions. The vector X represents the input vector, y the input target, I the interpretability method, and rr the right reasons. It is composed of two loss functions, the first one (L_{pe}) to compute the prediction error and the second one (L_{rrr}) to calculate how the model is giving importance to the right reasons features.

In order to introduce adversarial training in this loss function, we propose to replace the input vector X with its adversarial attack, named X_{atk}. Thus, the adversarial right for the right reasons loss equations results in the equation 5.2.

$$Loss(X_{atk}, y, I, rr) = \lambda_1 \underbrace{L_{pe}(f_\theta(X_{atk}), y)}_{Prediction\ error} + \lambda_2 \underbrace{L_{rrr}(I(f_\theta(X_{atk})), rr)}_{RRR\ error} \tag{5.2}$$

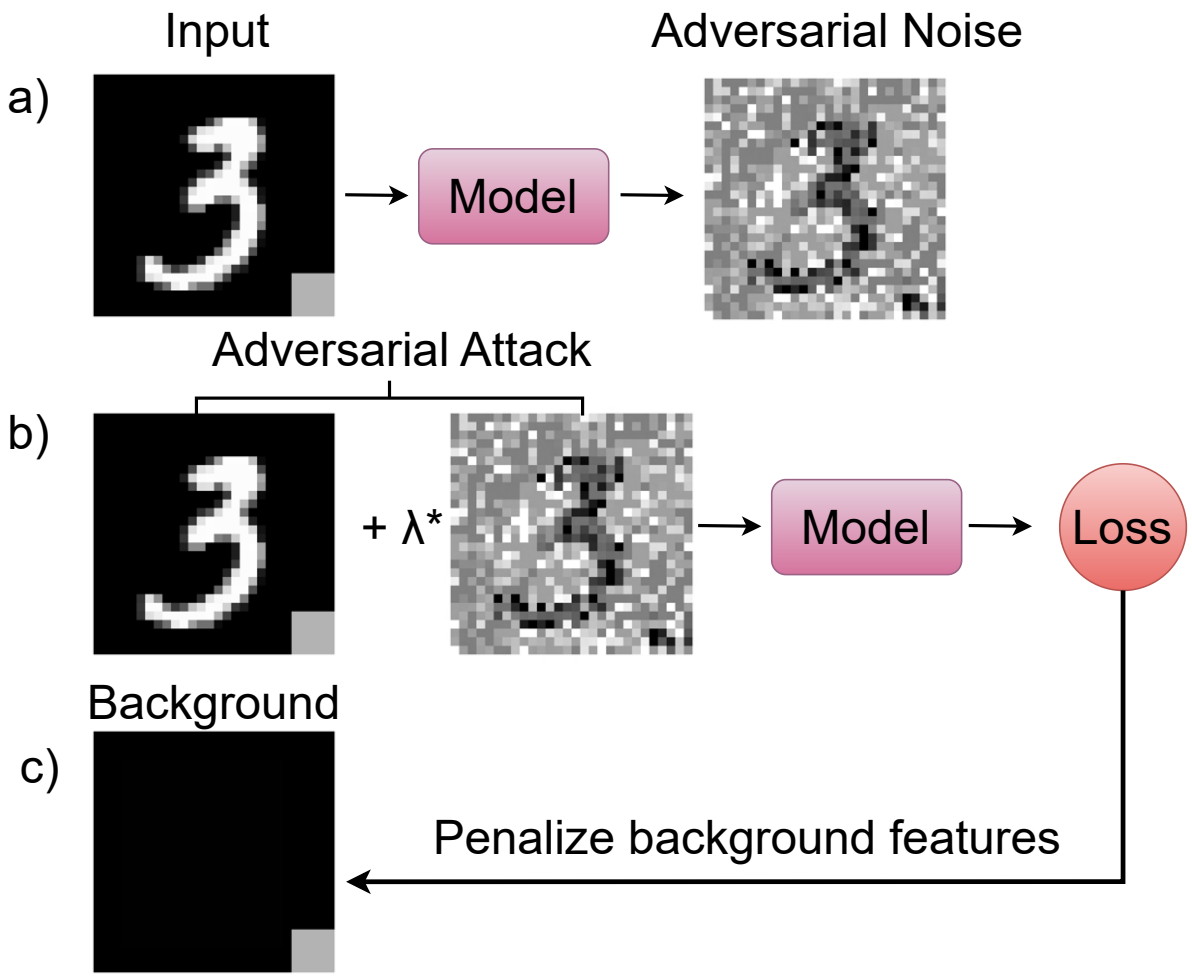

Figure 22 – **Steps of adversarial right for the right reasons method.** The ARRR method comprises three stages, presented in parts a), b), and c), respectively. Given an input image, first, it computes an adversarial noise using some adversarial attack approach. Second, it uses this adversarial noise to generate an adversarial attack and feed the model to compute the inference loss. From the loss, ARRR uses RRR methods to penalize the importance of background pixels, thus learning to focus on signal information.

5.2 EXPERIMENTS AND RESULTS

We rely on the experiments suggested by (RIEGER et al., 2020) to evaluate the proposed method. In this section, they have been grouped into two categories: (i) Structure-based problems, in which the model must be able to learn the structure of objects, and (ii) Texture-based problems, in which the extraction of features related to textures is critical for accurate classification. As we want to analyze whether the model extracts the *features* that are important to solve the problem, all the data sets used have information that is not important for the problem in question and is biased. Thus, they must learn to ignore those unimportant and biased features to have good results on the test set.

Figure 23 – **Examples of the Decoy MNIST and Color MNIST datasets.** Both datasets are built from MNIST samples and have ten classes. Besides, they have a bias in the training input information. Thus, if the model learns the shortcut to minimize the training loss, it will have low accuracy in the test set. Decoy MNIST has a gray patch class-indicative in the training samples, while the Color MNIST has a color indicative. However, these biases are different in the test set.

5.2.1 Structure-based

We have used two toy datasets based on MNIST to perform this analysis: (1) Color MNIST and (2) Decoy MNIST. Like MNIST, both databases comprise ten classes, 60,000 training, and 10,000 test images. The difference between them and the original MNIST lies in the information contained in each one. The first dataset has a color-class indicative in the training set, but the colors-class is different in the test set; thus, if the model learns to identify the color instead of the shape, it will have poor results in the test set. The second dataset, Decoy MNIST, has a gray-scale patch in a random image position, indicating the class. Thus, if the model learns to identify the patch instead of the object to classify shape (in this dataset, the object is a number from 0-9), it will achieve poor results. To evaluate the impact of the adversarial training in these models, we trained a simple CNN model with the FGSM attack and used it in combination with RRR, CDEP, and EG approach. It is important to highlight that image processing functions can solve both datasets' biases. However, it can be used to evaluate if the deep learning models can ignore such biases easily identified by humans.

From the results presented in Table 10, we can see that the adversarial training helps to improve the accuracy of all models, and even when we train the model only with the

FGSM, it achieves better accuracy than Vanilla, RRR, and EG training. In both tasks, Decoy Mnist and Color Mnist, the model should be capable of identifying the number structure (class information) to classify it correctly, so these results confirm the findings in (ZHANG; ZHU, 2019) and indicate that adversarial training can help the model to be right for the right reasons in structure-based problems.

Table 10 – Results of the structure-based problem.

Model	ColorMNIST	DecoyMNIST
Vanilla	0.2	60.1
CDEP	31.0	97.2
RRR	0.2	99.0
EG	10.0	97.8
FGSM	20.51	**99.70**
RRR + FGSM	19.90	**99.70**
EG + FGSM	11.35	98.90
CDEP + FGSM	**46.27**	99.55

5.2.2 Texture-based

In image classification problems, texture is an important signal information, and some tasks can be categorized as texture-based because the pattern to be extracted and identified is texture. This section evaluates the connection between adversarial training and the right for the right reasons considering a texture-based task. We use the ISIC Skin Cancer dataset (CODELLA et al., 2018), a benchmark comprising 21,654 images. Its task is to classify skin lesions as benign or malignant. However, the dataset has a bias in half of the benign images. The bias is a color patch present only in benign images; thus, if the model identifies that color patch, it can classify half of the benign lesions without knowing any pattern about the malignant or benign lesion. Therefore, it is an important benchmark to evaluate whether the model learns to identify the right or biased pattern. Figure 24 presents some samples from the ISIC dataset.

Table 11 presents the results obtained from this analysis. We disagreed with the experiments performed in (RIEGER et al., 2020) on this task. Since the dataset is unbalanced, the batch size parameter is important. Due to computational constraints, they used different batch sizes for each right for the right reasons. In this work, we re-implemented the RRR and the CDEP model with a ResNet-18 instead of a VGG to keep the batch size equal for every experiment. So, the first three rows of the table 11 represent the results obtained in the (RIEGER et al., 2020) work, and the last one is from our execution. Our implementation of the CDEP and RRR methods did not present any weakness because their results are better than those in (RIEGER et al., 2020).

In general, the adversarial training did not improve the evaluation metrics in this scenario, except when we compare the AUC metric on the CDEP with and without adversarial training.

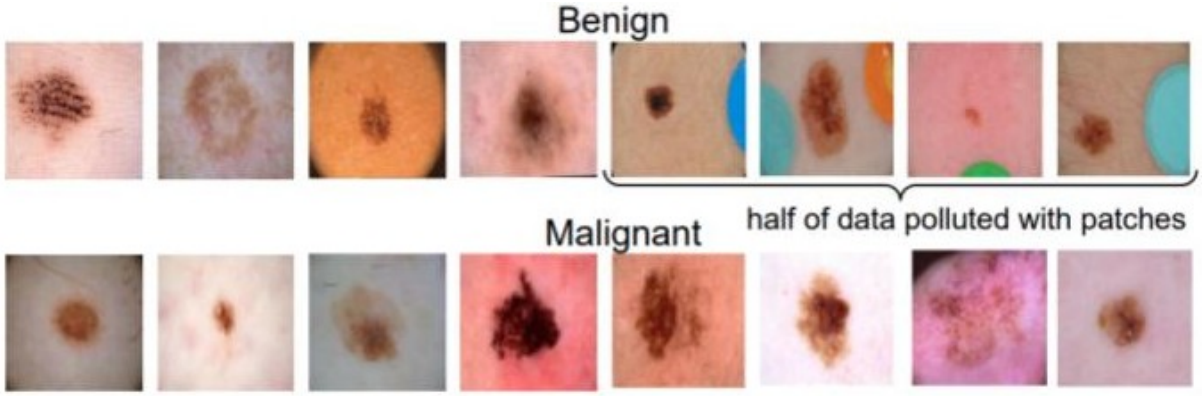

Figure 24 – **Examples of the ISIC dataset.**

Table 11 – Results of the texture-based problem.

Model	AUC (NO PATCHES)	F1 (NO PATCHES)	AUC (ALL)	F1 (ALL)
Vanilla (RIEGER et al., 2020)	0.87	0.56	0.93	0.56
RRR (RIEGER et al., 2020)	0.75	0.46	0.86	0.44
CDEP (RIEGER et al., 2020)	0.89	0.61	0.94	0.60
Vanilla	0.91	0.69	0.94	0.69
RRR	0.91	0.67	0.95	0.67
CDEP	0.88	0.62	0.91	0.62
FGSM	0.82	0.52	0.90	0.52
RRR + FGSM	0.89	0.67	0.94	0.67
CDEP + FGSM	0.89	0.64	0.94	0.64

This result indicates that adversarial training can not help the model be right for the right reasons on texture-based image classification tasks. Our intuition about these results is that the inputs generated by adversarial methods change the texture of the input image, which is very important for this task. Thus, this can change the class texture pattern, making it challenging for the model to learn this information.

5.3 CONCLUSION

In this chapter, we proposed the adversarial rights for the right reasons method (ARRR) to combine two important properties that deep neural networks must have: 1) adversarial robustness and 2) right for the right reasons. Our assumption is combining adversarial samples and right for the right reasons constraints can boost the model's robustness. We evaluate the proposed approach with two categories of image classification problems: structure-based and texture-based. The findings indicate that introducing adversarial training on the RRR loss helps the model robustness on the structure-based problem and achieves competitive results

on texture-based problems.

6 BACKGROUND DEPENDENCE OF VISUAL LANGUAGE MODELS

Deep learning models usually require a large dataset to achieve satisfactory generalization performance on image recognition tasks (KOLESNIKOV et al., 2020; KRIZHEVSKY; SUTSKEVER; HINTON, 2012). However, recent advancements in Visual Language Models (VLMs) (RADFORD et al., 2021) have enabled applications to perform zero-shot image classification with a surprising performance by simply querying a pre-trained model. These VLMs models are trained with millions of image-text pairs and optimized to align the similarity between both image representation and text representation, thus learning to correlate the visual with text information.

Even with its impressive achievements, it's essential to comprehend its constraints, just as we do with other deep learning models. For instance, we know that non-VLM models may learn spurious correlations between data and labels (TIAN et al., 2022), they are susceptible to adversarial attacks (GOODFELLOW; SHLENS; SZEGEDY, 2015), and they do not generalize to out-of-distribution samples (YANG et al., 2022). While these limitations have been extensively studied in standard-trained image recognition models, there is a lack of studies concerning the zero-shot image recognition models based on VLMs.

Previous work has studied the correlation between image backgrounds and image labels (XIAO et al., 2021). It demonstrated that image recognition models are biased towards this contextual information, and even human vision may rely on object context (TORRALBA, 2003). The same literature suggests that the problem relies on the fact that the background variation is much more common and diverse in the real world than training datasets. Thus, as LLMs and VLMs are trained using large datasets, it is crucial to analyze if it may reduce the problem due to the diversity of data.

This chapter[1] aims to comprehensively evaluate and present insights into how those zero-shot image classifiers based on VLMs utilize the image background information and whether they are susceptible to such biases. Our efforts range from an interpretability analysis aiming to understand how it attributes similarity scores in these situations to computing and comparing the similarity scores corresponding to each target category description for each image and its variations from the background shift.

The evaluation protocol comprises the ChatGPT+CLIP, ChatGPT+ALIGN (JIA et al., 2021b), CLIP, and ALIGN models (MENON; VONDRICK, 2023) as zero-shot image classification and compares their performance with standard architectures such as Vision Transformer (ViT) (DOSOVITSKIY et al., 2020) and ResNet (HE et al., 2015). The experiments are performed on ImageNet-9 (XIAO et al., 2021) and RIVAL10 Background (Developed in Chapter 4) datasets,

[1] This chapter is based on Santos, Flávio Arthur Oliveira, Maynara Donato de Souza, and Cleber Zanchettin. "Evaluating zero-shot image classification based on visual language model with relation to background shift." In Neural Information Processing Systems Conference: LatinX in AI (LXAI) Research Workshop 2023, New Orleans, USA, 2023.

which are benchmarks used for robustness evaluation. Our accomplishments contribute to the lack of analysis in this context, allowing a deeper understanding of the zero-shot LLM-based model's behavior and how it responds to the challenges.

6.1 METHODS

Zero-shot image classifier

We employed two distinct approaches for zero-shot image classification to conduct our analysis. The first leverages the VLM, computing the similarity score between the label text and the input image. The second approach involves the fusion of ChatGPT with VLM, referred to as Large Language Model + VLM (LLM+VLM) (MENON; VONDRICK, 2023). In the case of image classification using LLM+VLM, the model initially acquires category descriptions from the LLM. Next, it follows the computations as in Equations 6.1 and 6.2. Specifically, it calculates the average of similarities (ϕ) between the image embedding x and the descriptor text embedding d for each descriptor in a set of descriptors $D(c)$ associated with a fixed class c. Finally, it determines the image category based on the highest class score within the set of classes C, as present in equations 6.2.

$$s(c, x) = \frac{1}{|D(c)|} \sum_{d \in D(c)} \phi(d, x) \tag{6.1}$$

$$P(x) = \underset{c \in C}{\operatorname{argmax}}\, s(x, c) \tag{6.2}$$

We performed our analysis with two different VLMs, namely ALIGN (JIA et al., 2021b) and CLIP (RADFORD et al., 2021). ALIGN[2] (JIA et al., 2021b) is a visual language model which uses contrastive learning train to align image representation with text representation, it uses an EfficientNet (TAN; LE, 2019) to encoder image information and BERT (DEVLIN et al., 2018) as text encoder. Similarly, CLIP (RADFORD et al., 2021) is also a VLM that learns to produce similar representations for a given image and text pair. However, the CLIP uses the ViT (DOSOVITSKIY et al., 2020) as an image encoder and a causal language model as a text encoder.

Background shift evaluation protocol

We propose a comprehensive analysis to evaluate the limitations of zero-shot image classifiers based on LLM+VLM concerning image background shifts. The key questions for this analysis are as follows:

- **Q1:** What is the impact of the background shifts on VLM image classifiers?

[2] As the original align model is not available, we used the version from kakaobrain/align-base released in huggingface

- **Q2:** How does the VLM image classifier behave when dealing with images containing only signal information?
- **Q3:** How does the VLM image classifier behave with images containing only background information?
- **Q4:** What is the similarity score distribution for images with different backgrounds, and how does it relate to the model performance?

Questions 1, 2, and 3 will be answered based on the mixed-same, mixed-rand, mixed-next, only-fg, and only-bg challenges. To address question 4, we will process all images from mixed protocol and analyze how the zero-shot classifier attributes similarity scores, providing insights about the classifier's similarity attributions.

6.2 RESULTS

This section presents the experimental results and addresses the questions we have formulated. Our aim is not to create the most background robust model but to assess the zero-shot image classifier based on VLM and LLM+VLM, thereby contributing to a better understanding of its limitations and offering insights for future research. In chapter 4, we introduced the Right Reasons Data Augmentation (RRDA) method, aiming to enhance model fairness by improving model robustness against background shifts. Therefore, we have used their results as a baseline for assessing how a zero-shot image classifier handles background shifts.

