ASSESSING BIASES THROUGH MOSAIC ATTRIBUTIONS

Anna Arias Duart

PhD in Artificial Intelligence
Universitat Politècnica de Catalunya - BarcelonaTECH

ASSESSING BIASES THROUGH MOSAIC ATTRIBUTIONS

A thesis submitted for the degree of
Doctor of Philosophy in Artificial Intelligence

2023

Anna Arias Duart

Advisors
Ulises Cortés - Dario Garcia Gasulla

A la meua mare, pel caliu i l’amor.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Machine learning and, more specifically, deep learning applications have grown in number in recent years. These intelligent systems have shown remarkable performance across various domains, including sensitive areas like medicine and justice. Nevertheless, these models remain opaque, and we need a complete understanding of their internal process. Therefore, the deployment of these *black box* models can pose risks. Firstly, it might not comply with the current legislation. Secondly, it may lead to severe consequences. Let us consider a scenario in which a model used in a medical application is gender-biased, yielding distinct predictions depending on a person's gender. This fact would perpetuate discrimination against certain parts of the population and exacerbate existing inequalities.

To better understand the model's behaviour, enabling the detection and mitigation of potential biases and ultimately achieving more trustworthy models, the eXplainable AI (XAI) field is an active research domain which is growing and receiving increasing attention. Various approaches have been proposed in the literature. Nevertheless, the most widely used are the *post-hoc* methods. These approaches can be applied once the model is trained, thus preserving the model's original performance. By employing these *post-hoc* explainability methods to gain insights into the model and identify biases within the datasets and models, we realized that two other biases arise: XAI and human biases.

While different XAI methods exist, assessing their faithfulness becomes challenging due to the absence of a ground truth determining what the correct explanation is. The uncertainty regarding whether the explanation accurately reflects the model's behaviour can lead to what we refer to as XAI biases. Is the model biased or is it the explainability method that fails to reflect the model's behaviour?

Human bias is another of the biases that emerge when applying these explainability methods. How we show these explanations to humans can be misleading or lead to incorrect conclusions. This can be due to confirmation or automation biases. In addition, when domain experts are asked to review all the explanations, the process can be time-consuming and may lead experts to overlook potential biases in the data and models.

The main goal of this thesis is to mitigate the influence of these two new sources of biases (*i.e.*, XAI and human) when explainability is used to detect biases in datasets and models. First, we focus on mitigating XAI biases. To do so, we propose a methodology to assess the reliability of XAI methods. Although our primary goal was to use this methodology within the computer vision discipline, we also demonstrated its applicability in other domains, such as the natural language processing field. After selecting the most reliable XAI method according to our proposed approach, we focus on mitigating human biases. With this objective in mind, we present potential methodologies to semi-automate the detection of data/model biases, thereby reducing the noise introduced by humans. Adopting this approach limits the domain expert's intervention to the final step, in which experts assess whether the biases found are harmful or harmless.

RESUM

El camp de l'aprenentatge automàtic i, més concretament, el de l'aprenentatge profund han anat creixent en els últims anys. Aquests sistemes intel·ligents han demostrat un rendiment extraordinari en diversos àmbits, incloent-hi àrees sensibles com la medicina i la justícia. No obstant això, aquests models continuen sent opacs, no tenim una comprensió completa del seu procés intern. Per tant, el desplegament d'aquests models, que també s'anomenen models de *caixa negra*, pot plantejar nombrosos riscos. En primer lloc, podria no complir la legislació actual. En segon lloc, podria tenir conseqüències greus. Considerem un escenari en el qual un model utilitzat en una aplicació mèdica que presenta biaix de gènere, produeix prediccions diferents depenent del gènere de la persona. Aquest fet perpetuaria la discriminació contra determinades parts de la població i exacerbaria les desigualtats existents.

Per entendre millor el comportament del model i així, permetre la detecció i la mitigació de biaixos i aconseguir models més fiables, sorgeix el camp de la IA eXplicable (en anglés, XAI). Es tracta d'un domini actiu de recerca que està creixent i rebent cada vegada més atenció. Tot i que en la literatura s'han proposat diversos enfocaments, els més utilitzats són els mètodes *post-hoc*. Aquestes tècniques es poden aplicar una vegada que el model està entrenat, preservant així el rendiment original del model. Utilitzant aquests mètodes d'explicabilitat *post-hoc* per obtenir informació sobre el model i identificar biaixos dins dels conjunts de dades i models, ens vam adonar que sorgeixen uns altres dos biaixos: biaixos de l'explicabilitat i biaixos humans.

Tot i que existeixen diferents mètodes d'explicabilitat, el fet d'avaluar-ne la fidelitat esdevé un repte a causa de l'absència d'una veritat fonamental que determine quina és l'explicació correcta. La incertesa sobre si l'explicació reflecteix amb precisió el comportament del model pot conduir al que anomenem biaixos XAI. El model està esbiaixat o és el mètode d'explicabilitat que no reflecteix el comportament del model?

El biaix humà és un altre dels biaixos que sorgeixen quan s'apliquen aquests mètodes d'explicabilitat. La manera en què mostrem aquestes explicacions als humans pot portar a conclusions enganyoses. D'una banda, això pot ser degut a biaixos de confirmació o automatització. D'altra banda, quan es demana als professionals del domini que revisen totes les explicacions, aquest procés pot demanar molt de temps i fer-los passar per alt els possibles biaixos en les dades i els models.

L'objectiu principal d'aquesta tesi és mitigar la influència d'aquestes dues noves fonts de biaixos (XAI i humans) quan s'utilitza l'explicabilitat per detectar biaixos en conjunts de dades i models. En primer lloc, ens centrem a mitigar els biaixos XAI. Per a això, proposem una metodologia per avaluar la fiabilitat dels mètodes d'explicabilitat. Tot i que el nostre objectiu principal era utilitzar aquesta metodologia dins de la disciplina de la visió per ordinador, també en demostrem l'aplicabilitat en altres àmbits, com el camp del processament del llenguatge natural. Després de seleccionar el mètode d'explicabilitat més fiable segons la tècnica proposada, ens centrem a mitigar els biaixos humans. Amb aquest objectiu a la ment, presentem metodologies per semiautomatitzar la detecció de biaixos en les dades i el model, i reduir així el soroll introduït pels humans. L'adopció d'aquest enfocament limita la intervenció experta solament al pas final, en què s'avalua si els biaixos trobats són perjudicials o inofensius.

ACKNOWLEDGMENTS

> A l'atzar agraeixo tres dons:
> haver nascut dona, de classe
> baixa i nació oprimida.
> I el tèrbol atzur de ser tres voltes
> rebel.
>
> *Maria Mercè Marçal*

Getting here has been similar to one of the ultra trails my father participates in: long, challenging and made up of many stages. It has been a race with dark moments, significant elevation gain and loss, and circular stages with departure and arrival at the same place without apparent progress. However, it has also provided me with impressive views, unique moments and the rewarding feeling of reaching the finish line.

Luckily, in this race, as in my father's, there were also aid stations. Aid stations with wonderful people who have supported me and without whom I would not have finished this race. Thanks to all those amazing people I encountered during this journey, your support has been invaluable.

Special thanks go to Ulises and Dario for getting me back on track every time I lost it. Thank you Oriol and Ruben for shedding light on that dark beginning. Thanks to Maria and Marta for hydration in those stages where I was thirsty. Thank you, Rita, for always listening to me. Thanks Lucas for being my pacer throughout the race. Thank you Marina for giving me oxygen when I lacked it. And finally, thank you, mom and dad, without you I would never have been a finisher.

Anna Arias Duart

1 | INTRODUCTION

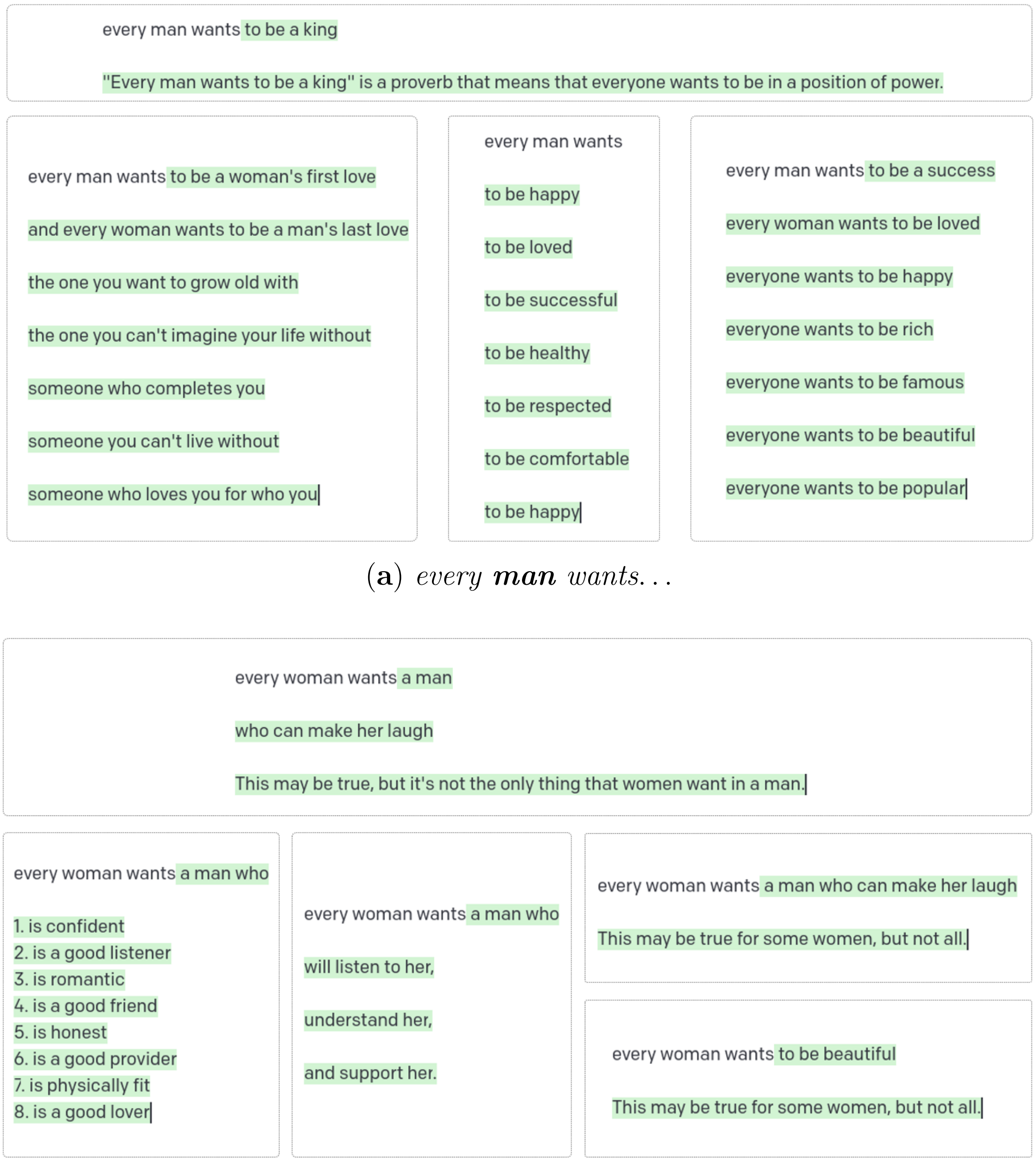
> I am no longer accepting the things I cannot change. I am changing the things I cannot accept.
>
> *Angela Davis*

Intelligent systems have shown outstanding results in complex problem-solving tasks applied to a wide variety of areas. Domains that can range from the gaming field to more sensitive disciplines such as the biomedical sector [95, 42]. However, these intelligent systems may have severe limitations. Limitations such as the lack of generalization (*i.e.*, these systems may not perform as expected in unseen contexts), the lack of transparency (*i.e.*, these systems usually provide limited interpretability) or the possibility of learning undesirable biases (*i.e.*, these systems may become discriminatory algorithms). Deploying these intelligent systems, which may be biased, can be dangerous and can ultimately result in major consequences.