6.2.1 Background sensitivity results

The mixed-same protocol uses a dataset composed of original signals but with backgrounds from a random image within the same category. Thus, if there is a high correlation between backgrounds that generally appear in the same object category and the category label, the model will also achieve high accuracy on this task. On the other hand, achieving high accuracy on mixed-rand and mixed-next means that the model is robust to background shifts and indicates that it classifies according to signal information instead of only background or considering the background. In addition to these analyses, we also compute the BG-Gap metric, which is the difference between mixed-same and mixed-rand accuracy, measuring how much the model correlates class background with the target category. Table 12 presents our results for this evaluation.

Impact of Background Challenge. The results confirm that all trained models exhibit limitations when faced with background shifts. As the BG-Gap is larger than 0 for all methods, it is evident that dealing with background challenges remains difficult in image classification, even for VLM-based models, which are trained with millions of (image, text) pairs. However, the ALIGN model achieved the lowest BG-Gap between the VLM approaches on the ImageNet-9 dataset and the lowest between all methods on the RIVAL10 dataset. This is surprising as

Table 12 – **Background challenge results.** Each row represents a challenge evaluation protocol with the architecture, training method, and dataset in columns Architecture, Method, and Dataset, respectively. The mixed same, mixed rand, and mixed next columns represent the accuracy results for the respective background challenge, while the BG-Gap means the difference between Mixed rand and Mixed same.

Architecture	Method	Dataset	Mixed same	Mixed rand	Mixed next	BG Gap	Original
ResNet-18	Standard	ImageNet-9	92.67	82.99	80.22	9.68	96.15
ResNet-18	ActDiff	ImageNet-9	90.27	84.47	83.26	5.80	93.46
ResNet-18	GradMask	ImageNet-9	86.77	76.34	73.43	10.42	90.79
ResNet-18	ADA	ImageNet-9	92.20	81.80	79.28	10.40	96.05
ResNet-18	RRR	ImageNet-9	91.90	82.12	78.77	9.78	95.31
ViT	Standard	ImageNet-9	94.15	86.84	84.69	7.3	98.35
ViT	ActDiff	ImageNet-9	95.98	90.27	89.46	5.7	98.99
ViT	GradMask	ImageNet-9	93.38	86.52	84.77	6.7	97.04
ViT	ADA	ImageNet-9	91.73	81.98	80.12	9.7	97.24
ViT	RRR	ImageNet-9	90.42	80.04	78.54	10.4	96.74
CLIP	Top-1	ImageNet-9	86.44	78.72	77.24	7.72	92.59
ALIGN	Top-1	ImageNet-9	85.75	79.95	77.35	**5.79**	91.70
ChatGPT+CLIP	Top-1	ImageNet-9	**89.36**	**80.89**	**79.21**	8.47	**94.03**
ChatGPT+ALIGN	Top-1	ImageNet-9	87.21	79.53	78.32	7.68	92.05
ResNet-18	Standard	RIVAL10	95.01	87.82	88.65	7.19	99.19
ResNet-18	ActDiff	RIVAL10	94.91	86.55	87.16	8.36	98.77
ResNet-18	GradMask	RIVAL10	90.65	83.96	84.34	6.69	96.61
ResNet-18	ADA	RIVAL10	95.20	88.64	89.35	6.55	99.07
ResNet-18	RRR	RIVAL10	94.82	87.89	88.67	6.92	98.90
ViT	Standard	RIVAL10	95.31	87.99	88.61	7.32	99.24
ViT	ActDiff	RIVAL10	96.92	92.26	91.47	4.65	99.62
ViT	GradMask	RIVAL10	96.52	90.81	91.09	5.71	99.49
ViT	ADA	RIVAL10	96.27	88.84	90.09	7.45	99.69
ViT	RRR	RIVAL10	53.01	34.09	35.19	18.94	64.76
CLIP	Top-1	RIVAL10	94.02	89.42	89.03	4.60	97.33
ALIGN	Top-1	RIVAL10	**96.15**	**93.30**	**92.66**	**2.85**	**98.68**
ChatGPT+CLIP	Top-1	RIVAL10	94.06	89.86	89.71	4.20	97.96
ChatGPT+ALIGN	Top-1	RIVAL10	94.53	91.22	91.22	3.31	97.13

Table 13 – **Individual foreground and background challenge results.** The results are presented individually for both datasets. The columns Only FG and Only BG mean the model accuracy results when facing images with only foreground and background information, respectively.

Architecture	Method	ImageNet-9		RIVAL10	
		Only FG	Only BG	Only FG	Only BG
ResNet-18	Standard	85.01	32.52	89.50	41.30
ResNet-18	ActDiff	86.96	16.20	91.01	40.50
ResNet-18	GradMask	77.24	23.98	85.41	32.50
ResNet-18	ADA	86.72	31.78	91.01	41.32
ResNet-18	RRR	86.10	30.32	88.47	39.69
ViT	Standard	91.80	42.35	91.43	42.36
ViT	ActDiff	95.31	41.04	95.04	44.91
ViT	GradMask	90.57	33.63	92.00	45.03
ViT	ADA	87.70	36.35	93.95	47.94
ViT	RRR	86.12	33.24	37.10	27.87
CLIP	Top-1	84.62	32.67	90.71	39.63
Align	Top-1	**87.24**	25.78	**95.95**	32.89
ChatGPT+CLIP	Top-1	86.99	**38.32**	91.88	**39.61**
ChatGPT+Align	Top-1	86.94	27.53	93.40	31.27

it is a zero-shot model, and other methods such as ActDiff (VIVIANO et al., 2021), GradMask (SIMPSON et al., 2019), ADA (SANTOS et al., 2019), and RRR (ROSS; HUGHES; DOSHI-VELEZ, 2017) use the object foreground mask to ignores the background information.

The only difference between CLIP and ChatGPT+CLIP, and between ALIGN and ChatGPT+ALIGN, is that the models with ChatGPT use the object feature descriptions in the prompt, while the CLIP and ALIGN use only the label name. Thus, does the object feature description help improve the background robustness? The answer is no, as ChatGPT-based models have a larger BG-Gap than its relative on ImageNet-9, and on RIVAL10, it only enhances the CLIP model by a tiny margin.

6.2.2 Signal and background analysis

The background sensitivity evaluation showed that the ALIGN model is more background robust than the other methods on the RIVAL10 dataset - including standard models. In this section, we evaluate whether zero-shot models can classify images accurately, considering only the signal information and not considering the background features. In this context, we force the model to make decisions based solely on signal features. In addition, we also evaluate

whether these models may classify only the background information as the object target to verify its accuracy. Table 13 presents the results of this analysis.

Considering only foreground information. The ALIGN model performs best on RIVAL10 when assuming only the signal information, achieving the highest accuracy among all evaluated models. It experiences a minor decrease of only 2.73% compared to its original test set accuracy. In contrast, both the ChatGPT+CLIP and individual CLIP models show a more significant drop, with approximately a 7% decrease in accuracy in the original test set. This decline is at least twice that of the best-performing model. It suggests that the classification-by-description approach used in ChatGPT+CLIP and the CLIP models may not perform foreground-only classification as accurately as the original image classification.

Considering only background information. Evaluating the performance of models using solely background information is challenging as it lacks object-specific information. Thus, we neither expect high nor low accuracy but rather randomness results. Nevertheless, we can analyze it jointly with only foreground information (only-FG). Suppose a model displays low accuracies for both only-BG and only-FG protocols but achieves high accuracy in the original classification task. This suggests that the model has learned to classify based on the co-occurrence of background and foreground information rather than performing separate background or foreground recognition. This insight is valuable in understanding the behavior of models like ChatGPT+CLIP, CLIP, ResNet-18, and ViT trained with Standard, GradMask, ADA, and RRR methods. These models obtained only-FG accuracies close to 85%, low only-BG accuracies, but high original accuracy.

6.2.3 Similarity analysis

The background sensitivity and foreground signal analysis have shown that CLIP, ChatGPT+CLIP, and ChatGPT+ALIGN are less robust to background shifts than ALIGN and make mistakes even when the input contains only signal information. However, the reasons for this behavior still need to be completely understood. Debugging deep learning models is difficult, as it requires various tools (KOKHLIKYAN et al., 2020; SHAH; FERNANDEZ; DRUCKER, 2019), interpretability methods, and specific analysis (ZHANG; ZHU, 2019). Nevertheless, as the classification by description (i.e., ChatGPT+CLIP and ChatGPT+ALIGN) approach has the advantage of being naturally interpretable (it classifies images based on the similarity between the input image x and text category c descriptions $D(c)$), we use their similarity scores to infer how they behave with different challenges protocols (namely original, mixed same, mixed rand, and mixed next). Figure 25a) and 25b) illustrate the pipeline to obtain each feature's similarity scores and the resulting similarity distributions for two categories (In the apprendix we present for all categories.).

Impact of dataset distribution on similarity scores. The results represent the distributions of each description's similarity scores for the different challenges. The similarity scores in the original test set differ significantly from the other challenges protocols, having higher

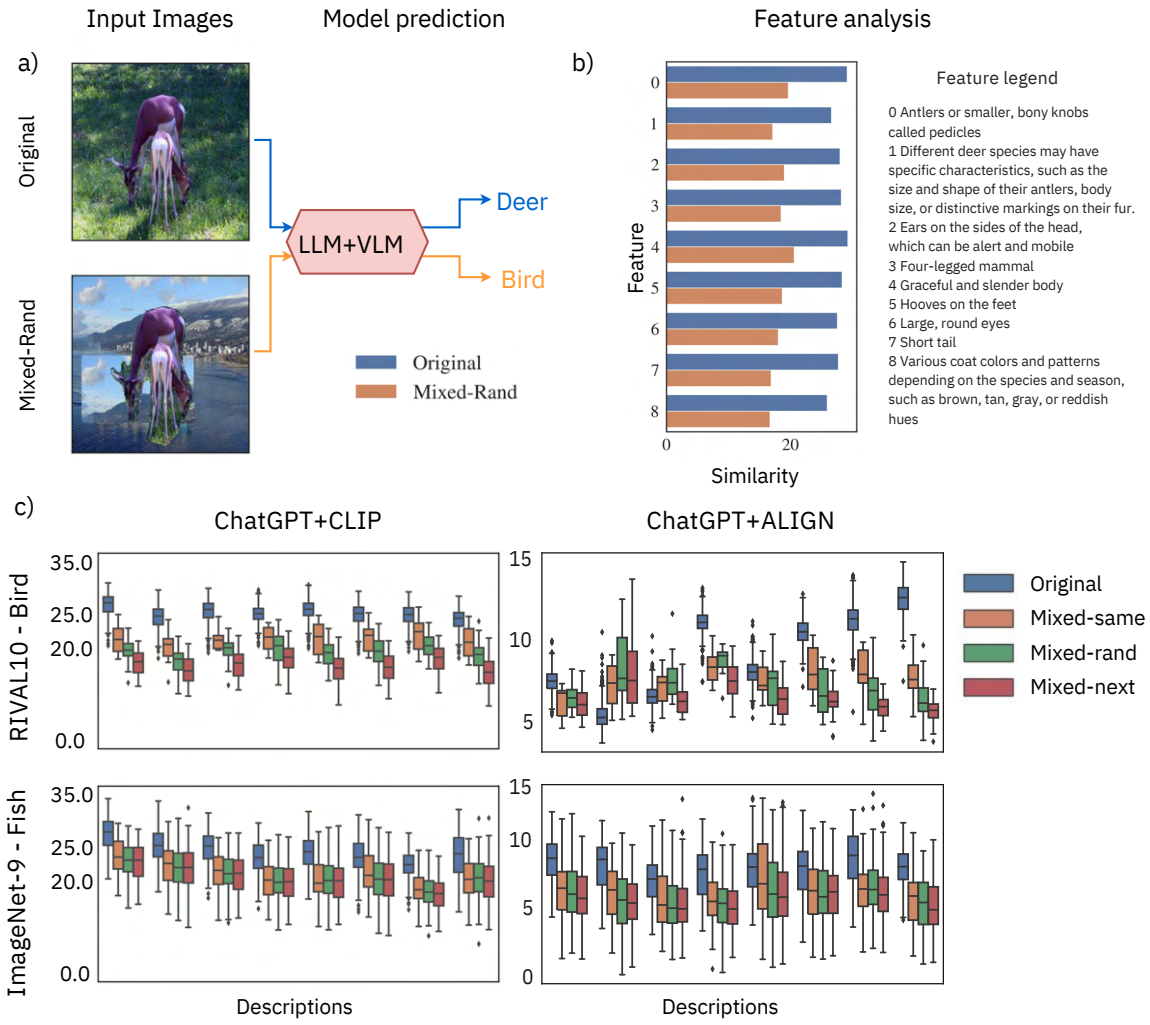

Figure 25 – **Illustration of the distributions of different description similarity scores. a)** Given the original input image and its mixed-same version, we compute the model prediction with the ChatGPT+CLIP model. **b)** After obtaining the model prediction for each image, we extract the similarity scores for each target description and compare their differences. The Feature legend box shows all the descriptions for the Deer category. **c)** For each original image and its version from mixed-same, mixed-rand, and mixed next, we perform the a) and b) pipeline with ChatGPT+CLIP and ChatGPT+ALIGN models. Next, we build a box plot for each challenge and category, thus comparing how the model attributes similarity scores in each challenge. The plot shows the results for the categories Bird from RIVAL10 and Fish ImageNet-9.

scores across all descriptions and categories for ChatGPT+CLIP, but it does not hold for ChatGPT+ALIGN. This result suggests that the ChatGPT+CLIP is mispredicted due to producing low similarity scores for target descriptions, but this does not happen with ChatGPT+ALIGN always.

6.2.4 Description score variability

The LLM+VLM models may exhibit two distinct behaviors when making errors on challenge images: 1) a decrease in the target similarity score, or 2) an increase in the similarity score of some non-target classes. Our similarity analysis revealed that ChatGPT+CLIP consistently yields low target similarity scores on challenge datasets. However, ChatGPT+ALIGN does not consistently exhibit the same pattern, leaving the reasons for its mistakes unclear.

To address this, we introduce the Score Variability metric (SV in equation 6.3) in this section. This metric quantifies how the model alters its similarity scores when predicting challenge images. For a given challenge image ($x_{challenge}$) and category c, the SV function calculates the ratio of changes relative to the equivalent original image similarity score. Consequently, the model may produce a negative ratio score (SV^-) or a positive ratio score (SV^+).

The negative ratio score means that the model is giving a higher similarity score between challenge image and class c than original image and class c, while a positive score means that the model decrease the similarity when faced challenge background. Therefore, if class c is not the target (Pred. column in table 14) and the similarity score is negative, it indicates that the model is recognizing a non-target object into the challenge image.

$$SV(x_{original}, x_{challenge}, c) = \frac{s(c, x_{original}) - s(c, x_{challenge})}{s(c, x_{original})} \tag{6.3}$$

The results from table 14 show that in SV^-, all variations in the target are higher than pred, while on SV^+, all variations from pred. are higher than the target. This implies that when the model decreases its similarity score, it tends to do so more prominently in the target class than in the non-target class. Conversely, when the model increases its similarity score, the elevation is more significant in non-target classes (i.e., predictions). Besides, overall, the ChatGPT+ALIGN has higher SV values than ChatGPT+CLIP, but the results on mixed rand and mixed next with SV^+ indicate that the ChatGPT+ALIGN gives a higher similarity score for the non-target category instead of giving low similarity score for target category.

Table 14 – **Score variability results.** Each row presents the results for a metric evaluated in a challenge and model. The columns Target and Pred. show the score variability (%) in the target and class predicted, respectively.

			ImageNet-9		RIVAL10	
Challenge	Model	Metric	Target	Pred.	Target	Pred.
Mixed same	ChatGPT+CLIP	SV^+	5.93	13.56	5.5	17.11
Mixed same	ChatGPT+ALIGN	SV^+	22.31	1.04	10.5	27.56
Mixed rand	ChatGPT+CLIP	SV^+	5.03	19.23	5.06	22.95
Mixed rand	ChatGPT+ALIGN	SV^+	21.16	77.34	9.46	40.6
Mixed next	ChatGPT+CLIP	SV^+	6.27	19.31	5.66	23.75
Mixed next	ChatGPT+ALIGN	SV^+	19.16	85.29	10.39	41.89
Mixed same	ChatGPT+CLIP	SV^-	10.05	4.4	11.68	3.75
Mixed same	ChatGPT+ALIGN	SV^-	24.07	10.40	20.04	11.79
Mixed rand	ChatGPT+CLIP	SV^-	12.31	4.11	15.26	5.13
Mixed rand	ChatGPT+ALIGN	SV^-	28.71	26.85	24.42	10.78
Mixed next	ChatGPT+CLIP	SV^-	13.45	4.39	14.44	5.05
Mixed next	ChatGPT+ALIGN	SV^-	30.04	11.09	21.97	11.77

6.3 CONCLUSION

This chapter analyzes how zero-shot image classifiers based on visual language models are sensitive to image background changes. We used the ImageNet-9 and RIVAL10 datasets as benchmarks, CLIP, ChatGPT+CLIP, ALIGN, and ChatGPT+ALIGN, as a zero-shot image classifier. Our findings indicate that all tested models have limitations when faced with background shifts. However, the ALIGN model achieved the best results across most metrics, demonstrating its effectiveness in handling the background challenge. In contrast, both the ChatGPT+CLIP and individual CLIP models showed a more significant decline in accuracy, suggesting that it may correlate the background information with the target label.

Additionally, we conducted a similarity scores and a description score variability analysis to understand the reason behind the ChatGPT+CLIP and ChatGPT+ALIGN wrong predictions. The results indicate that the ChatGPT+CLIP and ChatGPT+ALIGN predict wrongly due to different reasons. The ChatGPT+CLIP attributes a low similarity score to the object category when facing a non-target background, while ChatGPT+ALIGN attributes a higher similarity score to the non-target category.

7 MODEL INSPECTOR TOOL

Interpretability methods have become increasingly important with the growth in model complexity and the resulting lack of transparency in the decision-making process. The model transparency and interpretability are usually associated with the degree to which a human can understand the cause of a decision. When making predictions with a neural network, the data input is fed through many multiplication layers with the learned weights and non-linear transformations. A single prediction can involve millions of mathematical operations, thus making it difficult for humans to follow the exact mapping from input data to inference. We would have to consider millions of weights that interact in a complex way to understand a prediction by a neural network. We need specific interpretability methods to interpret the behavior and predictions of neural networks.

In recent years, several methods have been proposed to interpret deep learning model outputs (SIMONYAN; VEDALDI; ZISSERMAN, 2014; SELVARAJU et al., 2017; SUNDARARAJAN; TALY; YAN, 2017; SUDHAKAR et al., 2021). Given an input x, model f, and target category y, these interpretability methods build an attribution map a with the same size as x, where a_i means how much important the feature x_i for $f_y(x)$. There are some libraries developed with Python[1] that we can use to instantiate these methods and interpret models developed in Pytorch (PASZKE et al., 2019) or TensorFlow (ABADI et al., 2016), for example Captum [2], Innvestigate (ALBER et al., 2019), and TensorFlow Interpretability [3]. These libraries have an easy-to-use interface where we can instantiate the interpretability methods to produce the attribution maps for our inputs. Still, we need to codify all input interactions that we want to infer the impact of feature changes in model output or attribution maps, thus being a challenge to beginner or even intermediate users to debug its models.

To mitigate this issue, we developed the Model Inspector[4,5] tool that allows users to manipulate various visual features of an input image to understand better the model's sensitivity to different types of information. Our goal is to provide a more comprehensive framework for model understanding and help researchers and practitioners better understand the strengths and weaknesses of deep learning models in image classification. The Model Inspector also

[1] https://www.python.org/

[2] https://captum.ai/

[3] https://tf-explain.readthedocs.io/en/latest/

[4] This chapter is based on these two works 1) Santos, Flávio AO, Maynara Donato de Souza, Pedro Oliveira, Leonardo Nogueira Matos, Paulo Novais, and Cleber Zanchettin. "Image Classification Understanding with Model Inspector Tool." In International Conference on Hybrid Artificial Intelligence Systems (HAIS 2023), pp. 611-622. Cham: Springer Nature Switzerland, 2023. URL <https://link.springer.com/chapter/10.1007/978-3-031-40725-3_52>. 2) Santos, Flávio Arthur Oliveira, Cleber Zanchettin, José Vitor Santos Silva, Leonardo Nogueira Matos, and Paulo Novais. "A hybrid post hoc interpretability approach for deep neural networks." In International Conference on Hybrid Artificial Intelligence Systems (HAIS 2021), pp. 600-610. Cham: Springer International Publishing, 2021. URL <https://link.springer.com/chapter/10.1007/978-3-030-86271-8_50>

[5] Model inspector tool is available at <https://github.com/faos/image-classifier-model-inspector>

has two novel methods, which we named U Analysis (UA) and Iterative post-hoc attribution (IPHA); U Analysis allows us to evaluate the importance of different patches in an image and understand the impact of removing them on the model's classification performance, while IPHA defines interpretability as a optimization problem and finds which input features mostly contributes to the model decision.

As this chapter presents a tool with two methods, we discuss each of them in a independent section. In the following, we will present the U Analysis in section 7.1, IPHA in section 7.2, and finally the Model inspector in section 7.3.

7.1 U ANALYSIS

Different methods were proposed to visualize features and concepts learned by the neural network models, which have a performance that is less 'interpretative' and are usually qualitatively evaluated. These methods compute how much each input feature contributes to the model output/prediction, but they do not explain the input features' interdependence nor the order of importance. In this chapter, we propose the U Analysis (UA), a systematic method to verify the co-dependence of input image patches for model prediction, a group of patches that must coexist for the model to predict accurately.

Algoritmo 4: U Analysis.

Input: Given a trained deep learning model f, input image x, target, interpretability vector I, window size w, order type $order$, and noise type n

$region\_weights \leftarrow get\_region\_importance(I, w)$;
$region\_sorted \leftarrow sort\_region(region\_weights, order)$;
$gen\_batch \leftarrow remove\_regions(x, region\_sorted, noise)$;
$pred\_batch \leftarrow f(gen\_batch)$;
$y\_batch\_pred \leftarrow pred\_batch.argmax(1)$;
$pos\_pred\_correct \leftarrow where(y\_batch\_pred == target)$;
$u\_triples \leftarrow []$;
for $i = 0$ **to** $len(pos\_pred\_correct) - 1$ **do**
    $idx\_left = pos\_pred\_correct[i]$;
    $idx\_right = pos\_pred\_correct[i + 1]$;
    if $(idx\_right - idx\_left)$ *is larger than one* **then**
        $middle\_idx \leftarrow choice\_between(idx\_left, idx\_right)$;
        $u\_triples.append((idx\_left, idx\_middle, idx\_right))$;
return $u\_triples$

The Algorithm 4 presents the U Analysis steps with a Python-based syntax. Given an input image x, model f, and attribution map I, the UA method first computes the importance of each x's patch of dimension $W \times W$ by summing up all attribution weight of each respective feature. Next, it sorts the x's patches by importance. It cumulatively replaces each one of the input images by noise, creating new x_i images, where the x_i images are equal to x, except that

they do not have the information about the first i patches (i.e., gen_batch variable). Thus, assuming that the input image has N patches, the x_N image does not have any information (i.e., only noise such as zeros, ones, or Gaussian noise). Figure 26 shows a sample of the UA processing.

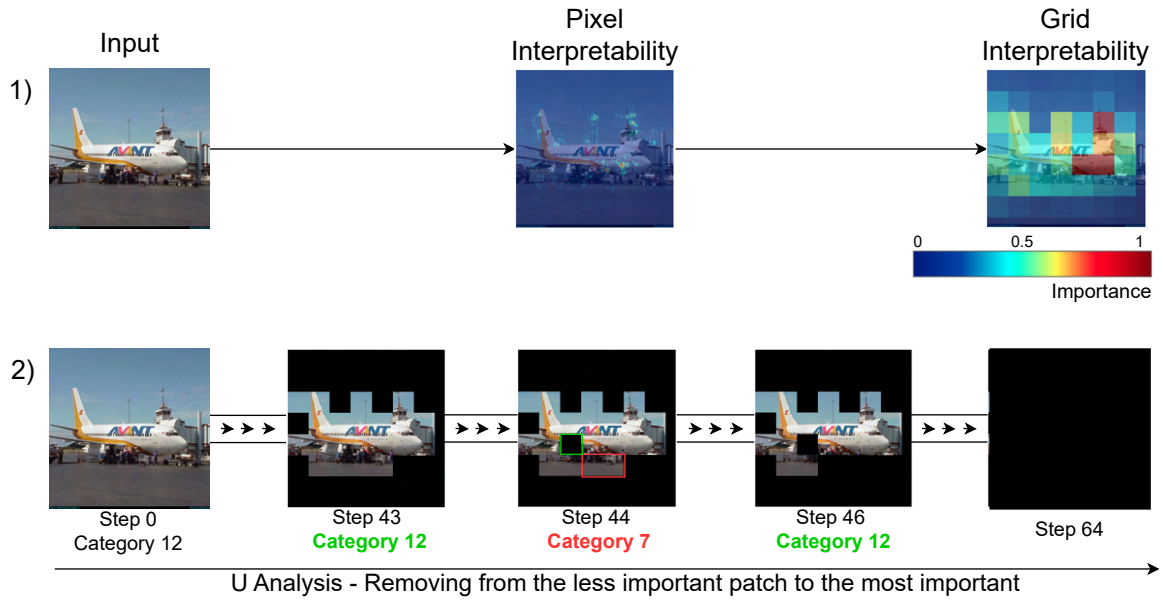

Figure 26 – **U Analysis pipeline.** The UA has two main steps: 1) given an input image, it computes the model inference and interpretability, and then it processes the grid-level interpretability. 2) Given the grid level interpretability, it sorts each image patch according to its importance. It removes patch-by-patch from the input image from the least important to the most important.

After constructing the sequence of new images, it is possible to see that all the information from image x_i is present in $x_{j<i}$. Thus, if we have a case where $left < middle < right$ and $f(x_{left}) = y$, $f(x_{middle})! = y$, and $f(x_{right}) = y$, it means that the information contained in x_{left} and x_{right} is sufficient for the model to infer correctly. However, the patches in x_{middle} but not in x_{right} create negative strength for y when they coexist with the other patches. We call this counter-intuitive case a U-occurrence.

7.2 ITERATIVE POST HOC ATTRIBUTION

Given a trained neural network f and an input vector x, the interpretability methods produce an interpretability map (or attribution maps) map whose dimension is equal to x dimension and the map_i value in i position means how important is the feature x_i to $f(x)$ prediction. Due to the subjectivity of the term $importance$, the evaluation of interpretability methods is mostly qualitative. Sometimes, it is hard to evaluate the interpretation because different interpretability methods produce different interpretations from the same model and input. Thus, due to its subjectivity, in this chapter, we propose a direct hybrid approach combining

optimization methods with the deep neural network to select the features responsible for producing the model prediction.

The first definition says that if a region r is relevant to the model prediction, the model prediction will decrease if we erase its information. Next, the second definition shows how we can compare the importance of two distinct regions.

- **Definition 1** Given a model $f : R^n \rightarrow \{0, 1\}$. A region r is important to f prediction only if $f(x) > f(degrade(x, r))$.
- **Definition 2** A region r_i is more important than r_j to f prediction only if $f(degrade(x, r_i)) < f(degrade(x, r_j))$.

Given a trained deep neural network f and an input image x, we would like to know which pixels from x most contribute to the $f(x)$ output according to definitions 1 and 2. Thus, we can model this question as an optimization problem present in the equation 7.1. To solve the $argmax$ problem in the equation 7.1, we can employ search and optimization algorithms such as Hill Climbing (GENT; WALSH, 1993), Ant Colony (DORIGO; BIRATTARI; STUTZLE, 2006), Genetic Algorithms (GOLDBERG, 1989), Particle Swarm (KENNEDY; EBERHART, 1995), and others.

$$\arg\max_{mask} Importance(mask) = f(mask \odot x + (1 - mask) \odot C) \tag{7.1}$$

The algorithm 5 and figure 27 present the IPHA method. It receives the deep learning model f, an input vector x, and returns two versions of the vector x, one with the important features ($x\_important$) and the other with only the non-important features of x ($x\_non\_important$).

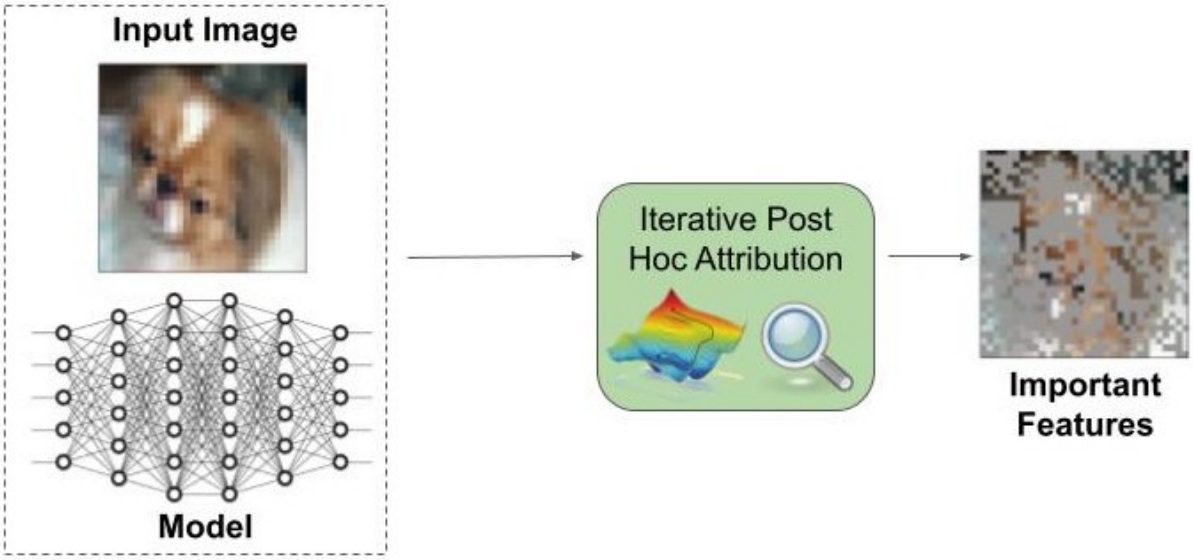

Figure 27 – Overview of the iterative post hoc attribution method.

Algoritmo 5: Iterative post hoc attribution.

Input: Given a trained deep learning model f, input vector x, constant vector C, and an optimizer algorithm

$P_y \leftarrow f(x)$;
$eval \leftarrow lambda\, x, c, mask : f(mask \odot x + (1 - mask) \odot C)$;
$important\_features \leftarrow optimizer(f, x, c, eval)$;
$x\_important \leftarrow important\_features \odot x + non\_important\_features \odot C$;
$x\_non\_important \leftarrow non\_important\_features \odot x + important\_features \odot C$;
return $x\_important, x\_non\_important$

Since we want to select the important pixels in x, the mask parameter in the equation 7.1 is a vector composed of 0 and 1 with the exact dimension of x, that is $mask \in R^n$. The vector C is a constant vector responsible for filling the information in x removed by mask. C is a constant because we do not want to insert any new pattern in x, so it needs to be unbiased. If we search in the space $\{0, 1\}^1$ to select the mask that maximizes the $Importance(mask)$ function, we should evaluate 2^n masks. This has a high computational cost for the analysis and can be impractical. However, we can search for an approximate solution using local search methods to mitigate this computational cost. The local search methods are iterative algorithms that begin with an arbitrary solution to the problem and make small changes to find better solutions.

We can use local search methods to mitigate the 2^n (n is the number of features) cost and find an approximate solution to our problem. We jointly employ the local search method Hill-Climbing (GENT; WALSH, 1993) with the neural network f to obtain preliminary results. The algorithm 6 presents our hill-climbing implementation.

Algoritmo 6: Hill Climbing.

Input: $f, x, C, num\_iterations, num\_neighbors$

```
best_mask ← random(x.shape, 0, 1) ;
for i ← 0; i < num_iterations do
    neighbors ← get_neighbors(best_mask, num_neighbors) ;
    next_eval ← −INF ;
    next_node ← NULL ;
    for mask ∈ neighbors do
        if next_eval < eval(mask) then
            next_eval ← eval(mask) ;
            next_node ← mask ;
    if eval(best_mask) < next_eval then
        best_mask ← next_node ;
return best_mask
```

As we can see from the algorithm 6, the $get\_neighbors$ function is a significant part of this

method because it generates new solutions from the actual best solution. Since our solution is a mask composed of 0 and 1, to create a new neighbor, we randomly select a position in the mask and change to 0 a grid of dimension 2×2 around it. Another important part of our solution is the $eval$ function. We have defined it as the $Importance$ function present in the equation 7.1.

7.3 MODEL INSPECTOR

The image comprises different types of visual information, such as shape, color, texture, patterns, and objects. Each type of information may impact the model classification decision in different ways. For example, a model may be biased to texture, color, or shape (NGUYEN; YOSINSKI; CLUNE, 2015; ALCORN et al., 2019; HENDRYCKS et al., 2021; ZHANG; ZHU, 2019). Thus, tweaking these types of information in the input image and evaluating the model with the new image can help assess the classification model's robustness regarding different versions of the same signal, thus producing a local analysis of the model. Interpretability methods also can be used to debug image classification models. They produce how important is each input image pixel for model decision and attribution map to visualize. Thus, the pixel importance can be used to compute metrics such as Top-K erasing and RFS (MOAYERI et al., 2022)

Beyond these types of visual information, an image can be composed of two main spatial regions: foreground and background. The foreground is the image's main focus, which includes the subjects or objects of interest. On the other hand, the background is all the information that is not in the foreground. Usually, in image classification tasks, we want to classify the information that is in the foreground. Thus, it can be considered the signal while the background is the context. Background robustness is the ability of an image classification model to classify a signal even when it is on a different background. (XIAO et al.,) showed that image classification models may be biased to background information and make a wrong prediction even when the foreground is present in the image but has a not common background. Thus, it is important to evaluate if the signal information is enough for the model to classify the image accurately or if the model is biased to background information.

The discussion presented so far shows we need a pipeline to evaluate the image classification model weakness. Therefore, we propose the image classification Model Inspector tool [6], whose goal is to allow users to evaluate image classification robustness against different types of transformations. It comprises three modules in which the user can evaluate the image classification against different image processing functions, interpretability maps, and signal vs. noise sensitivity. The Model Inspector is developed as a Web App in which the user can load models from Pytorch (PASZKE et al., 2019) or Timm[7] library and interact with its input-output results. In the following, we will detail each Model inspector's module.

[6] https://github.com/faos/image-classifier-model-inspector
[7] https://timm.fast.ai/

7.3.1 Image processing

The image processing module comprises several image transformation functions that add noise to the input image. The user can apply these transformations in the input image and visualize the model output difference with the original image, thus getting insights about the model robustness related to the transformation. The module implements three types of noise: Gaussian, Shot, and Impulse. Although we may add them intentionally, they can be naturally caused by phenomena such as random variations in light, sensor noise in the camera, interference in the transmission process, or bit errors. These noise functions can also be found in (HENDRYCKS; DIETTERICH, 2019), where the authors created ImageNet (DENG et al., 2009a) variations with them to evaluate the robustness of several image classification architectures on the ImageNet (DENG et al., 2009a) dataset.