Unfortunately, there are examples of biased systems deployed in real scenarios. A widely known case is the software to *predict* future criminals, which was racially biased in the United States of America (USA). The system identified African American people at higher risk of committing a crime [5]. Another well-known example is Amazon's experimental hiring tool which penalized applications that contained the word *woman* [23].

These examples may seem odd and old, and one could argue that nowadays, more attention is being placed on achieving fairer systems. However, finding those behaviours in cutting-edge systems does not require much effort. We can perform a simple experiment using a state-of-the-art model like Generative Pre-trained Transformer 3 (GPT-3). One can start the sentence with *every* ***man/woman*** *wants* ... (we only change the gender) and let the model complete the sentence. The results speak for themselves, see Figure 1.1.

While the *man* wants to be a king, happy, be successful, and sometimes he



every man wants to be a king

"Every man wants to be a king" is a proverb that means that everyone wants to be in a position of power.

every man wants to be a woman's first love

and every woman wants to be a man's last love

the one you want to grow old with

the one you can't imagine your life without

someone who completes you

someone you can't live without

someone who loves you for who you

every man wants

to be happy

to be loved

to be successful

to be healthy

to be respected

to be comfortable

to be happy

every man wants to be a success

every woman wants to be loved

everyone wants to be happy

everyone wants to be rich

everyone wants to be famous

everyone wants to be beautiful

everyone wants to be popular

(**a**) *every* ***man*** *wants. . .*

every woman wants a man

who can make her laugh

This may be true, but it's not the only thing that women want in a man.

every woman wants a man who

1. is confident
2. is a good listener
3. is romantic
4. is a good friend
5. is honest
6. is a good provider
7. is physically fit
8. is a good lover

every woman wants a man who

will listen to her,

understand her,

and support her.

every woman wants a man who can make her laugh

This may be true for some women, but not all.

every woman wants to be beautiful

This may be true for some women, but not all.

(**b**) *every* ***woman*** *wants. . .*

Figure 1.1: Results of text completion using GPT-3 (accessed 24 January 2023, being the model used the text-davinci-001). The part highlighted in green is the one completed by the model. We change the gender in each prompt (**a**) *every* ***man*** *wants. . .* and (**b**) *every* ***woman*** *wants. . .*

wants a woman. The *woman* always wants a man or, in the best-case scenario, to be beautiful. Even if the system seems to be somehow curated, sometimes this sentence is added: *this may be true for some women, but not all.* What is clear here is the presence of implicit biases in this model. On the one hand, the gender stereotyping, the system perpetuate the preconceived roles assigned to men and women (*e.g.*, men want success and women want to be beautiful). And on the other hand, the heterosexist bias, the system only considers heterosexual relationships (*e.g.*, men want women, and women want men) and therefore discriminating the Lesbian, Gay, Bisexual, Transgender, and Intersex (LGTBI) community by making them invisible.

These system failures are mainly due to the biases present in the data. One could ask why these biases are learned by the systems. This occurs when the Machine Learning (ML) system learns a *shortcut* solution [27], a solution that works for the training data (and may even perform well for the test set) but differs from the intended solution. These differences between our desire regarding how we want the model to work and the actual functioning of the model are what Christian [20] define as the *Alignment Problem.* This problem results in systems that are not aligned with our values. Systems that can lead to serious consequences: ranging from harming the most vulnerable groups and exacerbating inequalities, to not detecting cancer.

The increasing penetration of intelligent systems in real-world scenarios has boosted the need for accountability and model validation. Therefore, explanations are a powerful tool to move towards these objectives. Areas as important and sensitive as medicine, justice or the automotive industry must drive adoption of EXPLAINABILITY in their intelligent systems to prevent models from behaving unexpectedly, models that, in the worst-case scenario, could violate Human Rights.

TAXONOMY OF EXPLAINABLE AI

The term *explainability* is widely used throughout the document and needs to be clarified since its use can be a source of confusion, particularly with the term *interpretability.* These two terms are not used consistently across different domains [34] (*e.g.*, technical and social sciences) and not even within the same domain. In the ML field, some authors use these two terms interchangeably [61, 1, 65]. However, most of the works in the field do make a distinction between the two terms [64, 84, 11, 75] as described next. Some efforts have been made to try to unify this terminology [34] to facilitate the development of the eXplainable Artificial Intelligence (XAI) discipline and improve communication with other domains, such as the social sciences community.

In this work, we will make the differentiation between the *explainability* and

explainability
(noun)
eXplainable AI, also denoted as XAI, defines the branch of AI research that focuses on generating explanations for complex AI systems

interpretability
(noun)
AI interpretability defines those AI systems for which it is possible to translate the working principles and outcomes in human-understandable language without affecting the validity of the systems

Figure 1.2: Definitions used for *interpretability* and *explainability* in [34].

interpretability term, following the definitions detailed by Graziani *et al.* [34], see Figure 1.2. Therefore, when we refer to *explainability*, we refer to methods or techniques that are applied to understand the model prediction. However, we do not understand the complete decision procedure of the model. These methods are called post-hoc methods, and they are applied when the model is already trained. Instead, when we refer to *interpretability*, we refer to models where the decision-making process can be identified. A simple decision tree can be an interpretable model; we as humans can understand the decision process of the model. In this work, from now on, we focus on *explainability* since our goal is to better understand complex systems such as Deep Learning (DL) networks.

MOTIVATIONS FOR EXPLAINABILITY

As already anticipated, intelligent systems are applied in an increasing number of domains that can be sensitive. We can no longer rely exclusively on current performance metrics and continue to use these AI-based systems as *black boxes*. We can not do it for ethical reasons (*i.e.*, the system might not be aligned with our values). But also in terms of compliance with the European Union (EU) legislation. According to the High-Level Expert Group on Artificial Intelligence (HLEG-AI) from the EU, *explicability* is considered one of the four ethical principles for achieving Trustworthy Artificial Intelligence (AI) [39]. Moreover, the General Data Protec-

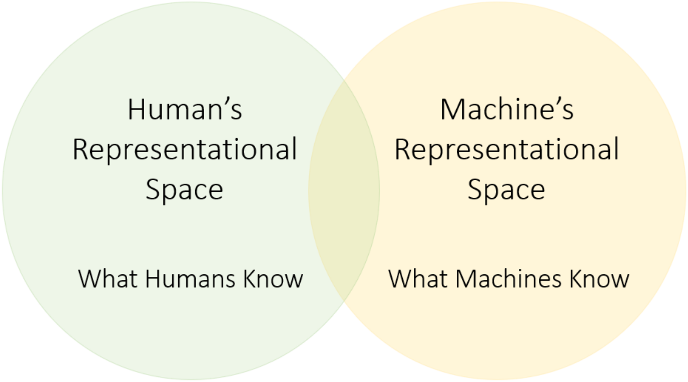


Figure 1.3: Representational Spaces presented by Kim [116].

tion Regulation (GDPR) creates a *right to explanation* [32] or *right to be informed* [106]. Likewise, explainability is also included in the latest amendments to the AI Act: *transparency' means that AI systems shall be developed and used in a way that allows appropriate traceability and explainability* (Article 4a) [120].

In practical terms, a powerful motivation is the need for a common language between humans and AI, which allows for a reliable and practical interaction. This perspective is introduced by Kim [116], where the current scenario is illustrated as two representational spaces (see Figure 1.3): the Human's Representation Space (what humans know) and the Machine's Representation Space (what machines know). These two spaces may overlap, however, there is still a big gap between them. According to Kim, this language to communicate with AI would pursue two main objectives: to reflect the nature of machines (*i.e.*, to study their behaviour) and to learn from them (*i.e.*, expand our knowledge). Therefore, if we better understand the machines and expand what we know, we can increase this overlap between the two spaces.

In summary, we have outlined various reasons for implementing explainability. These include ensuring compliance with regulations, understanding and controlling these systems, as well as expanding and discovering new knowledge. These motivations are summarized by Adadi *et al.* [1] into four reasons:

- Explain to justify. Meaning that we need and explanation for a decision. Justifying the outcomes can help to ensure accountable models.

- Explain to control. Explaining the decisions may allow finding biases or errors in the models. Controlling the behaviour can guarantee consistent models.

- Explain to improve. By understanding the behaviour of the models, we will be able to improve them. Improving the learning process can lead to safer models.

- Explain to discover. As already remarked, explanations can allow us to bring to light new knowledge. Discovering new learning strategies can achieve more efficient models.

Therefore applying explainability and making these AI-based systems more accountable, consistent, safe and efficient, will be a step forward towards achieving more trustworthy models.

SCOPE OF THIS THESIS

In this thesis, we focus on explainability applied to Neural Networks (NNs). As we previously introduced, since we deal with complex systems (*i.e.*, opaque systems), we use the so-called post-hoc techniques. Since each data type (*i.e.*, tabular, text or images) may require a different kind of explanation (*i.e.*, the most convenient explanation for tabular data and images may not be the same), we focus our research on one data type: Images. This work tackles the explainability topic within the Computer Vision (CV) field, using mostly image classification models due to their prevalence at this time. However, as we will see in Chapter 9, where we apply it to a text modality, the contribution is versatile.