Gaussian, Shot, and Impulse noise are transformations that change the image color of texture. However, in addition to them, the image processing module also has spatial transformations (e.g., Patch Shuffle, Horizontal Shuffle, Vertical Shuffle) that deform the shape of the objects in the image. Still, it keeps the texture and color information, so it is helpful to infer whether the object's shape is important to the model predictions. Patch shuffle transformation splits the input image in disjoint squared patches with size $W \times W$, shuffles them, and creates a new image. On the other hand, horizontal shuffle creates disjoint horizontal patches with height H and size equal to the input image, shuffle them, and create a new image. The vertical shuffle is similar to the horizontal; the difference is that the patches are vertical, so their size is equal to the image height, and the width is W. Figure 28 presents the pipeline of this module and an example of each analysis function.

7.3.2 Interpretability

The interpretability module comprises two main components: (1) interpretability methods and (2) U Analysis. In the first component, we implement a wrapper for the Captum library so the user can select the interpretability method and visualize its outputs. The second component implements the U Analysis discussed before, where the user can choose the noise method and window size. Besides, if the analysis finds counter-intuitive samples, it will show them. It is important to highlight that the attribution map used in the U Analysis is the output of the first component.

7.3.3 Signal

The signal module allows the users to interact with the input by selecting which region of the input image they consider the signal. The users may select the signal with three formats: rectangle, circle, and polygon. After selecting the signal, the module computes the signal-to-noise and background texture analyses. The signal-to-noise analysis calculates the importance of the signal region and compares it with the context to verify which region is more important to

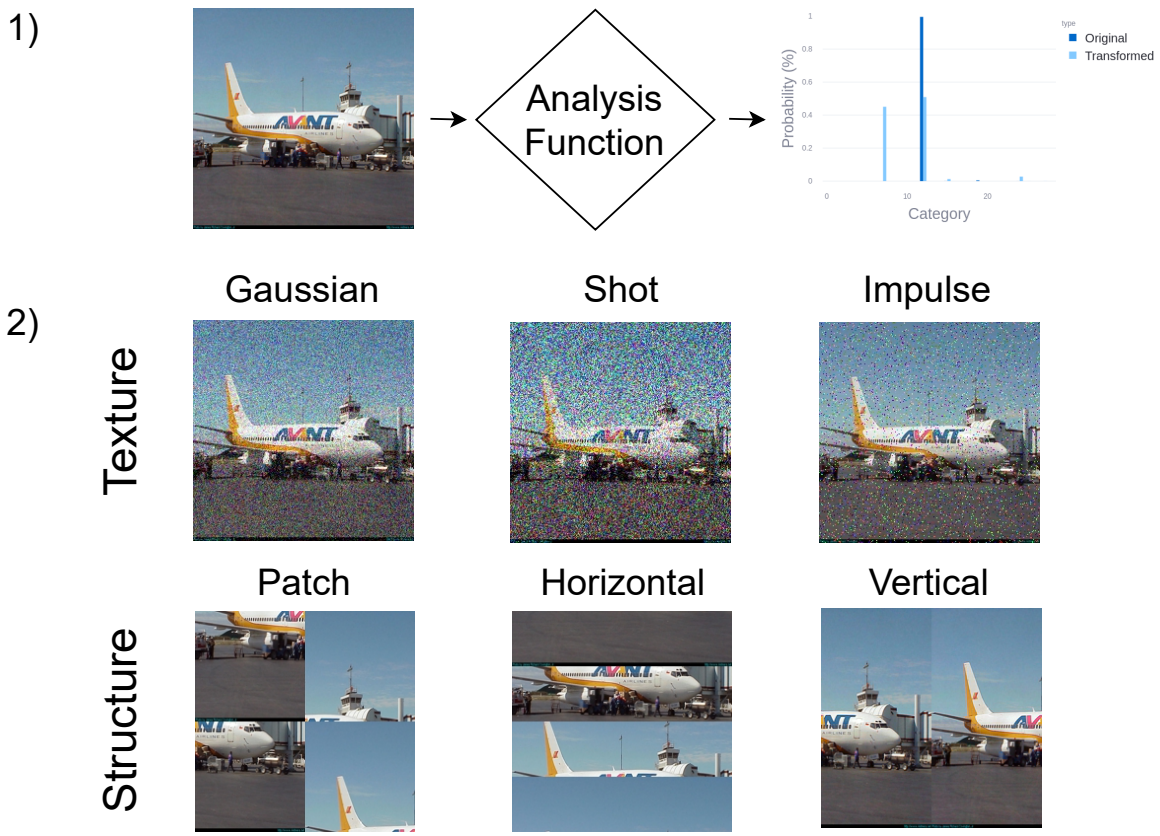

Figure 28 – **Image processing pipeline.** Part 1) shows the main pipeline of the image processing. Given an input image, the user should select which function it will use to process the image and compare the model inference with the original image inference. Part 2) shows the functions available in the model inspector; they are grouped into texture and structure.

the model. The background texture analysis allows the user to apply all the image transformations from the first module to the image background only, thus verifying if the model decision is impacted by background texture changes while keeping the signal information. The noise analysis proposed in (MOAYERI et al., 2022) inspired the background texture analysis. Figure 29 presents the signal pipeline.

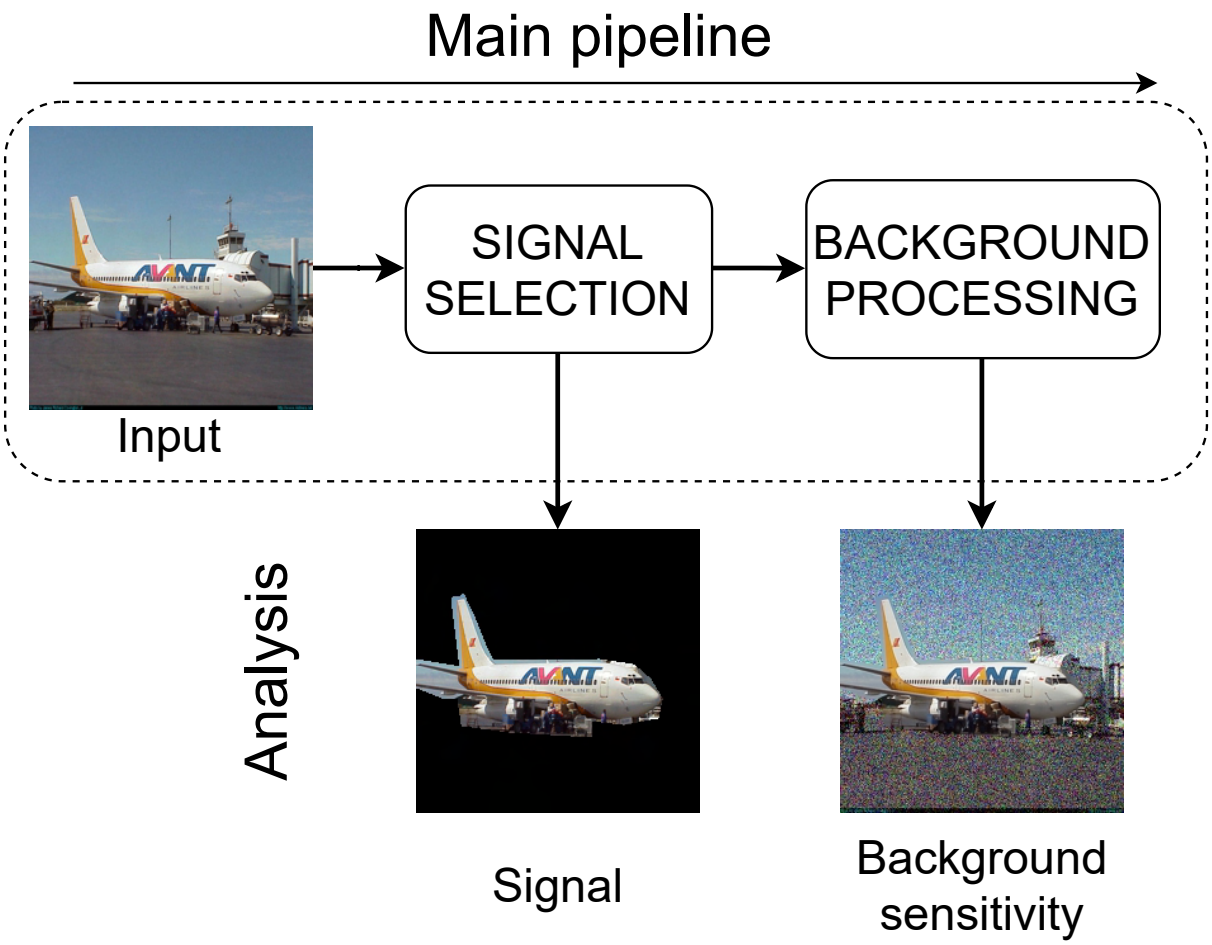

Figure 29 – **Signal analysis pipeline.** This module allows the user to select the signal of the input image using different formats, for example, polygon, rectangle, and circle. After selecting the signal, it can add the transformation to the image background to verify the model sensitivity to background changes while keeping the original signal. The main pipeline comprises three steps: 1) select the input image, 2) select the signal information, and 3) apply the background processing functions. The signal selection and background processing have an output to compare the model inference using only each information.

7.4 EXPERIMENTS AND RESULTS

We group the experiments and results according to each method and tool. Thus, next we present a section for each one.

7.4.1 U Analysis

This section presents the experiments and results achieved with the U Analysis. First, we describe the datasets and architecture used and then present the percentage of U occurrence found. To perform the experiments, we used the CIFAR-10 (KRIZHEVSKY; HINTON et al., 2009) and Self-Taught Learning 10 (STL-10) (COATES; NG; LEE, 2011) datasets. The CIFAR-10

dataset comprises 60,000 32x32 color images grouped into ten classes and has 50,000 images for training and 10000 for testing. It is a well-balanced dataset. Thus, each class has 5,000 training and 1,000 testing images.

On the other hand, STL-10 has the same classes as CIFAR-10 (i.e., airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck) but has 13,000 96x96 color images, where 5,000 are for training and 8,000 for testing. We train a ResNet-18 (HE et al., 2015) instance for each dataset using SGD with a learning rate $1e-2$. Each network was trained by 50 epochs, and we chose the model with the best accuracy on the test set to perform the U Analysis.

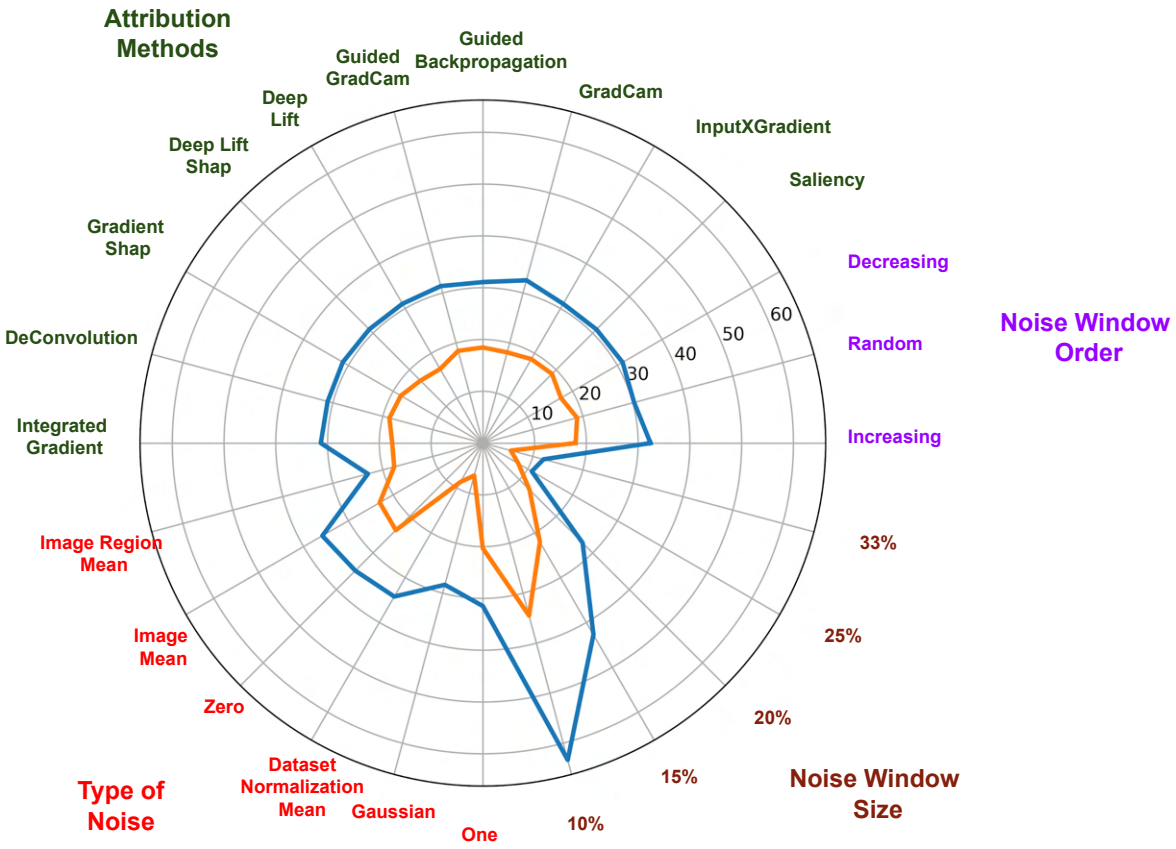

Figure 30 – **U analysis results.** The graph shows the results grouped by each parameter type (i.e., Attribution methods, type of noise, noise window size, and noise window order) and value. Besides, each curve represents the results for each dataset. The blue curve represents the results obtained from the CIFAR-10 dataset, while the orange is with STL-10.

The U analysis has several hyperparameters. For example, we can use different interpretability methods to compute the contribution of each input pixel to the model output. The order in which we sort the patch can be random, increasing, or decreasing, and we can replace the original patch information with several types of noise, and the patch's size (noise window size) itself is a parameter. To perform the U Analysis with the ResNet-18, we use 10 different interpretability methods, 3 sorting types, 6 noise types, and 5 different patch sizes, resulting

in 900 runs for each dataset. Figure 30 presents the results achieved by the U analysis with all datasets and parameters. The results group the U occurrence for each hyperparameter value to infer which configuration is more susceptible to finding counter-intuitive behavior in image classification.

The findings show that all the attribution methods used have almost the same U occurrence. Thus, they affect it in the same way. This conclusion is similar to the sorting method, in which all of them have almost the same U occurrence percentage, except the Increasing order in CIFAR-10, which is slightly higher than others. Although the attribution methods and noise window order have close U occurrence percentages, the type of noise has different values for each parameter value. The U occurrence was the lowest for both datasets when we used the image region mean and Gaussian noise. We argue that this behavior can be due to different reasons. For example, while the Gaussian noise does not represent information regarding the dataset, it is easy for the model to ignore it; the image region mean is a statistic of the patch that was removed. Thus, it still has information about the original patch.

The results show that the noise window size is the most important hyperparameter, with a patch size of 10%, the parameter value with the most U occurrence in all scenarios. This result indicates that tiny patches instead of bigger ones may impact the ResNet-18, as the 33% presents a low U occurrence. In addition, the ResNet-18 may correlate the features of lower patches instead of bigger ones.

7.4.2 ITERATIVE POST HOC ATTRIBUTION

To evaluate the IPHA method, we have compared the mask obtained using the IPHA with Hill-Climbing with a wide range of interpretability methods (e.g., Saliency, Guided Backprop, GradCam, Guided GradCam, and Integrated Gradients). Since the mask represents the location of the important features, we compute the model outputs in two scenarios: (i) when we use only the pixels of the mask and (ii) when we do not use the features in the mask (i.e., (1 - mask)). Since it obtains the most important features in the first scenario, the model output must be high (or close to when using the full features), while the model output must be low in the second scenario. We trained a ResNet (HE et al., 2015) model with the CIFAR-10 (KRIZHEVSKY; HINTON et al., 2009) dataset to perform the experiments.

In the equation 7.1, we have presented our optimization problem. The C vector is a constant that must be a neutral value. To be fair, in the experiments, we have tried different types of the constant vector. In figure 31, we present all types of constants used: the noise value is black (0), Gaussian, Normalization mean, and white (1). The rows represent the types of noise, and the columns represent the mask with the most important pattern according to each interpretability method. It is important to highlight that to obtain the most important features from the Saliency, GradCam, Guided Backpropagation, Guided GradCam, and Integrated Gradients, we have computed the interpretability and selected the top-k features, where k is 512 because we have 1024 features in total.