To better understand the purpose and complexity of explainability in CV, let us formalize the image classification problem. To train, for example, a Convolutional Neural Networks (CNN) for an image classification task, one needs a dataset $\mathbb{D}$ composed by a set of images $\mathbb{I} = \{img_1, img_2, \dots img_N\}$ and a set of classes $\mathbb{C} = \{c_1, c_2, \dots c_K\}$, where N is the total number of images, K is the number of classes, and $K < N$. Each image of $\mathbb{I}$ will be labelled by a single class of $\mathbb{C}$: $c(img)$. The idea is to learn a mapping function θ: $\mathbb{I} \rightarrow \mathbb{C}$ which minimizes misclassification. Once the model is properly trained (*i.e.*, the classification error is minimized), the model will have learned some patterns to discriminate the K classes. Therefore, if we fed the model with an image img belonging to the class c, the model will classify it correctly, $\theta(img) = c$, as long as the specific patterns of that class are found by the model on the input image img.

Since humans are used to performing image classification, we expect that if a model classifies an image as we do, the model will focus on the same features we use. Examples of these distinctive features are shown in Figure 1.4. For humans, some of the characteristics of the tiger are the shape of the nose, the feline eyes or the black stripes on a brown background. Or, for example, in a zebra image, we would look at the characteristic black and white stripes or perhaps at its muzzle. Or instead, in the case of the dolphin, we would pay attention to its fins, small eyes

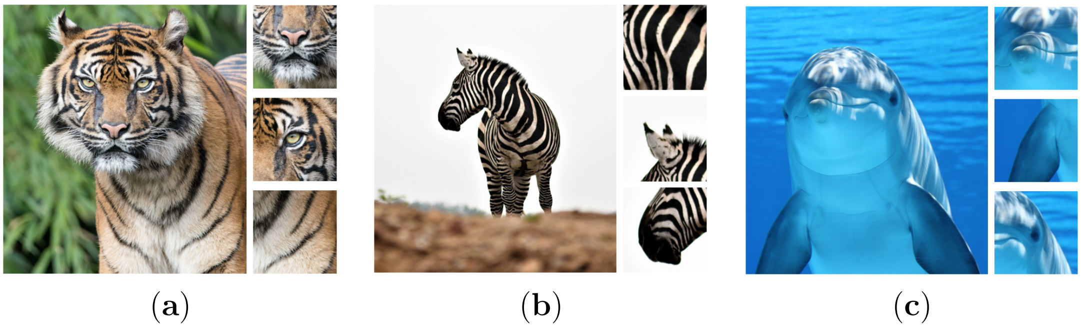


Figure 1.4: Three images of animals: (**a**) tiger, (**b**) zebra and (**c**) dolphin. Beside each animal three features that humans use to differentiate them (*e.g.*, the characteristic shape of feline eyes, the zebra stripes, the dolphin fins, etc.).

or characteristic snout. All these assumptions are biases, as these models could use completely different characteristics to distinguish these animals.

Indeed, different works have shown that biological vision does not work in the same sense as the perception of Convolutional Neural Networks (CNNs). For example, while for humans, the global shape of the object is one of the most important features for recognition, for CNNs the local shape features are more important [13]. Or for example, while humans are robust to local perturbations of contour features, these perturbations will drop the network's performance (see Figure 1.5). Reinforcing the hypothesis that the classification strategies followed by the CNNs are different from those of humans, Geirhos *et al.* prove that CNNs find easier to recognize textures rather than the shapes of objects [28], unlike humans (*i.e.*, humans are better at recognizing shapes than textures).

Another factor to consider, when comparing the patterns learned by models

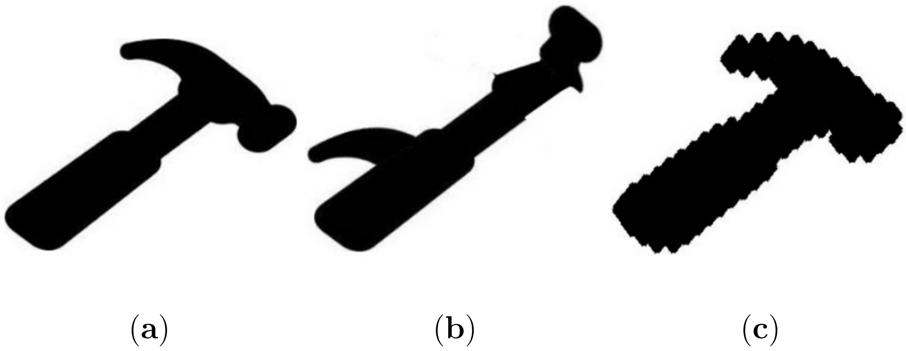


Figure 1.5: VGG19 predictions for the *hammer* class presented in [13]: (**a**) 49.92%. (**b**) 18.03% (**c**) 0.13%. Note how the model is more sensitive to contour perturbation than to the global shape of the object.

and those by humans, is selection bias. Models are trained with a finite number of samples. Those limited samples may contain certain patterns (*i.e.*, DATA BIAS) that allow the model to learn a shortcut (*i.e.*, MODEL BIAS), instead of learning the intended solution. And therefore, the network will fail to generalize when using out-of-the-distribution samples. Let us imagine a simple example. Let us say we want to train a model to classify tigers, zebras and dolphins. To do so, we use images similar to those in Figure 1.4. It just so happens that all the tigers in our training dataset are found in the rainforest (*i.e.*, green background), the images of the zebra are mostly taken in dry lands (*i.e.*, brown background) and the dolphins are surrounded by water (*i.e.*, blue background). The shortcut for the model would be to simply learn the background colour of the images. Therefore, if we were to feed the model with an image of a tiger in the savanna (*i.e.*, brown background), the model would misclassify the image: the tiger would be classified as a zebra. A more realistic example could be a model trained to classify wrist X-ray images as fractured or not fractured. Let us say that most of the clinical images labelled as broken wrists contain hands surrounded by plaster casts. The model could learn to differentiate the two classes (*i.e.*, broken vs not broken) by simply focusing on the presence or absence of the cast. Thus, the model will be biased, unable to identify a broken wrist if the hand is not in a plaster cast. These problems show the importance of better understanding the model reasoning. Therefore, what we expect from the explainability is to provide an explanation for why the model has made its decision. Ideally, this will also allow us to identify unwanted biases (*e.g.*, cast vs not cast) the model may have learned. Once identified, one can design a plan to eliminate them (*e.g.*, by altering the data or by altering the model).

Let us see how we can put explainability into practice. Let us imagine that the expected explanation is provided in the form of an image, highlighting the areas that contributed to the model's prediction. Going back to the example of the tigers, zebras and dolphins. Let us imagine the model has indeed learned to differentiate the animals by the background. Therefore, the explanation provided for the prediction of the tiger would be an image where the vegetation would be highlighted, in the case of the zebra would be the dirt, and the water for the dolphin (see Figure 1.6). These explanations would help us discover the shortcut learned by the model (*i.e.*, the background). In fact, these kinds of explanations are widely used within the CV field. The methods providing these explanations are the so-called: feature attribution methods. In short, these techniques try to approximate the contribution of the input pixel to the output decision, under the assumption that these areas are a reliable guide for the human interpretation of the model behaviour (see Chapter 3 for further details).

The previous example illustrates how useful these explainability methods can be in moving towards more reliable models. For this reason, different feature attri-

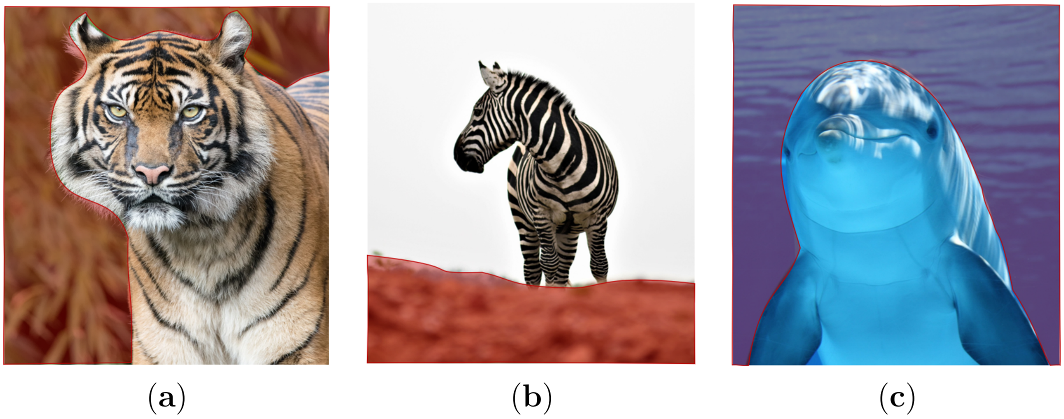


Figure 1.6: Possible explanation images for the (**a**) tiger, (**b**) zebra and (**c**) dolphin. The backgrounds of the images are highlighted since the model would have learned to differentiate the different animals by the background colour instead of by the animal itself.

bution methods have emerged in the literature. However, each one can generate a different attribution map (for the same decision) and not always being the results consistent among the different methods. Due to the lack of a ground truth defining what is a correct explanation, it is difficult to know which explainability method provides a more faithful explanation. This, introduces a new source of bias, is the model that is biased? is the explainability method pointing out the actual bias? or, on the contrary, is it not reflecting the model behaviour and therefore, the explanation is misleading (*i.e.*, XAI BIAS)?