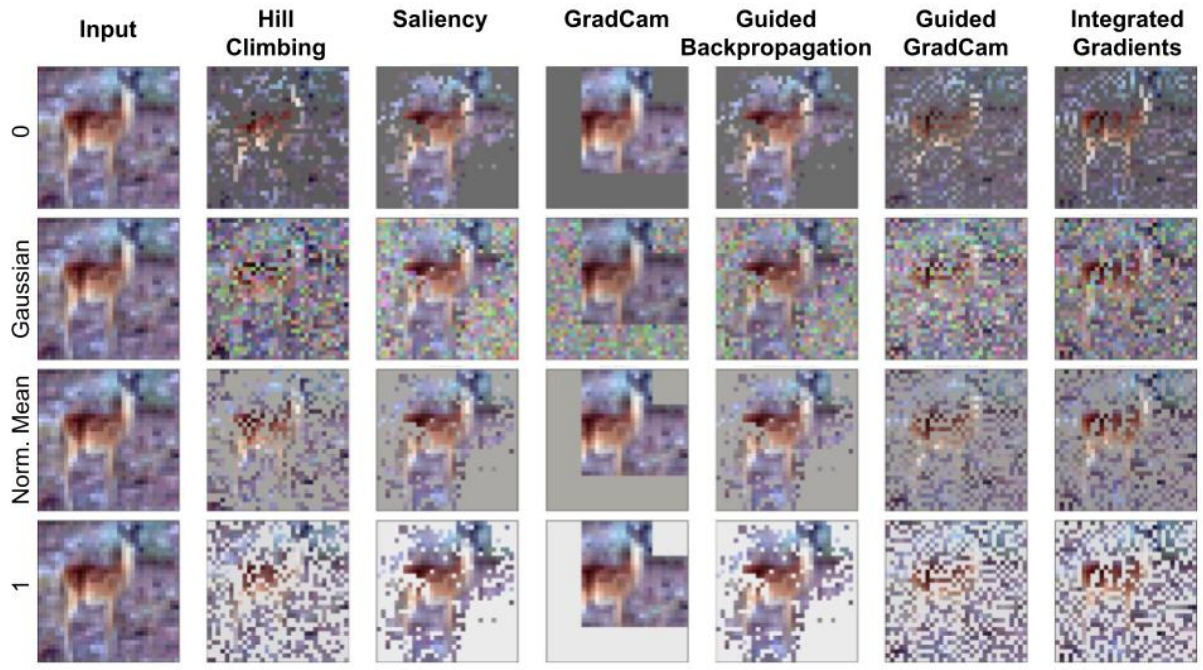

Figure 31 – Samples of noisy types used in the experiments.

We have defined the Feature Impact index (FII) present in equation 7.2 to evaluate our method. The goal of FII is to compute the absolute distance between the model output with the complete information $f(x)$ and the model output using only the selected features $f(x_{selected})$. The selected feature may be the (i) non-important features or the (ii) important features. Thus, when we use the non-important as the selected feature, we expect that the FII returns a low value. However, when we use the important features, the FII must be higher.

$$FeatureImpactIndex(f, x, x_{selected}) = abs(f(x) - f(x_{selected})) \tag{7.2}$$

Figure 32 presents the results of the analysis from the mask with the less important pixels. As the mask indicates the less important pixels, we expect that the model considering only those pixels is low, so the difference between the original model output and the output based on the non-important features will be higher. Since we believe only the non-important pixels, we must replace the important with some neutral value. Thus, we use different constant values to produce the results. Each y-axis in the graph means the impact of the model when using the respective constant type. The results show that the Hill Climbing (HC) approach has found the less important features and achieved almost 95% when considering the 0 value as constant. Besides, all results obtained from the HC method are higher than 80%. Although the other interpretability methods are established in the literature, they did not achieve any results above 80%.

From the figure 33, we can see the results from the analysis of when we use the mask pointing to the important features. Since the mask points to the important features, we expect that the model output is high. So, the difference between the original model output and the

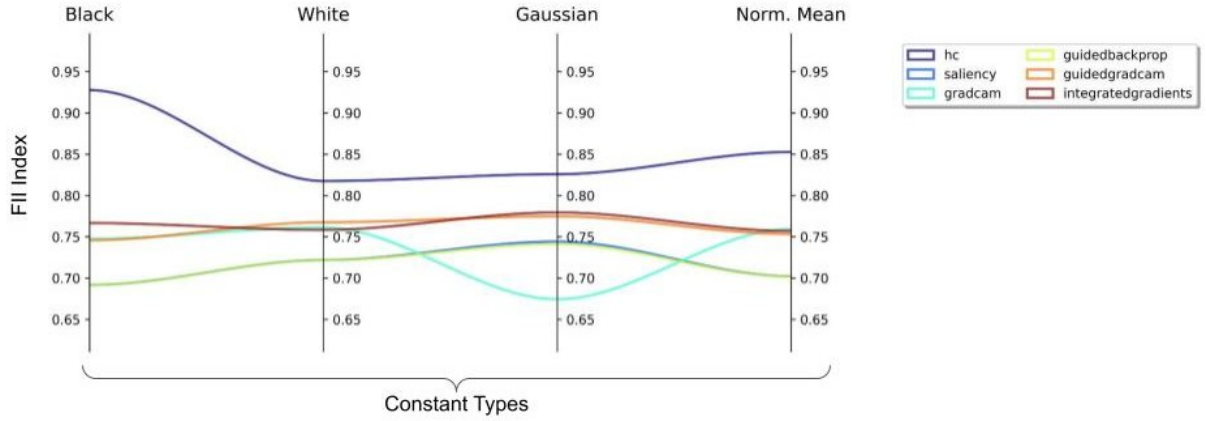

Figure 32 – Results of the first scenario, evaluating the *non-important* features selected by each method. The FII index in the y-axis is the average of all images in the test set of the CIFAR-10. Each y-axis represents the results obtained from different types of constant values. Examples of the constant values are present in the figure 31. Since we are selecting the non-important features in this graph, if the method select it correctly then the model output is close to 0 and the difference between the original prediction the non-important feature prediction should the higher. Thus, the higher the FII index, the better the interpretability method in this scenario.

output based on the important features is lower. We also used different constant values in the evaluation process as in the previous scenario. The results show that, in our evaluation scenario, the GradCam is the best method to indicate the important feature. Their average difference is lower than 30% in every constant value. Considering only 0 as the constant value, the HC approach could find the non-important features and produce results close to the GradCam. However, in other scenarios, their outputs were at least 20% higher than the GradCam.

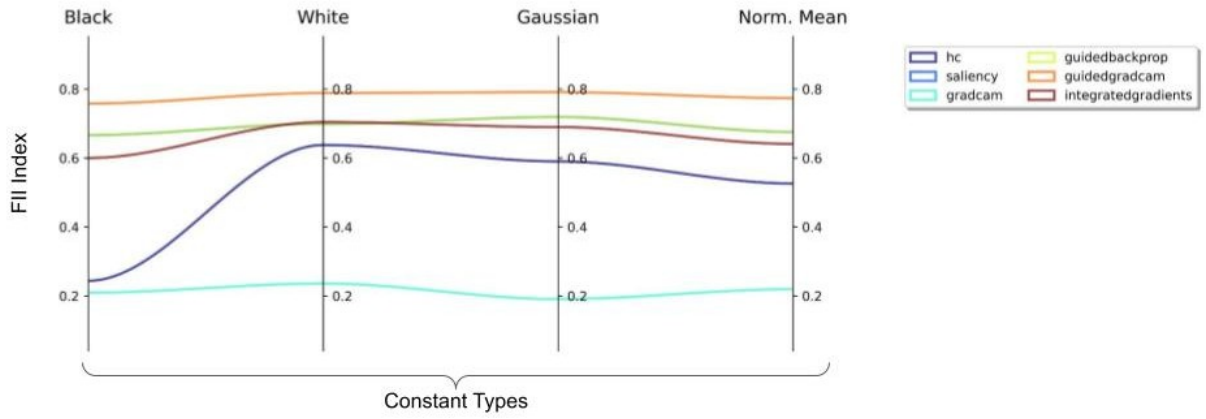

Figure 33 – Results of the second scenario, evaluating the *important* features selected by each method. As this scenario refers to the most important features, the model output with the selected features is expected to be close to the model output with the full features. Thus, a lower the FII index better is the interpretability method in this scenario.

The results obtained from the GradCam method in the second scenario are intriguing because it remained consistent in all constants. We believe this result is because when we select the lowest top-512 features using the GradCam attribution maps, the model selects squared regions, thus erasing a meaning pattern (e.g. body of the deer). This particularity of GradCam lies in its ability to generate an attribution map that matches the size of the input vector. To achieve this, its interpretability derived from the convolutional layer must be resized to the input size (In our experiment, we resized an 8x8 attribution (from the last convolutional layer) to fit a 32x32 input size). Consequently, this resizing process may result in neighboring pixels having nearly identical attribution values. On the other side, the other interpretability methods selects specific pixels. The Figure 31 highlights this difference.

7.4.3 Model inspector demonstration

This section analyzes the model inspector tool and shows how the user can use it to infer insights about image classifier models. This analysis uses a ResNet-18 architecture trained with FGVC Aircraft (MAJI et al., 2013) dataset to classify the aircraft manufacturer. Figure 34 presents the outputs obtained from the model inspector and has three crops extracted.

Part 1 shows the Image processing module applying the Gaussian transformation on the input image, while part 2 is the Patch shuffle transformation. Part 3 shows a sample of the signal module when we select the aircraft as the signal and apply the Gaussian noise into the background. All three parts have a barplot on the right to compare the ResNet output when we input the original image and the respective transformed image. Part 1 shows that when we insert Gaussian noise in the input image, the model changes its prediction, thus being sensitive to Gaussian noise. Part 2 also shows that when we destroy the spatial information with patch shuffle, the model also changes its prediction, which means that the signal structure is important for the prediction. Finally, part 3 shows that when we insert noise only in the background, the model does not change its prediction. Thus, joining these results with part 1, we can conclude that the model is sensitive to change in the signal only as it keeps its decision when we keep the original signal.

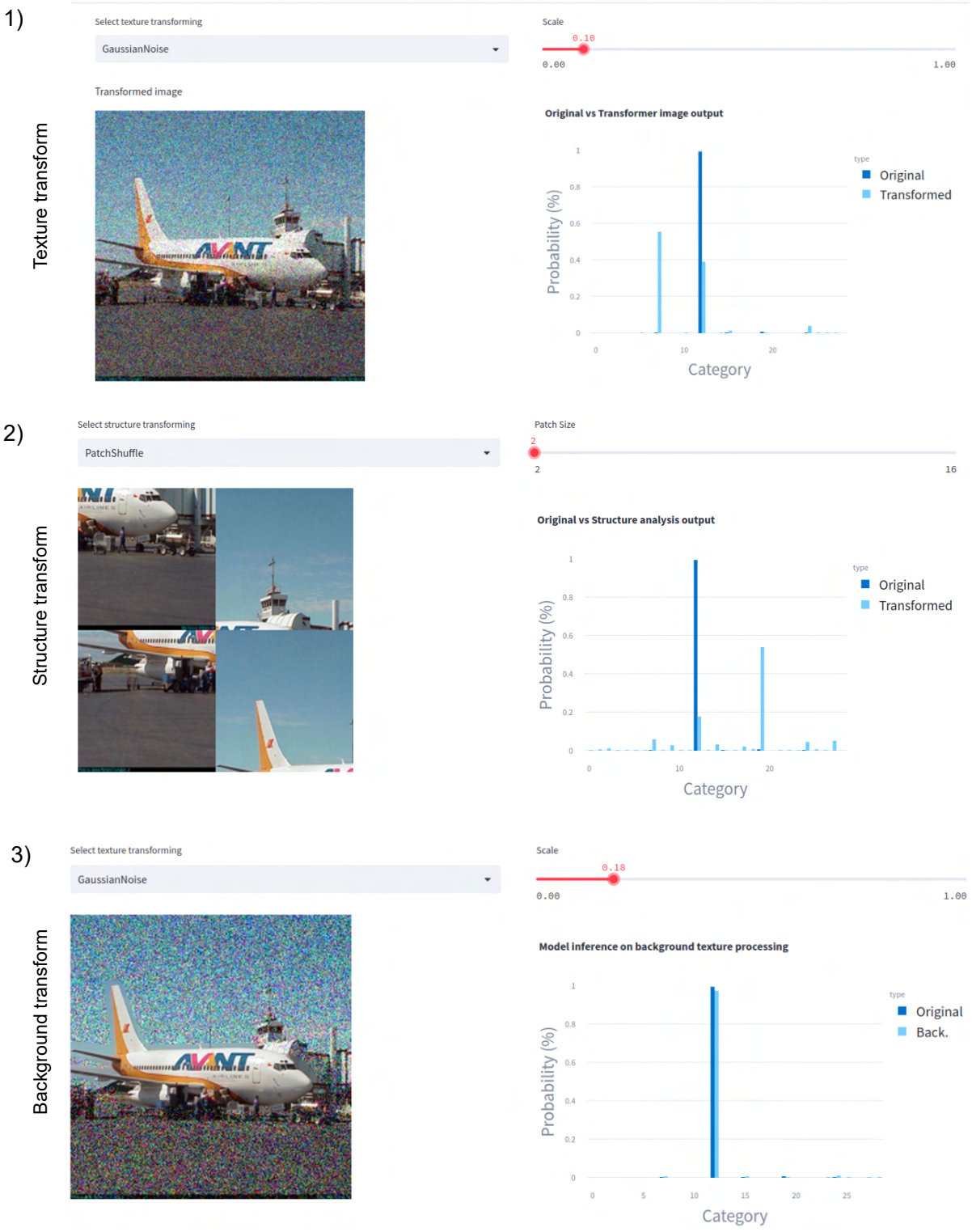

Figure 34 – **Model inspector demonstration.** Parts 1 and 2 show the outputs of the Image processing module for texture and structure transformation, respectively. Part 3 shows the result of the signal background texture transformation. On the left side, all parts have a select box so the user can select the transformation, and on the right side, there is a slider so the user can select the parameter value for the transformation.

7.5 CONCLUSION

In this chapter, we presented U Analysis, IPHA, and the Model Inspector tool. U Analysis is a novel method for visualizing and interpreting the behavior of image classification models. It allows us to understand the importance of patches in an image and their interactions, which can be used to understand how models make inferences and identify their weaknesses. IPHA is an optimization view for the interpretability of deep learning models. In addition, the Model Inspector tool allows users to interact with the input image and analyze the robustness of image classification models by changing visual information, such as texture, color, and shape.

Our experiments show that after applying U Analysis, we can demonstrate that the U-occurrence phenomenon can occur in some cases, thus showing the image classification models have counter-intuitive feature interaction. Besides, the IPHA results show that it can select the less relevant features to the model prediction more accurately than the interpretability methods used in the experiments. Finally, we also showed that Model Inspector can be used to evaluate the robustness of image classification models to different versions of an image and to detect biases in the model's decision-making process.

The U Analysis, IPHA, and Model Inspector are powerful tools for understanding and interpreting image classification models and can be used to improve their performance and identify their weaknesses. These methods may help further research in model interpretability and lead to developing even more advanced tools for analyzing and understanding deep learning models.

8 CONCLUSIONS

This thesis shows that sometimes deep learning models for image recognition make the right inference (e.g., segmentation, classification) based on feature information that is not considered relevant for the problem-domain experts (e.g., shortcut learning, background information). We illustrated this weakness with examples such as Grey matter segmentation in Chapter 3, Background bias in Chapter 4, and a Toy problem in Chapter 5. We argue that it is possible to improve this by dynamically removing the background information that the model is attributing high importance and training the model (Active image data augmentation, Chapter 3). Augmenting the background information helps the model ignore the unrelated information from the training sample and infer based on the signal information (Right reasons data augmentation, Chapter 4). We also show that adding adversarial samples into the right for the right reasons training pipeline helps the trained model focus on the information's structure (Adversarial right for the right reasons, Chapter 5). Finally, we evaluated the robustness of visual language models to background shift, a common problem in standalone models and still present in large models (Chapter 6). We also proposed a new tool and two analysis methods to help practitioners debug its models and gain insights about its inference: the Model inspector tool, U Analysis, and Iterative post hoc attribution (Chapter 7).

After evaluating the proposed methods with experimental pipelines, we show that ADA enhances the robustness of the U-Net by helping it segment the spinal cord grey matter considering the signal information. The proposed RRDA improves the model bias to background information and indicates that a high signal-to-noise ratio does not necessarily mean the model is robust to the background. In addition, the ARRR method is effective in enhancing model robustness in structure-based and challenging to use in texture-based image classification problems. Finally, the VLMs background analysis has shown that ALIGN is more background robust than CLIP, and the Model inspector tool, U Analysis, and IPHA methods demonstrate the occurrence of counter-intuitive feature interactions and provide a more accurate selection of relevant features compared to existing interpretability methods. These tools offer powerful insights into the model behavior, aiding in performance improvement and weakness identification.

The main objective of this thesis is to improve the deep learning-based image recognition models concerning robustness and interpretability. The results and conclusions demonstrate that we achieved this goal as the ADA, RRDA, and ARRR methods allow us to train more robust models, focusing on features related to the problem (signal and structure) instead of spurious features (e.g., background and color bias). In addition, the VLMs background analysis, IPHA, U Analysis, and Model Inspector tool address the interpretability issue, enabling us to understand how VLMs make an inference, objectively interpret the model inference, and allow the user to verify how the model changes its inferences according to image transformations.

8.1 PUBLICATIONS

This section presents a summary of all contributions made during the development of this thesis. These works illustrate the breadth and depth of our efforts to make advancements toward model robustness and interpretability. The publications are sorted by the year.