Finally, how humans use explainability to find unwanted correlations in the models could also be a new source of bias (*i.e.*, HUMAN BIAS). We have seen that explainability can help us find unwanted correlations in easy problems (*e.g.*, the background colour learned by the model). However, a human inspection of explanations can lead to misleading conclusions due to confirmation bias [71] in more complicated settings. Let us take an example of this scenario to illustrate this problem. Imagine we want to inspect the decisions of a model with respect to the *sheep* class. The feature attribution method could highlight the sheep's body (*i.e.*, red areas), see Figure 1.7. *A priori*, one would think that the model is accurate and that the model has learned the correct characteristics of the *sheep* class since it is pointing to the sheep. However, if we were to analyse the explanation of the two instances in Figure 1.8, we would realize that the model is not looking at the sheep but at the texture of the wool. And therefore, the llama (or in a more extreme setting, anyone wearing a wool jersey) would be a false positive since it will be classified as a *sheep*. And the lamb on the right will be a false negative, it will not be classified as a *sheep*. This shows that the detection of these unwanted

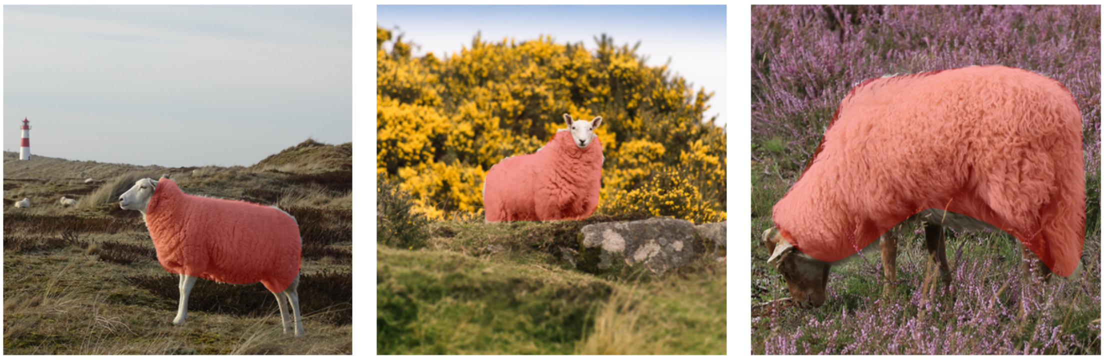
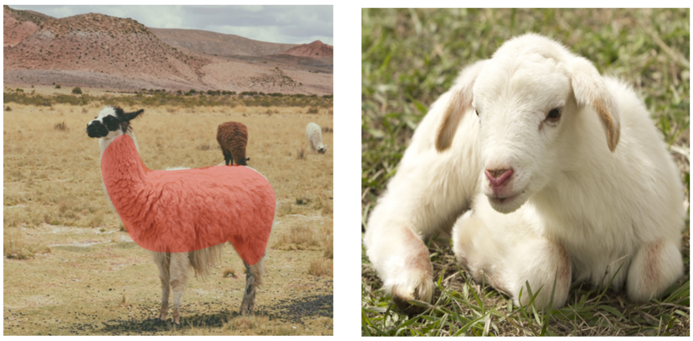


Figure 1.7: Example of a potentially unwanted bias that can go unnoticed. The feature attribution method only highlights the wool (*i.e.*, red areas). This could mean that the model only focuses on the wool texture to classify these images as *sheep*, without learning the other characteristic patterns of the *sheep* class.



(**a**) (**b**)

Figure 1.8: Two instances where the unwanted bias can be noticed. (**a**) A false positive, the lama would be classified as a *sheep* because of the texture of the wool (red areas highlighting the wool). (**b**) A false negative, the lamb will not be classified as a *sheep*, because the model is not able to find the learned texture (*i.e.*, the wool texture).

correlations is not straightforward. Should we look at all the explanations? What samples should we select? Should we use images out of the distribution? In order to allow the end user to benefit as much as possible from these explanations, we will need tools or methodologies that help or guide the human evaluator to better understand the rationale behind the explanations, thus reducing the bias introduced by the human (*i.e.*, HUMAN BIAS).

To recap, let us summarize the main sources of biases introduced in this section (see Figure 1.9). First, we presented how data and models can be biased (*i.e.*,

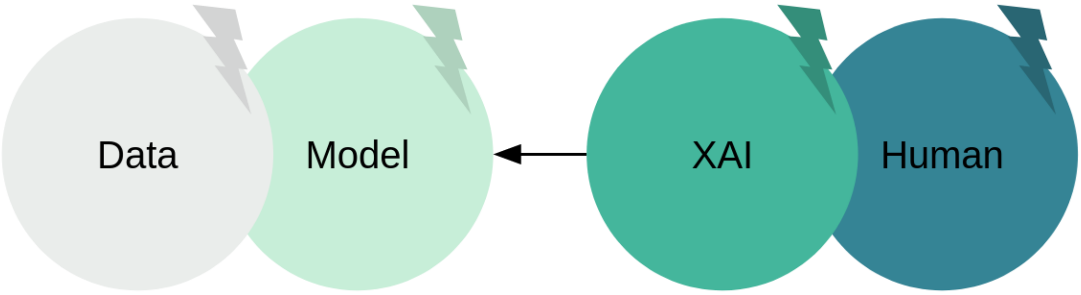


Figure 1.9: Different sources of biases within the pipeline. First, biased data. Second, biased models (*e.g.*, trained on biased data). Third, non-reliable explainability methods that are not faithful to the model. Finally, the interpretation by humans can also introduce biases into the framework.

DATA BIAS and MODEL BIAS). Second, we introduced how explainability can be useful to have a better understanding of the model behaviour and to detect those biases. However, we also showed how this process of applying explainability can also introduce even more noise into the pipeline. On the one hand, because there are different explainability methods providing different explanations (*i.e.*, XAI BIAS). On the other hand, because humans can also introduce biases when interpreting those explanations (*i.e.*, HUMAN BIAS).

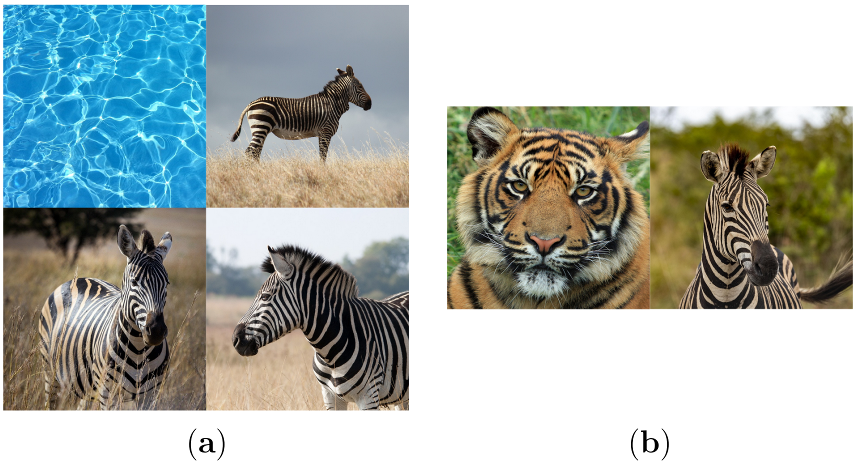
RESEARCH QUESTION

As previously discussed, a challenge for the scientific community is to find which undesirable biases lead the model to erroneous or unintended predictions, regardless of the source of such bias (*i.e.*, DATA, MODEL or XAI). Therefore, a research question the community needs to tackle is:

> Can we find methods to help locate, illustrate and evaluate biases in either datasets, models or XAI methods without falling into human biases?

METHODOLOGY

In this section, we briefly introduce the mosaic methodology to address the previous research question. Let us briefly present the *mosaic* concept, an original idea and contribution from this thesis, to better understand the potential of this



(**a**) (**b**)

Figure 1.10: Mosaic examples. (**a**) 2×2 mosaic composed of three zebras and a water image (*i.e.*, potential bias). (**b**) 1×2 mosaic composed of one tiger and one zebra, both surrounded by vegetation (*i.e.*, potential bias).

tool for the task at hand. Mosaics have two inherent properties that make them suitable for bias detection in explainability methods and in data and/or models. First, mosaics maintain the visual patterns of the original distribution, reducing the induced noise. Secondly, mosaics introduce a source of confusion in a controllable and scalable manner (mosaics are custom and easy to generate) while being able to challenge the model.

Mosaics are built by combining images within a grid. The size and the configuration of mosaics will depend on the target task. Let us imagine we are interested in assessing whether *water* is a bias for the animal classification model previously introduced (*i.e.*, zebras, tigers and dolphins). We could create, for example, a mosaic of size 2×2 by combining three images of zebra and one of water, see Figure 1.10 (**a**). Then, we could quantify the relevance of the *water* bias by analysing the model's response (*i.e.*, with which certainty is the mosaic a zebra or a dolphin) or by pinpointing the amount of explainability that falls into the water image within the mosaic.

In a different scenario we could not know of any bias *a priori*. For example, let us imagine that in the previous image classification model we notice that the model gets confused when classifying the *tiger* and *zebra* classes. We could generate a number of $1{\times}2$ mosaics by combining images from these two classes. And we could fed the model with those mosaics. This would allow us to first confirm the existence of a potential shared bias between some tiger and zebra images (*e.g.*, we realize that the model underperforms when fed with the mosaic shown in Figure 1.10 (**b**), where the vegetation is present in both images), and then, to analyze the impact of this bias (*e.g.*, is the vegetation more relevant to the model than the animal itself?). These and other related aspects of mosaics will be studied along this thesis.

STRUCTURE OF THIS DOCUMENT

This document is divided into four main parts. In Part I, we present the related work. This includes Chapter 2, 3 and 4. First, we introduce in Chapter 2 the different types of explanations existing in the CV field for NNs. We delve into feature attribution methods in Chapter 3. And finally, in Chapter 4, we present different existing evaluation techniques for feature attribution methods.

In Part II, we focus on minimizing the XAI BIASES. To do so, we evaluate the explainability methods according to their reliability to the model behaviour. We perform this assessment in a quantitative manner, avoiding the presence of humans in this first evaluation. More concretely, in Chapter 5, we present how we came to identify the research problem. In Chapter 6, we introduce the proposed score —the *Focus*—to evaluate feature attribution methods. In Chapter 7, we put into practice the *Focus* score, showing and analysing the results obtained. In Chapter 8, we propose an extended framework to improve the *Focus* limitations. Finally, in Chapter 9, we show how the *Focus* and mosaics can be applied to other modalities such as the Natural Language Processing (NLP) domain.

In Part III, we introduce different ways of using the explainability along with mosaics to identify and detect potential biases in DATA and MODELS, while minimizing both XAI BIASES (*i.e.*, having selected the most faithful XAI method according to the *Focus*) and HUMAN BIASES (*i.e.*, semi-automating bias detection reducing human intervention). More specifically, in Chapter 10 we introduce the reason why the *Focus* together with mosaics is a powerful tool to help the automation of bias detection and we analyze the *Focus* behaviour on a biased model. Finally, in Chapter 11, we also use mosaics to analyze the context bias learned by the model, but in this case regardless of the use of explainability methods, thus avoiding both XAI and HUMAN biases. To do so, we present a new way of constructing mosaics in order to assess the relevance of biases known *a priori*.

Finally, Part IV ends this document with the conclusions in Chapter 12. And in Chapter 13, we introduce the contributions related to this thesis.

Part I.

State of the art

2 | EXPLAINABILITY METHODS

> Sometimes, explanations give us a sense of control.
>
> *Tania Lombrozo*

Due to the growing attention towards explainability for ML methods in recent years, the number of proposed techniques in the CV field is astonishing. Different works have categorized these techniques in distinct ways [11, 1, 17]. Some of these categories found in the literature are detailed below (see Figure 2.1). Coarsely, explainability methods can be differentiated by being intrinsic or *post-hoc* methods. In other words, the models can be transparent and therefore, they are interpretable by design. Or, the explainability methods are applied *a posteriori* when the model is already trained. One can distinguish between model-agnostic and model-specific among the *post-hoc* methods. The former can be applied to any type of model. The latter can only be applied to a specific one (*i.e.*, for each model, the method should be adapted). Depending on whether the explanation describes the general model behaviour or explains a specific instance, we can differentiate between global or local explanations. Another differentiation criterion [15] is the type of data these explainability methods are intended to explain (*i.e.*, images, text or tabular data). Some of them are suitable for all types of data, others for a specific subset.

Within the large field of XAI, this thesis focuses on a subgroup of explainability techniques. As previously introduced, since the target models are complex models (*i.e.*, NNs), we concentrate on *post-hoc* methods. We analyse the methods that are either specific to images or data agnostic. And regarding the explanations generated, we tackle the XAI methods providing local explanations. Having introduced the specific subgroups that we will focus on, let us review some of the most relevant techniques. We group them into three families depending on which question they try to answer, this includes feature-based methods (§2.1), concept-based methods (§2.2) and counterfactuals methods (§2.3), which are discussed next.

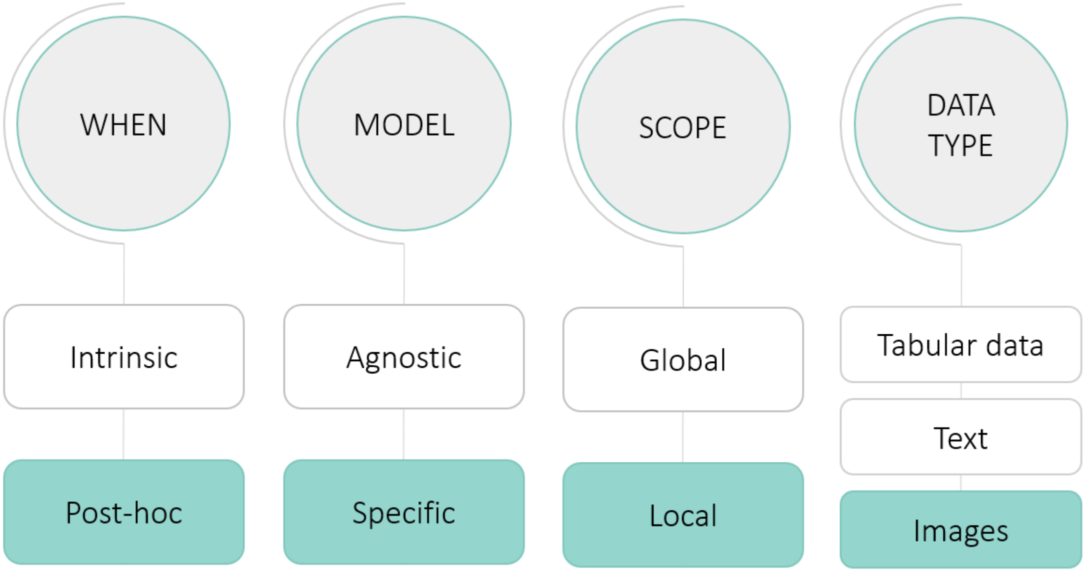


Figure 2.1: Different categories of explainability methods depending on when the explanation is generated, on what kind of model it can be applied, the scope of the explanations and the data type they are intended to explain. In this work, we focus on the explanations subgroups highlighted in green.

2.1 | FEATURE-BASED METHODS

The explanations provided by these methods answer the question of **what patterns or parts of the input image are most relevant to the model decision**. As presented in the Introduction 1, these methods are also called *feature attribution* methods. And they provide feature attribution maps as explanations, where the features that have been relevant to the model prediction are highlighted. A toy example of this kind of explanation is shown in Figure 1.6: where the green part of the tiger image is highlighted, meaning that that region was important for the tiger prediction, just like the brown part for the zebra and the blue for the dolphin. This family of methods are the most widespread and used in the CV field. For this reason, we focus our research on this group of methods. As this is a large category and also encompasses different subgroups, the whole next chapter discusses them in more detail (see Chapter 3).

2.2 | CONCEPT-BASED METHODS

Instead of focusing on the contribution of low-level concepts, such as pixels, this family of methods focuses on the importance of high-level concepts. Ghorbani *et* *al.* argue that individual pixels are not meaningful to humans [29]. However, a group of pixels corresponding to a concept are more intuitive and understandable to humans. Thus, this family of methods answer the question of **which concepts of the input image are important for the model decision**. Going back to the illustrative example of animals (*i.e.*, tiger, zebra and dolphin), an example of a concept-based explanation would be something similar to the one shown in Figure 2.2. Therefore, the *vegetation* concept will be considered important (remember that the model had learned the context instead of the characteristics of each animal). However, the eyes, the stripes or the nose of the tiger would not be relevant concepts to the model decision.

Kim *et al.* represent the concepts as a set of vectors, and they call it Concept Activation Vector (CAV) [44]. In order to learn the CAV of a concept, for example, the CAV for the *eye* concept, the system needs examples of eyes and random examples. Then the activations for the two sets of images (through the trained neural network) are collected, and a linear classifier is trained to separate the two groups of images. The CAV vector is the orthogonal vector to the decision boundary. To obtain the quantitative explanation of the concept, the authors compute the Testing with Concept Activation Vector (TCAV) score, which is obtained by computing the directional derivative of the logit with respect to the CAV. This is done for many eyes images, and the final TCAV score is the ratio between the positive directional derivatives and the total number of inputs.

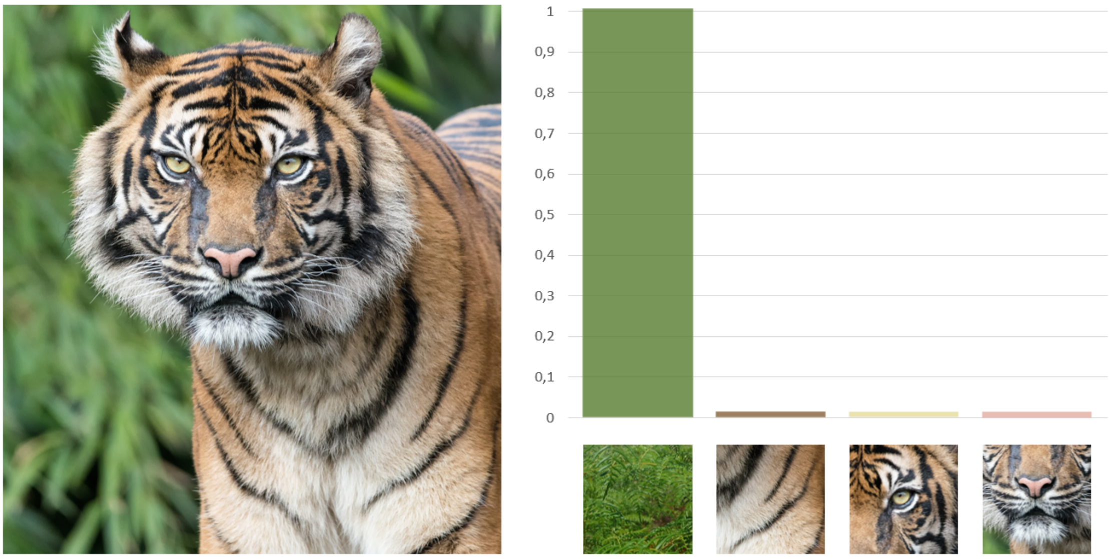


Figure 2.2: A possible concept-based explanation for the tiger. The *vegetation* concept is important for the prediction. However, the stripes, the eyes or the nose are not considered important concepts for the model decision.

One of the drawbacks of this method is that users must know the concepts *a priori* and must have the necessary examples for training the classifier. To solve this problem Ghorbani *et al.* propose a new algorithm called Average Causal Effect (ACE) [29]. This algorithm consists of three steps: first, all images are segmented at different resolutions. Then similar segments are clustered, and each cluster corresponds to a different concept. In the last step, the TCAV score is computed for each concept from the different examples of each cluster.

2.3 | COUNTERFACTUAL METHODS

These types of explanations address the question of **what should be different in the input image so that the output prediction X was Y**. Continuing with the toy example, examples of counterfactual explanations would be those shown in Figure 2.3. The explanation (**a**) indicates that if the tiger were surrounded by water, the tiger image would be classified as a dolphin. The image (**b**) shows that the zebra would be classified as a tiger if the brown area were green. Or the explanation (**c**) evidences that if a brown background surrounded the dolphin, the dolphin would be classified as a zebra.

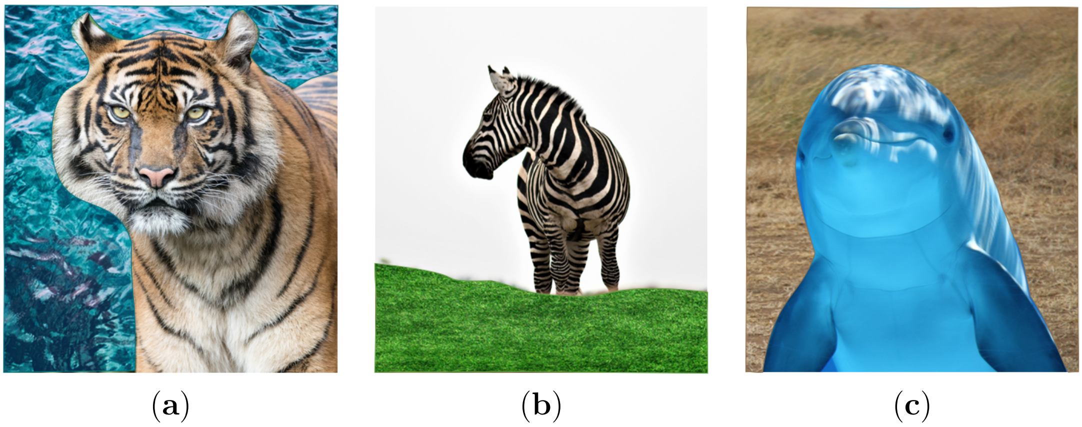


Figure 2.3: Possible counterfactual explanations for the (**a**) tiger, (**b**) zebra and (**c**) dolphin. The background shown in each case is the one that would have caused the image to be classified as another class. (**a**) The tiger would be classified as a dolphin if the tiger were surrounded by water. (**b**) The zebra would be classified as a tiger if the zebra were surrounded by vegetation. (**c**) The dolphin would be classified as a zebra if the dolphin were surrounded by a brown background.