1. Santos, Flávio Arthur Oliveira, Cleber Zanchettin, Leonardo Nogueira Matos, and Paulo Novais. "Active image data augmentation." In Hybrid Artificial Intelligent Systems: 14th International Conference, HAIS 2019, León, Spain, September 4–6, 2019, Proceedings 14, pp. 310-321. Springer International Publishing, 2019. (HAIS, 2019)

2. Santos, Flávio Arthur Oliveira, Cleber Zanchettin, José Vitor Santos Silva, Leonardo Nogueira Matos, and Paulo Novais. "A hybrid post hoc interpretability approach for deep neural networks." In International Conference on Hybrid Artificial Intelligence Systems, pp. 600-610. Cham: Springer International Publishing, 2021. (HAIS 2021)

3. Arthur Oliveira Santos, Flávio, Cleber Zanchettin, Leonardo Nogueira Matos, and Paulo Novais. "On the Impact of Interpretability Methods in Active Image Augmentation Method." Logic Journal of the IGPL 30, no. 4 (2022): 611-621. (IGPL 2022)

4. Santos, Flávio AO, Maynara Donato de Souza, Pedro Oliveira, Leonardo Nogueira Matos, Paulo Novais, and Cleber Zanchettin. "Image Classification Understanding with Model Inspector Tool." In International Conference on Hybrid Artificial Intelligence Systems, pp. 611-622. Cham: Springer Nature Switzerland, 2023. (HAIS 2023)

5. Santos, Flávio Arthur O., Maynara Donato de Souza, and Cleber Zanchettin. "Towards Background and Foreground Color Robustness with Adversarial Right for the Right Reasons." In International Conference on Artificial Neural Networks, pp. 169-180. Cham: Springer Nature Switzerland, 2023. (ICANN 2023)

6. Santos, Flávio Arthur Oliveira, and Cleber Zanchettin. "Exploring Image Classification Robustness and Interpretability with Right for the Right Reasons Data Augmentation." In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, pp. 4147-4156. 2023. (ICCVW 2023)

7. Santos, Flávio Arthur Oliveira, Maynara Donato de Souza, and Cleber Zanchettin. "Evaluating zero-shot image classification based on visual language model with relation to background shift." In Neural Information Processing Systems Conference: LatinX in AI (LXAI) Research Workshop 2023, New Orleans, USA, 2023. (LXAI NIPS 2023)

8.2 LIMITATIONS AND FUTURE WORKS

Despite the improvements achieved through our contributions, it is important to recognize the limitations that each one of them has. Next, we will highlight the individual limitations and thoughtful consideration for future works and directions that need more investigation.

Application: A natural extension of the ADA and RRDA is the application for other tasks of computer vision (i.e., object localization) and other domains, such as natural language processing (NLP) and time-series (TS). Although both implementation and proposal are generic to use binary input masks, defining them for NLP is challenging because of issues such as ambiguity and semantic complexity. Hence, they must be well-defined and guarantee no loss of information and context.

Methods combination: As the IPHA method searches for a mask that points out the important features and another one that points out the non-important features, we could combine the IPHA method with ADA and evaluate its impact as an interpretability method. Besides, we can also evaluate the interpretability maps of the RRDA models with IPHA to verify if there is any impact concerning non-RRDA models.

IPHA: The main contribution of the IPHA is to visualize interpretability as an optimization problem with well-defined objectives, with the subjectivity of concepts as importance or attribution. Here, we only evaluated it with the Hill Climbing (GENT; WALSH, 1993) method to search for an optimization solution. However, it is only a proof-of-concept and still has room for improvement. Naturally, we can apply other search methods such as Ant Colony (DORIGO; BIRATTARI; STUTZLE, 2006), Genetic Algorithms (GOLDBERG, 1989), Particle Swarm (KENNEDY; EBERHART, 1995), and others (HOOS; STÜTZLE, 2015). In addition, we can improve the mask structure generated by IPHA, forcing it to generate masks with some specific structure (e.g., rectangle, circle, square, and other).

Model inspector: Model inspector allows users to perform input image transformations and verify how the model changes its predictions, thus possibly identifying model weakness and bias. In the current version, the Model Inspector only allows user load weights of image classification models available from timm [1] library and Pytorch [2]. Thus, a future implementation is necessary to allow users to load their custom models or even a generic model implemented using a different framework. Besides, it also can be extended to other computer vision tasks such as image segmentation or localization.

Signal masks: The results (especially the RRDA) showed that with the input signal mask, we can train background robust models and guide the model to focus on the signal during the inference process. However, obtaining this mask is costly, as we need an additional label for each input data. Thus, we need additional improvements on RRDA to not consider the input mask, build the input mask automatically (without human labeling), or at least label a few examples and achieve the same robustness level. These variations are still open problems that

[1] <https://github.com/huggingface/pytorch-image-models>
[2] <https://pytorch.org/vision/stable/models.html#classification>

need to be solved.

REFERENCES

ABADI, M.; BARHAM, P.; CHEN, J.; CHEN, Z.; DAVIS, A.; DEAN, J.; DEVIN, M.; GHEMAWAT, S.; IRVING, G.; ISARD, M.; KUDLUR, M.; LEVENBERG, J.; MONGA, R.; MOORE, S.; MURRAY, D. G.; STEINER, B.; TUCKER, P.; VASUDEVAN, V.; WARDEN, P.; WICKE, M.; YU, Y.; ZHENG, X. Tensorflow: A system for large-scale machine learning. In: *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*. [s.n.], 2016. p. 265–283. Disponível em: <https://www.usenix.org/system/files/conference/osdi16/osdi16-abadi.pdf>.

ALBER, M.; LAPUSCHKIN, S.; SEEGERER, P.; HäGELE, M.; SCHüTT, K. T.; MONTAVON, G.; SAMEK, W.; MüLLER, K.-R.; DäHNE, S.; KINDERMANS, P.-J. innvestigate neural networks! *Journal of Machine Learning Research*, v. 20, n. 93, p. 1–8, 2019. Disponível em: <http://jmlr.org/papers/v20/18-540.html>.

ALCORN, M. A.; LI, Q.; GONG, Z.; WANG, C.; MAI, L.; KU, W.-S.; NGUYEN, A. Strike (with) a pose: Neural networks are easily fooled by strange poses of familiar objects. In: *Proceedings of the Conf. on Computer Vision and Pattern Rec. (CVPR)*. [S.l.: s.n.], 2019.

AMAZON scientists applying deep neural networks to custom skills. <https://www.amazon.science/blog/amazon-scientists-applying-deep-neural-networks-to-custom-skills>. [Accessed 20-Oct-2022].

BACH, S.; BINDER, A.; MONTAVON, G.; KLAUSCHEN, F.; MÜLLER, K.-R.; SAMEK, W. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, Public Library of Science, v. 10, n. 7, p. e0130140, 2015.

BAO, Y.; CHANG, S.; YU, M.; BARZILAY, R. Deriving machine attention from human rationales. In: RILOFF, E.; CHIANG, D.; HOCKENMAIER, J.; TSUJII, J. (Ed.). *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*. Association for Computational Linguistics, 2018. p. 1903–1913. Disponível em: <https://doi.org/10.18653/v1/d18-1216>.

BEERY, S.; HORN, G. V.; PERONA, P. Recognition in terra incognita. In: *Proceedings of the European conference on computer vision (ECCV)*. [S.l.: s.n.], 2018. p. 456–473.

BILODEAU, B.; JAQUES, N.; KOH, P. W.; KIM, B. Impossibility theorems for feature attribution. *arXiv preprint arXiv:2212.11870*, 2022.

CARVALHO, D. V.; PEREIRA, E. M.; CARDOSO, J. S. Machine learning interpretability: A survey on methods and metrics. *Electronics*, MDPI, v. 8, n. 8, p. 832, 2019.

COATES, A.; NG, A.; LEE, H. An analysis of single-layer networks in unsupervised feature learning. In: JMLR WORKSHOP AND CONFERENCE PROCEEDINGS. *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. [S.l.], 2011. p. 215–223.

CODELLA, N. C.; GUTMAN, D.; CELEBI, M. E.; HELBA, B.; MARCHETTI, M. A.; DUSZA, S. W.; KALLOO, A.; LIOPYRIS, K.; MISHRA, N.; KITTLER, H. et al. Skin lesion analysis toward melanoma detection: A challenge at the 2017 international symposium on

biomedical imaging (isbi), hosted by the international skin imaging collaboration (isic). In: IEEE. *2018 IEEE 15th international symposium on biomedical imaging (ISBI 2018)*. [S.l.], 2018. p. 168–172.

DEEKS, A. The judicial demand for explainable artificial intelligence. *Columbia Law Review*, JSTOR, v. 119, n. 7, p. 1829–1850, 2019.

DENG, J.; DONG, W.; SOCHER, R.; LI, L.-J.; LI, K.; FEI-FEI, L. Imagenet: A large-scale hierarchical image database. In: IEEE. *2009 IEEE conference on computer vision and pattern recognition*. [S.l.], 2009. p. 248–255.

DENG, J.; DONG, W.; SOCHER, R.; LI, L.-J.; LI, K.; FEI-FEI, L. Imagenet: A large-scale hierarchical image database. In: IEEE. *2009 IEEE conference on computer vision and pattern recognition*. [S.l.], 2009. p. 248–255.

DEVLIN, J.; CHANG, M.-W.; LEE, K.; TOUTANOVA, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

DICE, L. R. Measures of the amount of ecologic association between species. *Ecology*, Wiley Online Library, v. 26, n. 3, p. 297–302, 1945.

DICE, L. R. Measures of the amount of ecologic association between species. *Ecology*, JSTOR, v. 26, n. 3, p. 297–302, 1945.

DONG, S.; WANG, P.; ABBAS, K. A survey on deep learning and its applications. *Computer Science Review*, Elsevier, v. 40, p. 100379, 2021.

DORIGO, M.; BIRATTARI, M.; STUTZLE, T. Ant colony optimization. *IEEE computational intelligence magazine*, IEEE, v. 1, n. 4, p. 28–39, 2006.

DOSHI-VELEZ, F.; KIM, B. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*, 2017.

DOSOVITSKIY, A.; BEYER, L.; KOLESNIKOV, A.; WEISSENBORN, D.; ZHAI, X.; UNTERTHINER, T.; DEHGHANI, M.; MINDERER, M.; HEIGOLD, G.; GELLY, S. et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

DOSOVITSKIY, A.; BEYER, L.; KOLESNIKOV, A.; WEISSENBORN, D.; ZHAI, X.; UNTERTHINER, T.; DEHGHANI, M.; MINDERER, M.; HEIGOLD, G.; GELLY, S.; USZKOREIT, J.; HOULSBY, N. An image is worth 16x16 words: Transformers for image recognition at scale. In: *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. Disponível em: <https://openreview.net/forum?id=YicbFdNTTy>.

DRUCKER, H.; CUN, Y. L. Improving generalization performance using double backpropagation. *IEEE transactions on neural networks*, v. 3, n. 6, p. 991–997, 1992.

DU, M.; LIU, N.; YANG, F.; HU, X. Learning credible deep neural networks with rationale regularization. In: WANG, J.; SHIM, K.; WU, X. (Ed.). *2019 IEEE International Conference on Data Mining, ICDM 2019, Beijing, China, November 8-11, 2019*. IEEE, 2019. p. 150–159. Disponível em: <https://doi.org/10.1109/ICDM.2019.00025>.

DUMITRU, M. A. K.-R. M.; PIETER, E. B. K. S. D.; KINDERMANS, J.; SCHÜTT, K. T. Learning how to explain neural networks: Patternnet and patternattribution. In: *International Conference on Learning Representations*. [S.l.: s.n.], 2018.

EISENSTEIN, J. Informativeness and invariance: Two perspectives on spurious correlations in natural language. In: *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. [S.l.: s.n.], 2022. p. 4326–4331.

ERION, G. G.; JANIZEK, J. D.; STURMFELS, P.; LUNDBERG, S.; LEE, S. Learning explainable models using attribution priors. *CoRR*, abs/1906.10670, 2019. Disponível em: <http://arxiv.org/abs/1906.10670>.

FLORA, M.; POTVIN, C.; MCGOVERN, A.; HANDLER, S. Comparing explanation methods for traditional machine learning models part 1: An overview of current methods and quantifying their disagreement. *arXiv preprint arXiv:2211.08943*, 2022.

FUCHS, F.; SONG, Y.; KAUFMANN, E.; SCARAMUZZA, D.; DÜRR, P. Super-human performance in gran turismo sport using deep reinforcement learning. *IEEE Robotics Autom. Lett.*, v. 6, n. 3, p. 4257–4264, 2021. Disponível em: <https://doi.org/10.1109/LRA.2021.3064284>.

GEIRHOS, R.; JACOBSEN, J.-H.; MICHAELIS, C.; ZEMEL, R.; BRENDEL, W.; BETHGE, M.; WICHMANN, F. A. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, Nature Publishing Group, v. 2, n. 11, p. 665–673, 2020.

GENT, I. P.; WALSH, T. Towards an understanding of hill-climbing procedures for sat. In: *AAAI*. [S.l.: s.n.], 1993. v. 93, n. Citeseer, p. 28–33.

GOLDBERG, D. E. Genetic algorithms in search. *Optimization, and MachineLearning*, Addison Wesley Publishing Co. Inc., 1989.

GOODFELLOW, I. J.; SHLENS, J.; SZEGEDY, C. Explaining and harnessing adversarial examples. In: BENGIO, Y.; LECUN, Y. (Ed.). *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. [s.n.], 2015. Disponível em: <http://arxiv.org/abs/1412.6572>.

GOOGLE Search Will Be Your Next Brain. <https://www.wired.com/2015/01/google-search-will-be-your-next-brain/>. [Accessed 20-Oct-2022].

HE, K.; ZHANG, X.; REN, S.; SUN, J. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. Disponível em: <http://arxiv.org/abs/1512.03385>.

HENDRYCKS, D.; DIETTERICH, T. G. Benchmarking neural network robustness to common corruptions and perturbations. In: *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. Disponível em: <https://openreview.net/forum?id=HJz6tiCqYm>.

HENDRYCKS, D.; ZHAO, K.; BASART, S.; STEINHARDT, J.; SONG, D. Natural adversarial examples. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2021. p. 15262–15271.

HOOS, H. H.; STÜTZLE, T. Stochastic local search algorithms: an overview. *Springer Handbook of Computational Intelligence*, Springer, p. 1085–1105, 2015.

IOFFE, S.; SZEGEDY, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: PMLR. *International conference on machine learning*. [S.l.], 2015. p. 448–456.

JIA, C.; YANG, Y.; XIA, Y.; CHEN, Y.-T.; PAREKH, Z.; PHAM, H.; LE, Q. V.; SUNG, Y.; LI, Z.; DUERIG, T. *Scaling Up Visual and Vision-Language Representation Learning With Noisy Text Supervision*. 2021.

JIA, C.; YANG, Y.; XIA, Y.; CHEN, Y.-T.; PAREKH, Z.; PHAM, H.; LE, Q. V.; SUNG, Y.; LI, Z.; DUERIG, T. *Scaling Up Visual and Vision-Language Representation Learning With Noisy Text Supervision*. 2021.

KENNEDY, J.; EBERHART, R. Particle swarm optimization. In: IEEE. *Proceedings of ICNN'95-international conference on neural networks*. [S.l.], 1995. v. 4, p. 1942–1948.

KHURANA, D.; KOLI, A.; KHATTER, K.; SINGH, S. Natural language processing: State of the art, current trends and challenges. *Multimedia tools and applications*, Springer, p. 1–32, 2022.

KINGMA, D. P.; BA, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

KOKHLIKYAN, N.; MIGLANI, V.; MARTIN, M.; WANG, E.; ALSALLAKH, B.; REYNOLDS, J.; MELNIKOV, A.; KLIUSHKINA, N.; ARAYA, C.; YAN, S.; REBLITZ-RICHARDSON, O. Captum: A unified and generic model interpretability library for pytorch. *CoRR*, abs/2009.07896, 2020. Disponível em: <https://arxiv.org/abs/2009.07896>.

KOLESNIKOV, A.; BEYER, L.; ZHAI, X.; PUIGCERVER, J.; YUNG, J.; GELLY, S.; HOULSBY, N. Big transfer (bit): General visual representation learning. In: SPRINGER. *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*. [S.l.], 2020. p. 491–507.

KRIZHEVSKY, A.; HINTON, G. et al. Learning multiple layers of features from tiny images. Toronto, ON, Canada, 2009.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, v. 25, 2012.

KURAKIN, A.; GOODFELLOW, I. J.; BENGIO, S. Adversarial machine learning at scale. In: *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. Disponível em: <https://openreview.net/forum?id=BJm4T4Kgx>.

LI, P.; LI, Y.; HSIEH, C.; ZHANG, S.; LIU, X.; LIU, H.; SONG, S.; YAO, X. Trimnet: learning molecular representation from triplet messages for biomedicine. *Briefings Bioinform.*, v. 22, n. 4, 2021. Disponível em: <https://doi.org/10.1093/bib/bbaa266>.

LIU, F.; AVCI, B. Incorporating priors with feature attribution on text classification. In: KORHONEN, A.; TRAUM, D. R.; MÀRQUEZ, L. (Ed.). *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*. Association for Computational Linguistics, 2019. p. 6274–6283. Disponível em: <https://doi.org/10.18653/v1/p19-1631>.

LUNDBERG, S. M.; LEE, S. A unified approach to interpreting model predictions. In: GUYON, I.; LUXBURG, U. von; BENGIO, S.; WALLACH, H. M.; FERGUS, R.; VISHWANATHAN, S. V. N.; GARNETT, R. (Ed.). *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*. [s.n.], 2017. p. 4765–4774. Disponível em: <https://proceedings.neurips.cc/paper/2017/hash/8a20a8621978632d76c43dfd28b67767-Abstract.html>.

MADRY, A.; MAKELOV, A.; SCHMIDT, L.; TSIPRAS, D.; VLADU, A. Towards deep learning models resistant to adversarial attacks. In: *International Conference on Learning Representations*. [S.l.: s.n.], 2018.

MAJI, S.; KANNALA, J.; RAHTU, E.; BLASCHKO, M.; VEDALDI, A. *Fine-Grained Visual Classification of Aircraft*. [S.l.], 2013.

MENON, S.; VONDRICK, C. Visual classification via description from large language models. In: *The Eleventh International Conference on Learning Representations*. [S.l.: s.n.], 2023.

MILLER, G. A. Wordnet: a lexical database for english. *Communications of the ACM*, ACM New York, NY, USA, v. 38, n. 11, p. 39–41, 1995.

MIOTTO, R.; WANG, F.; WANG, S.; JIANG, X.; DUDLEY, J. T. Deep learning for healthcare: review, opportunities and challenges. *Briefings in Bioinformatics*, v. 19, n. 6, p. 1236–1246, 05 2017. ISSN 1477-4054. Disponível em: <https://doi.org/10.1093/bib/bbx044>.