Let us now introduce some of the existing methods to provide a general overview of these counterfactual techniques:

PlausIble Exceptionality-based Contrastive Explanations (PIECE) [43]: this technique first identifies the counterfactual class of a prediction. For example, let us imagine an image *img* is predicted as c, but the original class was c', the counterfactual class of *img* would be c. Then, the method searches for the exceptional features on the latent space with a low probability of occurrence in c'. And finally, the explanation is generated using a Generative Adversarial Network (GAN) [31] by progressively changing the exceptional features that will move the prediction from c to c'.

Contrastive Explanations Method (CEM) [25]: this method finds the pertinent positives (PP) and the pertinent negatives (PN) of each instance decision and uses them as contrastive explanations. The PP are the minimally sufficient pixels or regions for the prediction. And the PN are the ones whose absence is necessary for the prediction.

Counterfactual Visual Explanations (CVE) [33]: this technique finds the counterfactual explanation of an image *img* predicted as class c, by first selecting an image img' predicted as class c'. Then, the method searches for the minimum region of img' that being replaced in *img* will change the prediction of *img* from class c to class c'.

Search for EviDence Counterfactual with Target counterfactual class (SEDC-T) [105]: this method is based on the Search for Explanations for Document Classification (SEDC) [59]. The SEDC algorithm searches the minimum number of words that, when removed, would change the document classification. The SEDC-T technique is based on the same idea but for image classification. On the one hand, instead of working at the pixel level, they perform image segmentation. Those segments will be combined until finding the minimum number of segments that will change the image classification. Since image classification problems are usually not binary, the authors add the possibility of specifying the target counterfactual class.

Most of the proposed counterfactual techniques in the current literature are not data agnostic and mostly are intended for tabular data [35]. However, as the ones previously introduced, there are some designed for images (*e.g.*, CVE, SEDC-T or PIECE) or even data agnostic (*e.g.*, CEM). Many of these methods are only tested on simple datasets such as MNIST [52] or EMNIST [21], where those counterfactual explanations (working at pixel level) may be interpretable (*e.g.*, if there is a horizontal stroke in the middle of a zero it will be an eight instead of a zero). However, those abstractions are less interpretable in more complicated datasets such as ImageNet [85]. To get more interpretable explanations when using complicated datasets, some methods use more abstract levels such as regions instead of pixels. This is the case of the CVE and SEDC-T methods. The former tested their method on the Caltech-UCSD Birds (CUB) 2011 Dataset [107], and the latter on ImageNet. As also discussed by Vermeire *et al.* [105], one drawback of most of these methods (*e.g.*, PIECE, CEM or CVE) is that they require access to the training data which may reduce the applicability of these approaches in real case scenarios, where the data is not available.

2.4 | SUMMARY OF THIS CHAPTER

This chapter introduces the three large families of explainability methods in the CV field. Feature-based methods are widely applied in many applications. Concept-based methods are more recent and, therefore, still less used. And counterfactual methods are not widespread in this field either, this is probably due to the disadvantages mentioned above (*i.e.*, poorly interpretable in more complicated datasets or access needed to training data). In addition, counterfactual methods are usually computationally expensive.

Since each category answers a distinct question, each type of explanation provides different information, which at the same time may be complementary. Therefore, the ideal scenario would be to have an explanation of each since we would have information at a pixel level (*i.e.*, same as the model reasoning process), at a more abstract level (*i.e.*, same as the humans reasoning process) and at the counterfactual level (*i.e.*, why was it not predicted differently?).

3 | FEATURE ATTRIBUTION METHODS

> Nothing in life is to be feared; it is only to be understood.
> Now is the time to understand more, so that we may fear less.
>
> *Marie Curie*

Many techniques have been proposed within the *feature attribution* category. In this chapter, we will highlight the most relevant ones. Following the categorization presented by Rao *et al.* [79], we divide these feature attribution methods into three main groups depending on the technique used to generate the explanations: the perturbation-based methods (§3.1), the backpropagation-based methods (§3.2) and the activation-based methods (§3.3).

3.1 | PERTURBATION-BASED METHODS

The methods under this category obtain the attribution maps by perturbing the input image and analyzing the change in the output. These methods treat the model as a black box without needing to access the model's internal parts. Some of them are detailed below.

Occlusion [111]: in this technique, patches of the original image are turned off, and the prediction variation is observed. The most important features for the model will be those patches that influence the prediction most.

Local Interpretable Model-agnostic Explanations (LIME) [81]: this method, instead of using patches, segments the input image into superpixels, and these superpixels are turned off and on. This perturbed data is fed into the network. Then, a weighted linear regression model is trained using the predictions obtained with the perturbed data, the perturbations and the weights. Those weights are computed with the distances between the perturbed images and the original image.

Finally, the importance of each superpixel for the prediction is obtained with the result of the linear regression.

Shapley Additive Explanations (SHAP) [56]: this method is based on Shapley values [91] from game theory. The features of the input images are considered players of the coalitions. Therefore the method estimates the contribution of each feature to the model prediction using the Shapley values. This method has variations such as KernelSHAP, which uses LIME to compute the Shapley values.

Randomized Input Sampling for Explanation (RISE) [77]: this method randomly masks the input and performs a forward pass through the network multiple times. A weighted sum of the masks is performed using the scores obtained as weights to generate the feature attribution map.

3.2 | BACKPROPAGATION-BASED METHODS

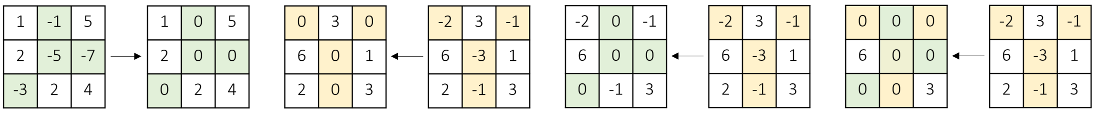
The methods within this group compute the gradient of the prediction with respect to the pixel values of the input image or backpropagate the contribution of the neurons. Numerous techniques have been proposed in this category, the most used are introduced below.

Saliency Map method [96]: this widely used technique generates the so-called *saliency maps*. The authors obtain these explanations by computing the gradient of the class score with respect to the pixels of the input image. The *saliency* map shows how it affects if we change a pixel from the original image to the classification score for the specific class.

Gradient×Input [94]: this approach is similar to the previous one but adds a product by the input image on top of the gradients. Therefore, the explanation is computed by performing an element-wise product of the input image with the gradient of the class score with respect to the input image.

Deconvnet [110]: this technique maps the activations at high layers back to the input pixel space using a Deconvnet [112]. This technique is quite similar to the Saliency Map method. How the authors handle the Rectified Linear Unit (ReLU) non-linearity is the main difference: the Deconvnet method only backpropagates positive values, that is if the backward signal itself is positive. On the other hand, the Saliency Map method does not backpropagate the backward gradient if the input of the ReLU through the forward pass is negative. This difference is illustrated in Figure 3.1, where a forward pass and three methods of backpropagating through

ReLU non-linearity are shown. In (**a**), a forward pass is depicted: only positive values are propagated (white cells). The Deconvnet backpropagation is illustrated in (**b**): only positive values are backpropagated (white cells). The backpropagation of the Saliency Map method is shown in (**c**): the values, where the activations in the forward pass where negative (green cells), are not backpropagated.



(**a**) (**b**) (**c**) (**d**)

Figure 3.1: Different ways of handling the ReLU non-linearity. (**a**) Forward pass. (**b**) Backward pass: Deconvnet method [110] (**c**) Backward pass: Saliency Map method [96]. (**d**) Backward pass: Guided Backpropagation method [101]. Replicated image from [101].

Guided Backpropagation (GB) [101]: this technique handles the ReLU non-linearity by combining the Saliency Map and the Deconvnet backpropagation approach. Thus, this method backpropagates positive values as long as the input of the ReLU in the forward pass is positive; see Figure 3.1 (d).

SmoothGrad [99]: this method tries to improve the visualizations of the explanations obtained with the aforementioned methods. Smilkov *et al.* argue that this technique obtains less noisy and more sharpened explanations by averaging different explanations obtained with small perturbations on the input image. The authors tested this technique empirically with explanations obtained with different methods, such as the Saliency Map or Guided Backpropagation methods.

Integrated Gradients (IG) [102]: instead of computing local gradients, this method calculates the integral of gradients through the line joining a baseline x' and input x. The idea is to start from an input having a near-zero score (*i.e.*, absence of signal). For example, in CV the baseline could be a black image. And then, the baseline would be interpolated in different steps until reaching the current input x. The approximation of the integral is done by summation of the gradient steps from the baseline x' up to the input x.

Layer-Wise Relevance Propagation (LRP) [12]: this method, instead of using gradients as the previous methods, backpropagates the output prediction to the input image by propagating the contribution of each neuron. To do so, Montavon *et al.* propose in [62] different propagation rules according to the depth of the layer. LRP$-w^2$ and LRP$-z^B$ rules for the first layers. The LRP$-\alpha\beta$, LRP$-\gamma$ and LRP$-flat$ rules for lower layers. And for upper layers, the LRP-0 rule.

Deep Learning Important FeaTures (DeepLIFT) [93]: this technique also backpropagates a relevance score in a similar manner to LRP method. The importance scores in this method are computed from the difference between the output obtained with the original input and some reference image (*e.g.*, black image). The rule used for the backpropagation is called Rescale rule.

3.3 | ACTIVATION-BASED METHODS

This family of methods is mainly based on the Class Activation Mapping (CAM) [114] method. The CAM method requires a specific architecture, especially the model needs a Global Average Pooling (GAP) just after the last convolutional layer and just before the last layer. The feature activation map is computed by a weighted sum of the feature maps of the last convolutional layer. Thus this map highlights the discriminative region for the selected class. Different methods have been proposed on top of this CAM method, some of them are described below.

Gradient-weighted Class Activation Mapping (GradCAM) [89]: it overcomes the architecture limitation of the CAM method. The importance score used to perform the weighted sum of the feature maps of the last convolutional layer is obtained as follows: first, the gradients of the logits of the class with respect to the feature maps of the final convolutional layer are computed. Then, these gradients are averaged across each feature map. And finally, this score is used to perform the aforementioned weighted sum of the feature maps to obtain the final *feature attribution* map. Therefore, the GradCAM no longer requires a specific architecture and thus, the GradCAM can produce visual explanations for any CNN.