MOAYERI, M.; POPE, P.; BALAJI, Y.; FEIZI, S. A comprehensive study of image classification model sensitivity to foregrounds, backgrounds, and visual attributes. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*. IEEE, 2022. p. 19065–19075. Disponível em: <https://doi.org/10.1109/CVPR52688.2022.01850>.

MOHSENI, S. Toward design and evaluation framework for interpretable machine learning systems. In: *Proceedings of the 2019 AAAI/ACM conference on AI, ethics, and society*. [S.l.: s.n.], 2019. p. 553–554.

MONTAVON, G.; BINDER, A.; LAPUSCHKIN, S.; SAMEK, W.; MÜLLER, K.-R. Layer-wise relevance propagation: An overview. In: _____. *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. Cham: Springer International Publishing, 2019. p. 193–209. ISBN 978-3-030-28954-6. Disponível em: <https://doi.org/10.1007/978-3-030-28954-6_10>.

MONTAVON, G.; LAPUSCHKIN, S.; BINDER, A.; SAMEK, W.; MÜLLER, K.-R. Explaining nonlinear classification decisions with deep taylor decomposition. *Pattern Recognition*, Elsevier, v. 65, p. 211–222, 2017.

MURDOCH, W. J.; LIU, P. J.; YU, B. Beyond word importance: Contextual decomposition to extract interactions from lstms. In: *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. Disponível em: <https://openreview.net/forum?id=rkRwGg-0Z>.

NAIR, V.; HINTON, G. E. Rectified linear units improve restricted boltzmann machines. In: *Proceedings of the 27th international conference on machine learning (ICML-10)*. [S.l.: s.n.], 2010. p. 807–814.

NGUYEN, A.; YOSINSKI, J.; CLUNE, J. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2015. p. 427–436.

OPENAI. *GPT-4 Technical Report*. 2023.

OTTER, D. W.; MEDINA, J. R.; KALITA, J. K. A survey of the usages of deep learning for natural language processing. *IEEE Transactions on Neural Networks and Learning Systems*, v. 32, n. 2, p. 604–624, 2021.

PAN, Y.; YAO, T.; LI, Y.; MEI, T. X-linear attention networks for image captioning. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*. Computer Vision Foundation / IEEE, 2020. p. 10968–10977. Disponível em: <https://openaccess.thecvf.com/content_CVPR_2020/html/Pan_X-Linear_Attention_Networks_for_Image_Captioning_CVPR_2020_paper.html>.

PARK, D. S.; ZHANG, Y.; JIA, Y.; HAN, W.; CHIU, C.; LI, B.; WU, Y.; LE, Q. V. Improved noisy student training for automatic speech recognition. In: MENG, H.; XU, B.; ZHENG, T. F. (Ed.). *Interspeech 2020, 21st Annual Conference of the International Speech Communication Association, Virtual Event, Shanghai, China, 25-29 October 2020*. ISCA, 2020. p. 2817–2821. Disponível em: <https://doi.org/10.21437/Interspeech.2020-1470>.

PASZKE, A.; GROSS, S.; MASSA, F.; LERER, A.; BRADBURY, J.; CHANAN, G.; KILLEEN, T.; LIN, Z.; GIMELSHEIN, N.; ANTIGA, L. et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, v. 32, 2019.

PAUL, S.; CHEN, P.-Y. Vision transformers are robust learners. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. [S.l.: s.n.], 2022. v. 36, n. 2, p. 2071–2081.

PERONE, C. S.; CALABRESE, E.; COHEN-ADAD, J. Spinal cord gray matter segmentation using deep dilated convolutions. *Scientific reports*, Nature Publishing Group, v. 8, n. 1, p. 5966, 2018.

PORISKY, A.; BROSCH, T.; LJUNGBERG, E.; TANG, L. Y.; YOO, Y.; LEENER, B. D.; TRABOULSEE, A.; COHEN-ADAD, J.; TAM, R. Grey matter segmentation in spinal cord mris via 3d convolutional encoder networks with shortcut connections. In: *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*. [S.l.]: Springer, 2017. p. 330–337.

POWERED by AI: Instagram's Explore recommender system. <https://ai.facebook.com/blog/powered-by-ai-instagrams-explore-recommender-system/>. [Accessed 20-Oct-2022].

PRADOS, F.; ASHBURNER, J.; BLAIOTTA, C.; BROSCH, T.; CARBALLIDO-GAMIO, J.; CARDOSO, M. J.; CONRAD, B. N.; DATTA, E.; DÁVID, G.; LEENER, B. D. et al. Spinal cord grey matter segmentation challenge. *Neuroimage*, Elsevier, v. 152, p. 312–329, 2017.

RADFORD, A.; KIM, J. W.; HALLACY, C.; RAMESH, A.; GOH, G.; AGARWAL, S.; SASTRY, G.; ASKELL, A.; MISHKIN, P.; CLARK, J. et al. Learning transferable visual models from natural language supervision. In: PMLR. *International conference on machine learning*. [S.l.], 2021. p. 8748–8763.

RIBEIRO, M. T.; SINGH, S.; GUESTRIN, C. " 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*. [S.l.: s.n.], 2016. p. 1135–1144.

RIEGER, L.; SINGH, C.; MURDOCH, W. J.; YU, B. Interpretations are useful: Penalizing explanations to align neural networks with prior knowledge. In: *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*. PMLR, 2020. (Proceedings of Machine Learning Research, v. 119), p. 8116–8126. Disponível em: <http://proceedings.mlr.press/v119/rieger20a.html>.

RONNEBERGER, O.; FISCHER, P.; BROX, T. U-net: Convolutional networks for biomedical image segmentation. In: SPRINGER. *International Conference on Medical image computing and computer-assisted intervention*. [S.l.], 2015. p. 234–241.

ROSS, A. S.; HUGHES, M. C.; DOSHI-VELEZ, F. Right for the right reasons: Training differentiable models by constraining their explanations. In: SIERRA, C. (Ed.). *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*. ijcai.org, 2017. p. 2662–2670. Disponível em: <https://doi.org/10.24963/ijcai.2017/371>.

SANTOS, F. A.; SOUZA, M. D. de; OLIVEIRA, P.; MATOS, L. N.; NOVAIS, P.; ZANCHETTIN, C. Image classification understanding with model inspector tool. In: SPRINGER. *International Conference on Hybrid Artificial Intelligence Systems*. [S.l.], 2023. p. 611–622.

SANTOS, F. A. O.; SOUZA, M. D. de; ZANCHETTIN, C. Evaluating zero-shot image classification based on visual language model with relation to background shift. In: *Neural Information Processing Systems Conference: LatinX in AI (LXAI) Research Workshop 2023, New Orleans, USA*. [S.l.: s.n.], 2023.

SANTOS, F. A. O.; SOUZA, M. D. de; ZANCHETTIN, C. Towards background and foreground color robustness with adversarial right for the right reasons. In: SPRINGER. *International Conference on Artificial Neural Networks*. [S.l.], 2023. p. 169–180.

SANTOS, F. A. O.; ZANCHETTIN, C. Exploring image classification robustness and interpretability with right for the right reasons data augmentation. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*. [S.l.: s.n.], 2023. p. 4147–4156.

SANTOS, F. A. O.; ZANCHETTIN, C.; MATOS, L. N.; NOVAIS, P. Active image data augmentation. In: SPRINGER. *Hybrid Artificial Intelligent Systems: 14th International Conference, HAIS 2019, León, Spain, September 4–6, 2019, Proceedings 14*. [S.l.], 2019. p. 310–321.

SANTOS, F. A. O.; ZANCHETTIN, C.; MATOS, L. N.; NOVAIS, P. On the impact of interpretability methods in active image augmentation method. *Logic Journal of the IGPL*, Oxford University Press, v. 30, n. 4, p. 611–621, 2022.

SANTOS, F. A. O.; ZANCHETTIN, C.; SILVA, J. V. S.; MATOS, L. N.; NOVAIS, P. A hybrid post hoc interpretability approach for deep neural networks. In: SPRINGER. *International Conference on Hybrid Artificial Intelligence Systems*. [S.l.], 2021. p. 600–610.

SATTARZADEH, S.; SUDHAKAR, M.; LEM, A.; MEHRYAR, S.; PLATANIOTIS, K.; JANG, J.; KIM, H.; JEONG, Y.; LEE, S.; BAE, K. Explaining convolutional neural networks through attribution-based input sampling and block-wise feature aggregation. *ArXiv*, abs/2010.00672, 2020.

SCHRAMOWSKI, P.; STAMMER, W.; TESO, S.; BRUGGER, A.; HERBERT, F.; SHAO, X.; LUIGS, H.-G.; MAHLEIN, A.-K.; KERSTING, K. Making deep neural networks right for the right scientific reasons by interacting with their explanations. *Nature Machine Intelligence*, Nature Publishing Group, v. 2, n. 8, p. 476–486, 2020.

SELVARAJU, R. R.; COGSWELL, M.; DAS, A.; VEDANTAM, R.; PARIKH, D.; BATRA, D. Grad-cam: Visual explanations from deep networks via gradient-based localization. In: *Proceedings of the IEEE International Conference on Computer Vision*. [S.l.: s.n.], 2017. p. 618–626.

SHAFAHI, A.; NAJIBI, M.; GHIASI, A.; XU, Z.; DICKERSON, J. P.; STUDER, C.; DAVIS, L. S.; TAYLOR, G.; GOLDSTEIN, T. Adversarial training for free! In: WALLACH, H. M.; LAROCHELLE, H.; BEYGELZIMER, A.; D'ALCHÉ-BUC, F.; FOX, E. B.; GARNETT, R. (Ed.). *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*. [s.n.], 2019. p. 3353–3364. Disponível em: <https://proceedings.neurips.cc/paper/2019/hash/7503cfacd12053d309b6bed5c89de212-Abstract.html>.

SHAH, S.; FERNANDEZ, R.; DRUCKER, S. M. A system for real-time interactive analysis of deep learning training. In: *Proceedings of the ACM SIGCHI Symposium on Engineering Interactive Computing Systems, EICS 2019, Valencia, Spain, June 18-21, 2019*. [S.l.: s.n.], 2019. p. 16:1–16:6.

SHRIKUMAR, A.; GREENSIDE, P.; KUNDAJE, A. Learning important features through propagating activation differences. In: PRECUP, D.; TEH, Y. W. (Ed.). *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*. PMLR, 2017. (Proceedings of Machine Learning Research, v. 70), p. 3145–3153. Disponível em: <http://proceedings.mlr.press/v70/shrikumar17a.html>.

SIMON, M.; RODNER, E.; DENZLER, J. Part detector discovery in deep convolutional neural networks. In: SPRINGER. *Asian Conference on Computer Vision*. [S.l.], 2014. p. 162–177.

SIMONYAN, K.; VEDALDI, A.; ZISSERMAN, A. Deep inside convolutional networks: Visualising image classification models and saliency maps. In: BENGIO, Y.; LECUN, Y. (Ed.). *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Workshop Track Proceedings*. [s.n.], 2014. Disponível em: <http://arxiv.org/abs/1312.6034>.

SIMPSON, B.; DUTIL, F.; BENGIO, Y.; COHEN, J. P. Gradmask: Reduce overfitting by regularizing saliency. *arXiv preprint arXiv:1904.07478*, 2019.

SMILKOV, D.; THORAT, N.; KIM, B.; VIÉGAS, F.; WATTENBERG, M. Smoothgrad: removing noise by adding noise. *Workshop on Visualization for Deep Learning, ICML*, 2017.

SPRINGENBERG, J. T.; DOSOVITSKIY, A.; BROX, T.; RIEDMILLER, M. A. Striving for simplicity: The all convolutional net. In: BENGIO, Y.; LECUN, Y. (Ed.). *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA,*

USA, May 7-9, 2015, Workshop Track Proceedings. [s.n.], 2015. Disponível em: <http://arxiv.org/abs/1412.6806>.

SRIVASTAVA, N.; HINTON, G.; KRIZHEVSKY, A.; SUTSKEVER, I.; SALAKHUTDINOV, R. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, JMLR. org, v. 15, n. 1, p. 1929–1958, 2014.

STRUMBELJ, E.; KONONENKO, I. An efficient explanation of individual classifications using game theory. *The Journal of Machine Learning Research*, JMLR. org, v. 11, p. 1–18, 2010.

STURMFELS, P.; LUNDBERG, S.; LEE, S.-I. Visualizing the impact of feature attribution baselines. *Distill*, 2020. Https://distill.pub/2020/attribution-baselines.

SUDHAKAR, M.; SATTARZADEH, S.; PLATANIOTIS, K. N.; JANG, J.; JEONG, Y.; KIM, H. Ada-sise: Adaptive semantic input sampling for efficient explanation of convolutional neural networks. *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2021.

SUNDARARAJAN, M.; TALY, A.; YAN, Q. Axiomatic attribution for deep networks. In: PRECUP, D.; TEH, Y. W. (Ed.). *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*. PMLR, 2017. (Proceedings of Machine Learning Research, v. 70), p. 3319–3328. Disponível em: <http://proceedings.mlr.press/v70/sundararajan17a.html>.

SZEGEDY, C.; ZAREMBA, W.; SUTSKEVER, I.; BRUNA, J.; ERHAN, D.; GOODFELLOW, I.; FERGUS, R. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

TAHA, A. A.; HANBURY, A. Metrics for evaluating 3d medical image segmentation: analysis, selection, and tool. *BMC medical imaging*, BioMed Central, v. 15, n. 1, p. 1–28, 2015.

TAKASE, S.; KIYONO, S. Rethinking perturbations in encoder-decoders for fast training. In: TOUTANOVA, K.; RUMSHISKY, A.; ZETTLEMOYER, L.; HAKKANI-TÜR, D.; BELTAGY, I.; BETHARD, S.; COTTERELL, R.; CHAKRABORTY, T.; ZHOU, Y. (Ed.). *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*. Association for Computational Linguistics, 2021. p. 5767–5780. Disponível em: <https://doi.org/10.18653/v1/2021.naacl-main.460>.

TAN, M.; LE, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In: PMLR. *International conference on machine learning*. [S.l.], 2019. p. 6105–6114.

TIAN, H.; ZHU, T.; LIU, W.; ZHOU, W. Image fairness in deep learning: problems, models, and challenges. *Neural Computing and Applications*, Springer, v. 34, n. 15, p. 12875–12893, 2022.

TORRALBA, A. Contextual priming for object detection. *International journal of computer vision*, Springer, v. 53, p. 169–191, 2003.

TORRES, J. M. M.; MEDINA-DEVILLIERS, S.; CLARKSON, T.; LERNER, M. D.; RICCARDI, G. Evaluation of interpretability for deep learning algorithms in eeg emotion recognition: A case study in autism. *Artificial Intelligence in Medicine*, Elsevier, p. 102545, 2023.

TRAMÈR, F.; KURAKIN, A.; PAPERNOT, N.; GOODFELLOW, I. J.; BONEH, D.; MCDANIEL, P. D. Ensemble adversarial training: Attacks and defenses. In: *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. Disponível em: <https://openreview.net/forum?id=rkZvSe-RZ>.

VIVIANO, J. D.; SIMPSON, B.; DUTIL, F.; BENGIO, Y.; COHEN, J. P. Saliency is a possible red herring when diagnosing poor generalization. In: *9th International Conference on Learning Representations, ICLR 2021*. OpenReview.net, 2021. Disponível em: <https://openreview.net/forum?id=c9-WeM-ceB>.

XIAO, K. Y.; ENGSTROM, L.; ILYAS, A.; MADRY, A. Noise or signal: The role of image backgrounds in object recognition. In: *International Conference on Learning Representations*. [S.l.: s.n.].

XIAO, K. Y.; ENGSTROM, L.; ILYAS, A.; MADRY, A. Noise or signal: The role of image backgrounds in object recognition. In: *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. Disponível em: <https://openreview.net/forum?id=gl3D-xY7wLq>.

XU, S.; VENUGOPALAN, S.; SUNDARARAJAN, M. Attribution in scale and space. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*. IEEE, 2020. p. 9677–9686. Disponível em: <https://doi.org/10.1109/CVPR42600.2020.00970>.

YANG, J.; WANG, P.; ZOU, D.; ZHOU, Z.; DING, K.; PENG, W.; WANG, H.; CHEN, G.; LI, B.; SUN, Y.; DU, X.; ZHOU, K.; ZHANG, W.; HENDRYCKS, D.; LI, Y.; LIU, Z. Openood: Benchmarking generalized out-of-distribution detection. In: KOYEJO, S.; MOHAMED, S.; AGARWAL, A.; BELGRAVE, D.; CHO, K.; OH, A. (Ed.). *Advances in Neural Information Processing Systems*. [S.l.]: Curran Associates, Inc., 2022. v. 35, p. 32598–32611.

ZEILER, M. D.; FERGUS, R. Visualizing and understanding convolutional networks. In: SPRINGER. *European conference on computer vision*. [S.l.], 2014. p. 818–833.

ZHAI, X.; KOLESNIKOV, A.; HOULSBY, N.; BEYER, L. Scaling vision transformers. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2022. p. 12104–12113.

ZHANG, C.; BENGIO, S.; HARDT, M.; RECHT, B.; VINYALS, O. Understanding deep learning requires rethinking generalization. In: *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. Disponível em: <https://openreview.net/forum?id=Sy8gdB9xx>.

ZHANG, T.; ZHU, Z. Interpreting adversarially trained convolutional neural networks. In: CHAUDHURI, K.; SALAKHUTDINOV, R. (Ed.). *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*. PMLR, 2019. (Proceedings of Machine Learning Research, v. 97), p. 7502–7511. Disponível em: <http://proceedings.mlr.press/v97/zhang19s.html>.

ZHANG, Y.; TIŇO, P.; LEONARDIS, A.; TANG, K. A survey on neural network interpretability. *IEEE Transactions on Emerging Topics in Computational Intelligence*, IEEE, 2021.

ZHU, Y.; PANG, L.; LAN, Y.; SHEN, H.; CHENG, X. Adaptive information seeking for open-domain question answering. In: MOENS, M.; HUANG, X.; SPECIA, L.; YIH, S. W. (Ed.). *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*. Association for Computational Linguistics, 2021. p. 3615–3626. Disponível em: <https://aclanthology.org/2021.emnlp-main.293>.

APPENDIX A – APPENDIX FOR CHAPTER 6

INTERPRETABILITY RESULTS

We performed an interpretability analysis in the manuscript's main text to understand how the ChatGPT+CLIP and ChatGPT+ALIGN model attributes the similarity score. Given an input image x and its background variations, we compute the similarity for each category description (i.e. $\phi(d, x)$) and build a panel for each category and its descriptions. This analysis is important because it enables us to understand how ChatGPT+CLIP and ChatGPT+ALIGN attributes the similarity scores in each situation.

Figures 36, 35, 37, 38, show the complete results for both datasets (ImageNet-9 and RIVAL10) and models (ChatGPT+CLIP and ChatGPT+ALIGN). The results are coherent with the manuscript's main text and show that the ChatGPT+CLIP attributes higher similarity scores to images from the original distribution. These results raise important questions about the CLIP correlates the image information with text information; as the same object information is present in all the challenges and the descriptions are about the object, why does CLIP decrease the similarity when we change the background? Addressing this question is out of the scope of this thesis, and it can be investigated in future works.

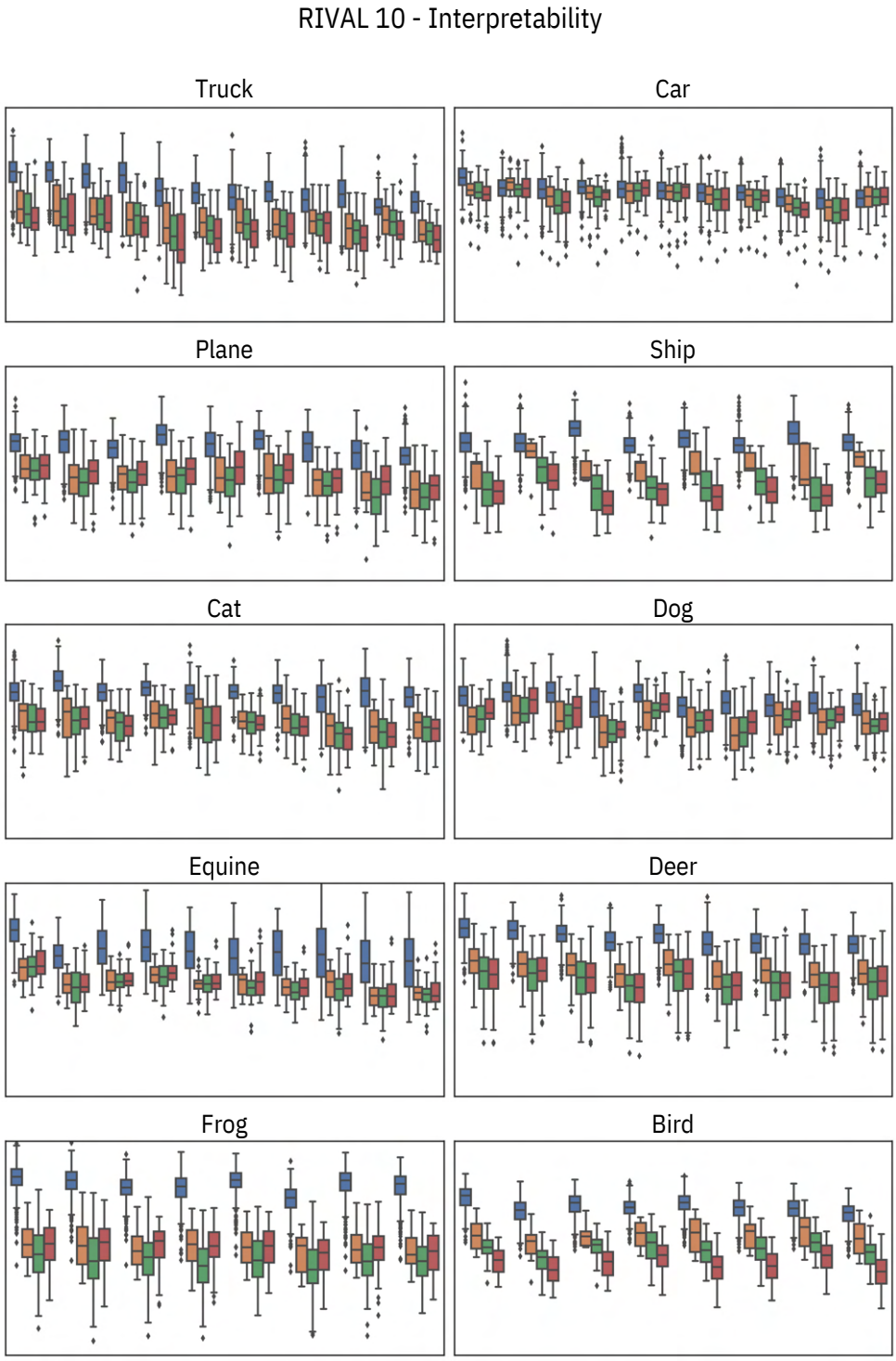

Figure 35 – **Interpretability analysis for all classes in RIVAL 10 dataset for ChatGPT+CLIP.** Due to paper size limitations, the main text of the manuscript showed the similarity scores distributions for two categories of each dataset. However, here we show all results for all categories. The box-plot colors follows the same pattern of the main text plot, which the blue color represents the original challenge, orange the mixed-same, green the mixed-rand and red the mixed-next.

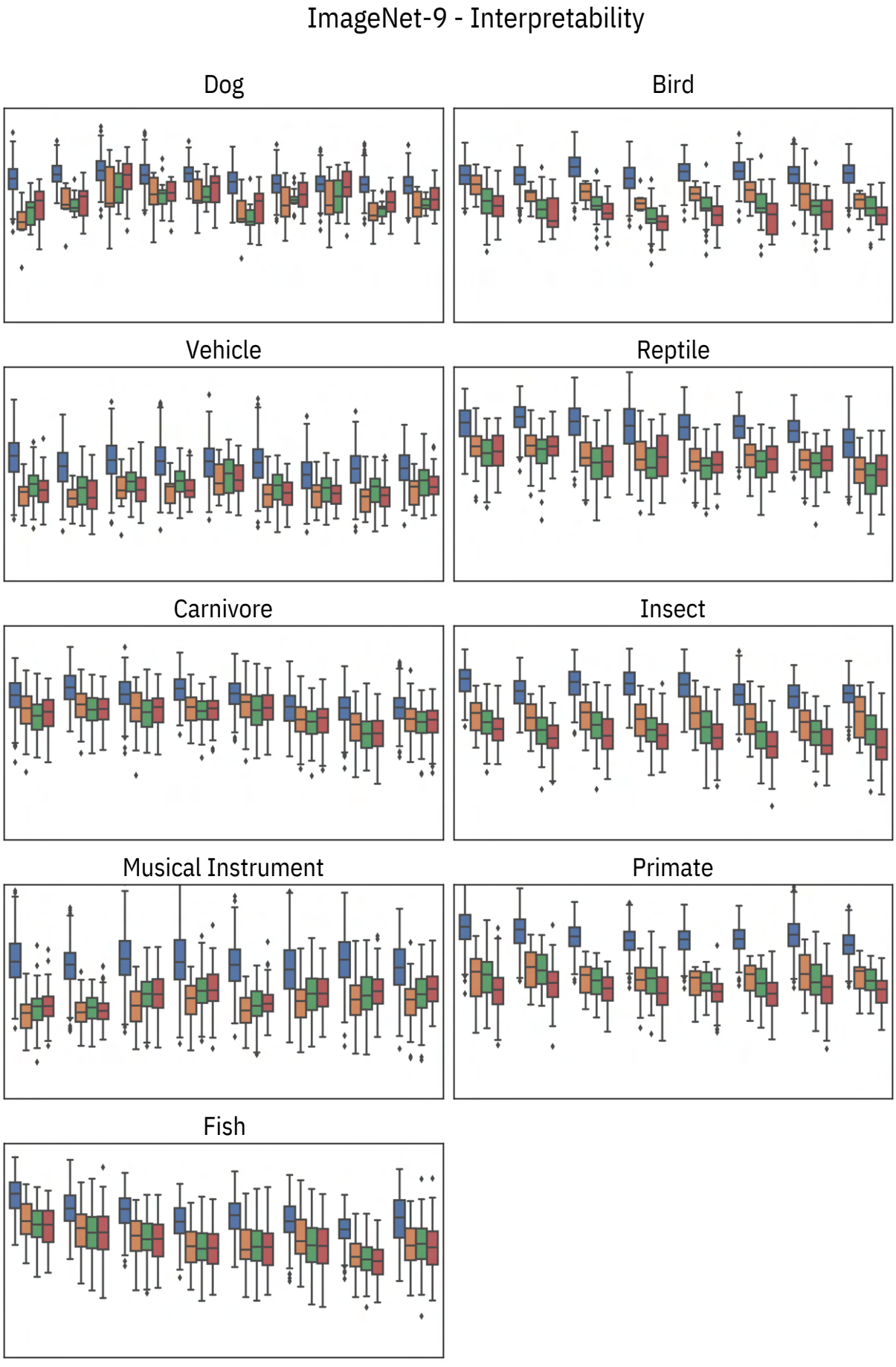

Figure 36 – **Interpretability analysis for all classes in ImageNet-9 dataset for ChatGPT+CLIP.**

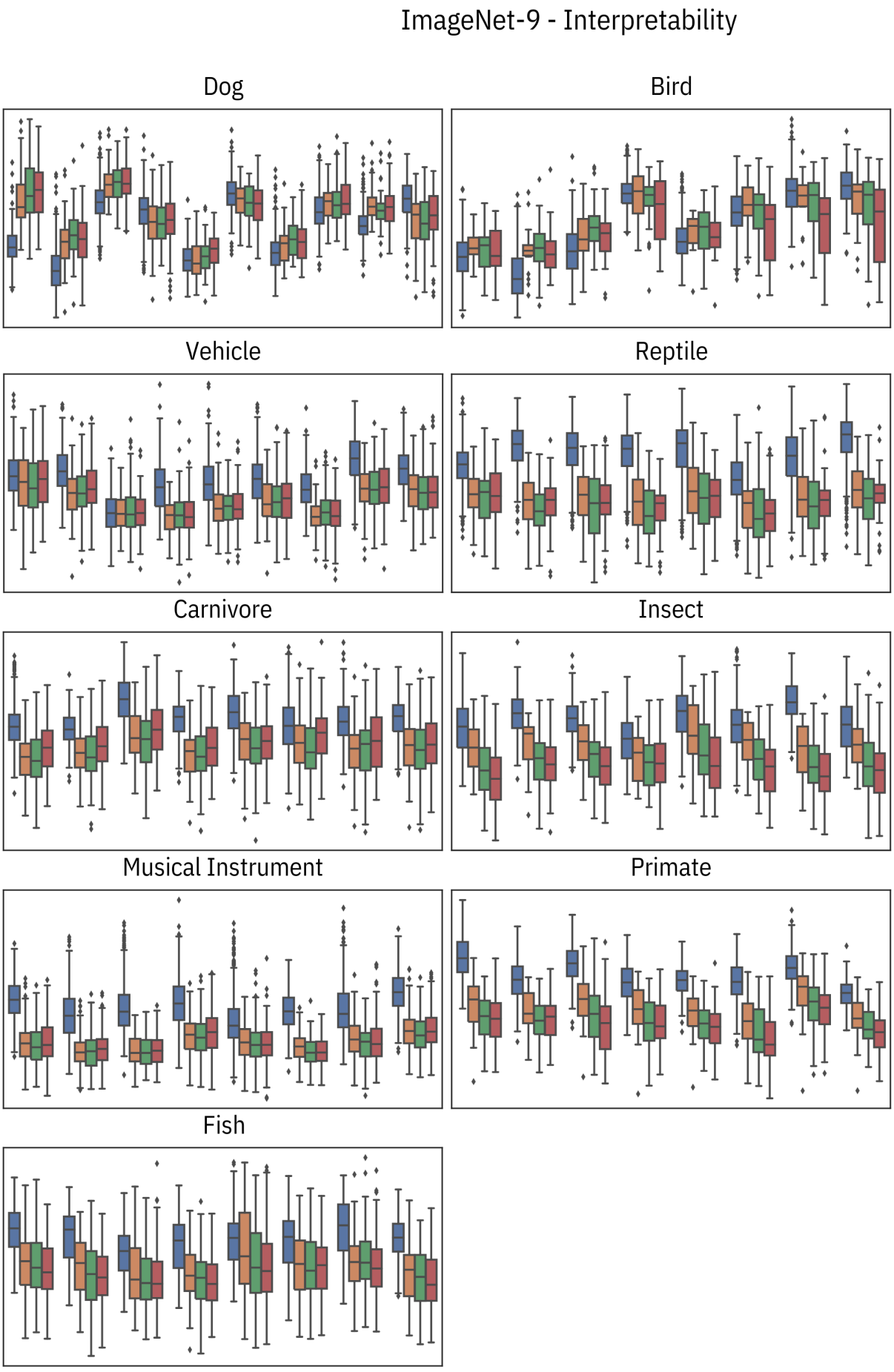

Figure 37 – **Interpretability analysis for all classes in ImageNet-9 dataset for ChatGPT+ALIGN.**

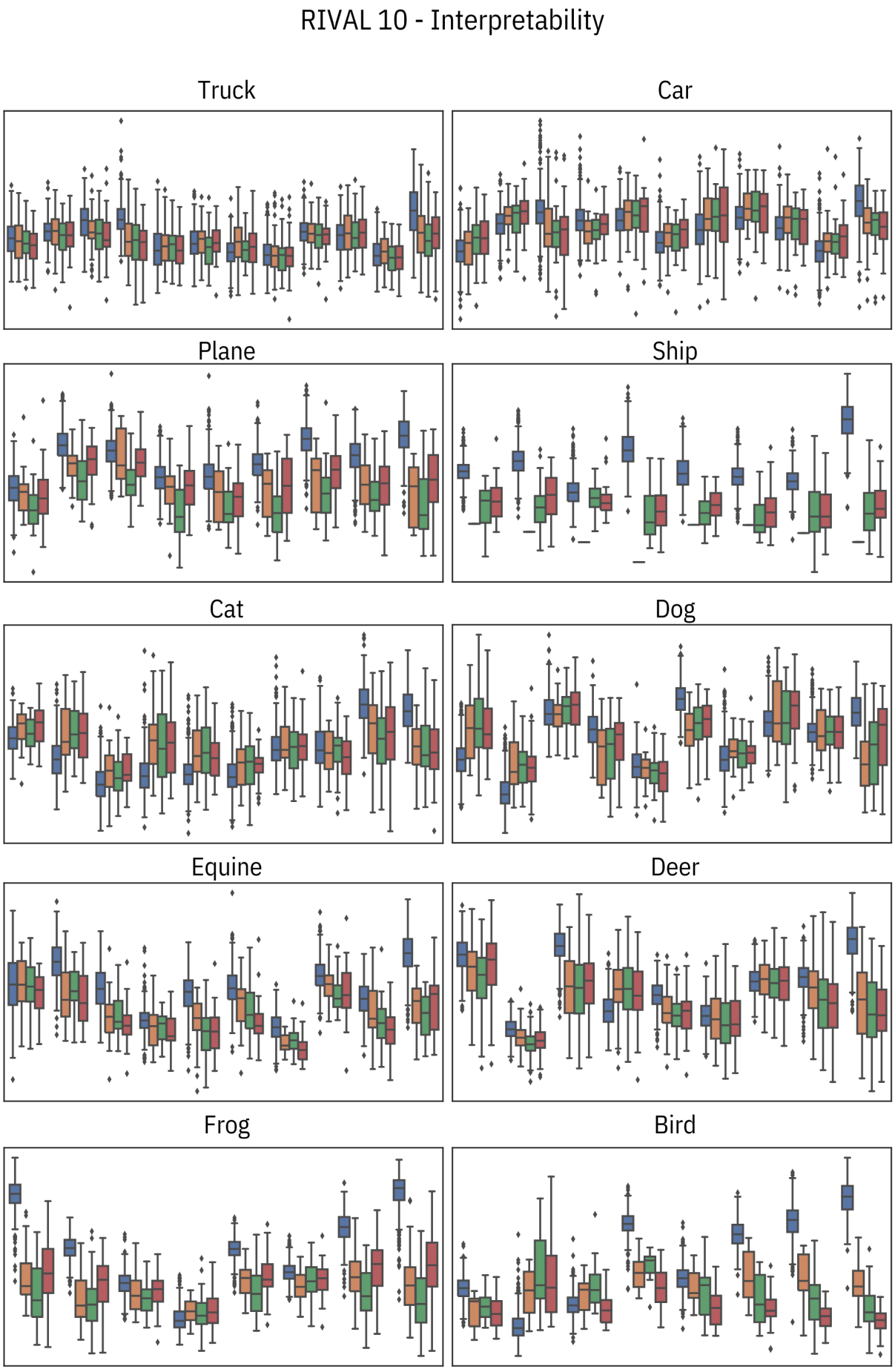

Figure 38 – **Interpretability analysis for all classes in RIVAL-10 dataset for ChatGPT+ALIGN.**