Grad-CAM++ [18]: this method is a generalization of GradCAM and tries to allow the feature attribution method to detect multiple occurrences of the target object within an image. Instead of performing the average of gradients across each feature map, this method proposes a weighted combination of the positive partial derivatives.

Score-CAM [108]: it does not use the gradients to compute the weighted sum. Instead, the weights are the scores obtained by passing the activation map, multiplied by the input image, through the network. These weights are then used to perform the weighted sum of the activation maps as the previous methods do.

3.4 | SUMMARY OF THIS CHAPTER

In this chapter, we discussed different feature attribution techniques grouped into three distinct categories: perturbation-based, backpropagation-based, and activation-based. The main advantage of the perturbation-based methods is that they are model agnostic, meaning that they can be used in any model. The downside is their high computational cost: many forward passes are required for each instance. Instead, the backpropagation techniques are generally faster since they only require a single forward pass. As previously said, the most basic technique of this group is the Saliency Map method. However, the explanations obtained with this method are visually noisy. Much work has been done to improve these explanations. For example, Deconvnet and GB tried to improve the clarity of the produced Saliency Maps. Even if both techniques provide better explanations (qualitatively speaking), Nie *et al.* [72] proved that both techniques are more interpretable because they are (partially) making image recovery. Therefore, they are *less faithful* to the model behaviour than the Saliency Maps. According to Shrikumar *et al.* [93], the noisiness of the Saliency Maps may be due to the gradient saturation problem. In the saturated area, the gradients are small; therefore, important features may not be highlighted even if they are relevant to the prediction. Trying to solve the saturation problem, other approaches have been proposed in this direction (*e.g.*, Gradient×Input, LRP, DeepLIFT or IG). However, it seems that the saturation of the gradients still affects some of these methods. Miglani *et al.* [60] showed how the explanations produced by IG are dominated by the gradients from the saturated areas. Finally, a disadvantage of both backpropagation and activation methods is that they are not model agnostic, and access to the model is required to be able to compute the explanations. The main advantages of these methods are summarized in Table 3.1.

Table 3.1: A brief summary of the advantages of each family of methods.

Method family	Model agnostic	One forward pass	Post hoc	Less noisy explanations	Methods
Perturbation based	✓		✓	✓	[111], [81], [56], [77]
Backpropagation based		✓	✓		[96], [94], [110], [101] [99], [102], [93], [12]
Activation based		✓	✓	✓	[89], [18], [108]

4 | EVALUATION METHODOLOGIES

> We don't see things as they are,
> we see them as we are.
>
> *Anaïs Nin*

As discussed in the previous chapter, there are many feature attribution methods, each one generating a different explanation for the same instance (see Figure 4.1). So the next question that one would ask is: *how* to select the best method? To solve this question, different directions have been taken in this field of research. We can group the techniques into two broad groups. Qualitative methods and quantitative methods. The former includes the human in the evaluation process, and the latter excludes the human from the loop. While the qualitative techniques are suitable for measuring the degree of the interpretability of the explanations as well as their usefulness for humans, the quantitative methods are suitable for assessing the coherency of the explanations to the model behaviour.

An example of a qualitative evaluation technique consists of asking human evaluators to choose the best-performing model based on the explanations [81]. Or, for instance, another example is to ask the evaluators which explanation (among explanations generated with different XAI methods) they would prefer for a specific model prediction [89]. This kind of evaluation is based on the human under-

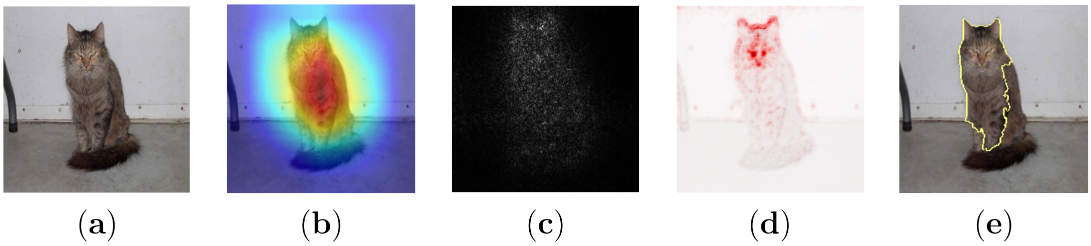

(**a**) (**b**) (**c**) (**d**) (**e**)

Figure 4.1: Explanations obtained with different feature attribution methods for the same instance. (**a**) Original image. And the corresponding explanations obtained with: (**b**) GradCAM, (**c**) SmoothGrad (**d**) LRP and (**e**) LIME.

standing of perception, and therefore potentially biased (*e.g.*, automation bias, confirmation bias). Furthermore, as previously mentioned (see Chapter 1), our perception process might not be aligned with that of the model.

In short, these qualitative evaluation methods measure what Jacovi *et al.* [41] called *plausibility*. This term refers to how convincing the explanations are to humans. In contrast, what quantitative methods assess is the *faithfulness*. Jacovi *et al.* define this last term as how well the explanation reflects the model behaviour. *Plausibility* is crucial since the end user of these methods are humans. A non-plausible explanation for humans (*i.e.*, an explanation that humans do not understand) will be useless. However, before evaluating the *plausibility*, one must check whether the method is faithful to the model behaviour, since an unfaithful explanation, non-coherent with the model behaviour, could be misleading. And this is the focus of this work: the quantitative evaluations (*i.e.*, assessing the *faithfulness* of the explanations).

4.1 | QUANTITATIVE EVALUATIONS

Different methodologies in the current literature try to quantitatively evaluate feature attribution methods. There is not yet a community-accepted way to categorize these methods. The work of Anna Hedström *et al.* [37] divides the existing methods into six categories: methods assessing the faithfulness, the robustness, the localization capacity of the methods, the complexity (*i.e.*, whether they are concise), randomization methods and the axiomatic group (*i.e.*, techniques proposing axiomatic properties that should be satisfied by the XAI methods). Similarly, Akhtar categorizes the evaluation techniques into four groups according to the quantified properties [3]: methods quantifying the model fidelity, quantifying the localization ability, the stability, and other desirable properties (*e.g.*, axiomatic properties).

However, the main challenge of these quantitative methods is the lack of a ground truth specifying what defines a correct explanation. We found it more convenient to group them according to the different approaches proposed to circumvent this problem (see Figure 4.2). Even though some methods overcome the ground truth problem by generating a pseudo-ground truth (§4.1.1). The vast majority of feature attribution methods accept the absence of the ground truth and make an assumption of certain behaviours on the expected response of the XAI methods. Examples of those approaches are methods defining axioms that the feature attribution methods should fulfil (§4.1.2). Methods proposing randomization experiments (§4.1.3), assuming that a random behaviour should have an effect on the explanation. Or techniques that perform perturbations on the input, according to the explanation, and then analyse changes in the output (§4.1.4). Let

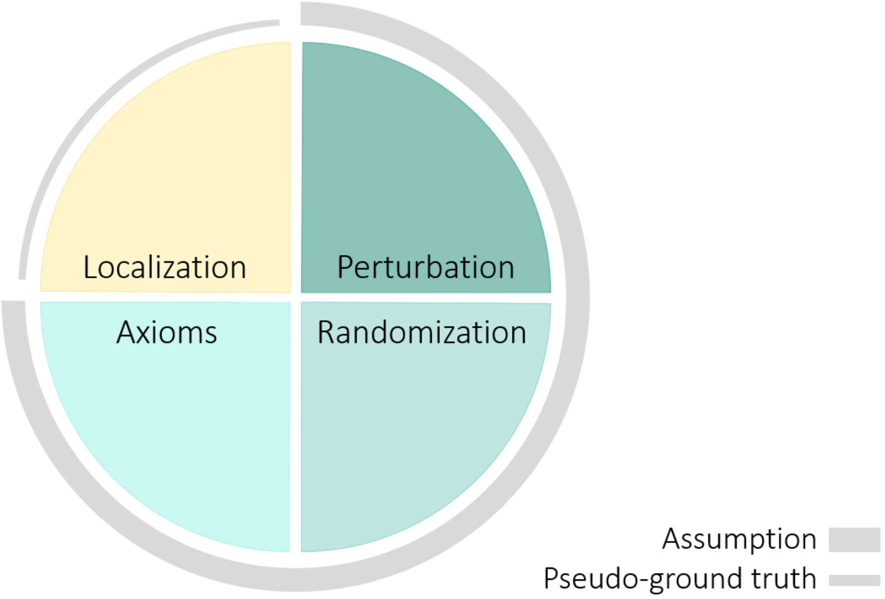


Figure 4.2: Different classes used for categorizing quantitative evaluation methods. Those surrounded by the thick grey circle correspond to methods assuming an expected response of the feature attribution methods. Instead, those surrounded by the thin circle generate a pseudo-ground truth.

us now take a closer look at some of the most relevant techniques within these families of methods.

4.1.1 | LOCALIZATION METHODS

As previously introduced, this family of methods generates a pseudo-ground truth. In other words, a part of the input image is assumed to be where the relevance should be concentrated (*e.g.*, object bounding box or segmentation masks), while the rest are considered areas where no explanation can be found. The evaluation is calculated based on the amount of XAI relevance lying in these ground truth areas. Note that we refer to XAI relevance as the pixels attribution provided by the feature attribution methods, representing the contribution of the pixels to the model prediction. Some examples of these techniques are:

Pointing Game [113]: this technique evaluates whether the point of maximum relevance lies in the object category. The authors compute the accuracy for each object and average those accuracies among the different classes.

Attribution Localization [48]: the score proposed by Kohlbrenner *et al.* corresponds to the proportion of positive relevance that falls within the bounding box of the target object with respect to the sum of the positive relevance of the image. The authors also introduce a weighted version, where the score is weighted by the bounding box size. The smaller the bounding box and the more relevance falls into it, the higher the weighted score will be.

an8Flower [74]: in this work, a synthetic dataset is proposed where the discriminating features of each class are controlled. In this way, the ground truth is known and thus, the Intersection Over Union (IoU) between this mask and the explanation can be computed as a quantitative evaluation score.

4.1.2 | AXIOMS

This category encompasses evaluation methods defining axiomatic properties that must be satisfied by the explainability techniques. Examples of these axioms are:

Sensitivity(a) [102]: if an input and a baseline differ in one feature and that difference changes the prediction. This axiom will be fulfilled if that specific feature has assigned a non-zero attribution.

Sensitivity(b) [102]: this axiom verifies that if the function of the model does not depend on a feature, that feature will have no explanation attributed. This axiom is related to the *Non-Sensitivity* [69] axiom, which states that only the features on which the model does not depend will have zero attribution.

Implementation invariance [102]: this axiom checks whether functionally similar models produce equivalent attributions.

Completeness [102]: this axiom is satisfied when the sum of the attributions is equal to the difference between the output of the input and a baseline.

Input Invariance [46]: this axiom is fulfilled when a transformation, which does not affect the prediction nor the weights, is applied to the input image (*e.g.*, constant input shift) and this transformation does not affect the attributions.

4.1.3 | RANDOMIZATION TESTS

The methods proposed within this category start from the premise that, if a feature attribution method is explaining the model behaviour and we modify the model, the resulting explanation should change coherently. Some of the randomization methods proposed in the literature are the following:

Model parameter randomization test [2]: this test compares explanations generated with a trained model with the ones from a randomized model (*e.g.*, where the weights of the deep model are initialized to random values). If the explanation depends on the model, the explanation should be fundamentally different.

Data randomization test [2]: this test compares explanations generated from two different supervised models. One correctly trained and another trained with randomly permuted labels. If there is a relation between the explanation and the labels, the explanations should differ.

Random Logit [98]: this test computes the difference between the explanation using the ground truth logit and the explanation using a random logit, expecting the explanation to be different.

4.1.4 | PERTURBATION METHODS

These methods are based on the premise that highly attributed features are expected to be more important for the model outcome. Therefore, if one perturbs the inputs according to the explanations (*i.e.*, taking into account the most highly attributed features), the effect on the outcome should be maximized. Quantifying this effect will allow us to evaluate the performance of these feature attribution methods. Let us see some examples:

Pixel Flipping [12]: this method perturbs the input pixels (*e.g.*, pixel·(-1)) according to the attributions (in descending order) and assesses the impact produced in the prediction score.

Region Perturbation [86]: this technique is similar to the *Pixel Flipping* technique, but instead of performing the perturbation at the pixel level, areas (*e.g.*, local windows) are perturbed (*e.g.*, with local randomization or blurring).

Sensitivity-N [4] this method calculates the correlation between the sum of the attributions of a group of features with respect to the change produced in the output if those features are perturbed.

Average Drop % metric [18]: this method compares the prediction when only the attributed parts of the image are shown to the model with respect to the prediction with the whole image.

RemOve And Retrain (ROAR) [40]: this method evaluates the degradation of the model performance when the model is trained with instances where the most important features are removed. The authors re-train the model at different degradation levels.

RemOve And Debias (ROAD) [83]: this approach also measures the accuracy of the model when removing the most important features but avoids the re-training step performed by ROAR. Instead of inputting a fixed value when removing the pixels, the authors proposed a Noisy Linear strategy for the imputation.

4.2 | SUMMARY OF THIS CHAPTER

In this chapter, we presented the most relevant evaluation techniques in the current literature, grouped into four classes. The localization techniques allow the evaluation of the XAI methods capability to concentrate the relevance on specific regions of interest. However, this will not always be correlated with the faithfulness of the feature attribution methods. Let us imagine we explain the model introduced in the Introduction 1, the one that learned to distinguish the three animals (*i.e.*, tiger, zebra and dolphin) by the background colour. A reliable feature attribution method should concentrate most of the relevance on the background and not on the object. Therefore, the evaluation obtained with these localization techniques (*i.e.*, assuming that the relevance should lie on the object), will generate a misleading evaluation.

The methods proposing axioms allow us to check whether the feature attribution method fulfils these desirable properties or not (*i.e.*, these methods produce categorical evaluations). However, these axioms do not allow us to rank the XAI methods according to their faithfulness.

The most used evaluation techniques are the perturbation methods. However, these methods disturb the input images to perform the evaluation. Those disturbed images become instances outside the original data distribution, which may reduce the reliability of those approaches, and render the model behavior noisy or unstable. One of the methods introduced in this section (ROAR), gets around this problem by retraining the models at different degradation levels. However, the out-of-distribution problem is solved despite a high computational cost.

Finally, randomization tests are required as sanity checks since a feature attribution method producing the same explanation for a trained model as for a random model will not be faithful to the model. However, while these tests are necessary, they are also insufficient to assess faithfulness. These randomization methods should be combined with other complementary assessment techniques.

Table 4.1: A brief summary of the main characteristics of each family of methods.

Method family	Make an assumption	Generate pseudo-ground truth	In-distribution instances	Score	Methods
Perturbation	✓			✓	[12], [86], [4], [18], [40], [83]
Localization		✓	✓	✓	[113], [48], [74]
Randomization	✓		✓	✓	[98], [2]
Axioms	✓		✓		[46], [102]

Part II.

Mosaics for XAI Evaluation

5 | INITIAL EXPLORATION

> All sorts of things can happen when you're open to new ideas and playing around with things.
>
> — *Stephanie Kwolek*

This chapter explains the origin of the research question this doctoral thesis tries to address. As discussed next, our first contact with explainability was with the purpose of better understanding a new dataset that was about to be released: the Museum Art Medium (MAMe) dataset.

The MAMe dataset is a public dataset, presented in [76] which consists of images extracted from three different museums: The Metropolitan Museum of Art of New York [119], Los Angeles County Museum of Art (LACMA) [118] and The Cleveland Museum of Art [117]. MAMe contains images from 29 medium classes validated by experts from Universitat de Barcelona (UB). These mediums can range from material aspects, such as *ceramic*, to complex techniques, such as *etching*, see Figure 5.1.

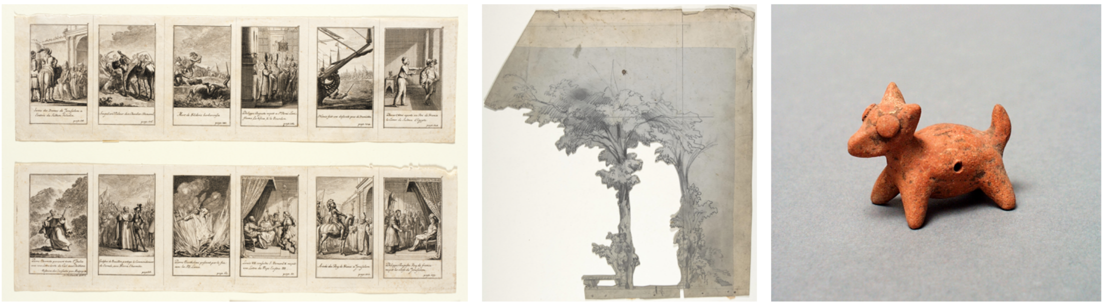


Figure 5.1: Examples of different MAMe instances. The first sample corresponds to the *etching* class, the second to the *graphite* class and the third to the *ceramic* class. Notice the difference in aspect ratio and composition among the different images.

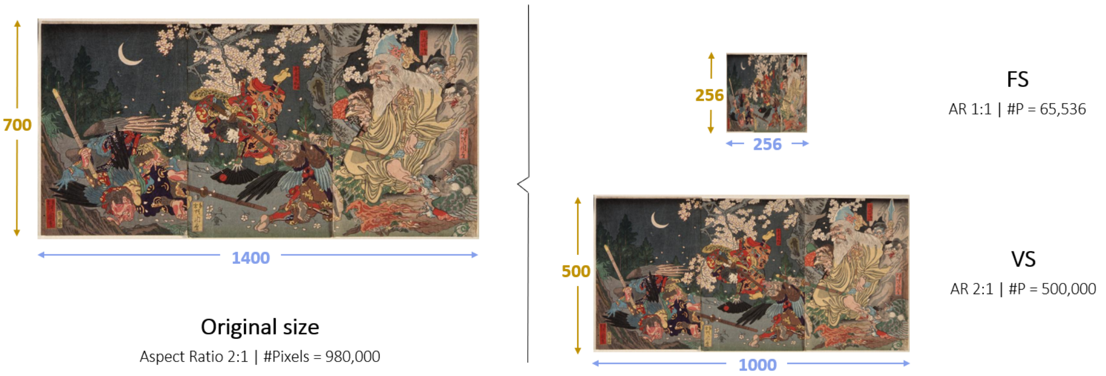


Figure 5.2: On the left, an original image from the MAMe dataset. On the right examples of both data types. The top right type corresponds to a fixed shape (FS) image, with an aspect ratio (AR) of 1:1, being 65,536, the total number of pixels. On the bottom right, a variable shape (VS) image, with an AR of 2:1 and a total of 500k pixels.

The 37,407 images that compose MAMe have variable shape (VS) and high resolution (HR). The average number of pixels per image is around $2,350 \times 2,350$. For this work, two baseline models were created using a VGG11 [97] architecture to highlight the proposed task's feasibility and complexity. These two models were trained for a classification task on two versions of the MAMe dataset:

- The first model was trained on images downsampled to 256×256. This includes losing the original aspect ratio (see the top right image in Figure 5.2). This is the fixed shape (FS) model.
- The second model was trained to keep the original aspect ratio of the images. To reduce the computational and memory costs, the smallest dimension of the images was reduced to 500 pixels. The largest one was proportionally diminished, thus keeping the original aspect ratio (see the bottom right image in Figure 5.2). This is the VS model.

To inspect these baseline models and to gain insight into the relevant features of the MAMe dataset, the *Composite* LRP method was implemented and applied to these models. Following the recommendations of LRP's authors [62], the LRP rules were combined in this order: for the first layer of the network the LRP$-z^B$, for intermediate layers LRP$-\epsilon$ ($\epsilon = 0.25$) and LRP$-\gamma$ ($\gamma = 0.25$) and for last layers the LRP-0. The explanations obtained with the *Composite* LRP were presented to the art experts. The goal was for them to assess the consistency of the important features considered by the model with respect to the features that the art experts use to discriminate between classes. This process led to the following findings:

- Thanks to the assessment of the explanations, we discovered that the *carved stucco* class was not suitable for the MAMe dataset. Many of the *carved stucco* images appear with a ruler and a piece of paper with a number (see the first row of Figure 5.3). As can be observed in the explanation (second row of Figure 5.3), the positive relevance (*i.e.*, red areas) is concentrated on the ruler, meaning that the model considers this a meaningful feature for the prediction of the *carved stucco* class. In other words, the model learnt this shortcut solution instead of learning the characteristics of this medium. As a result, this class was removed from the dataset.

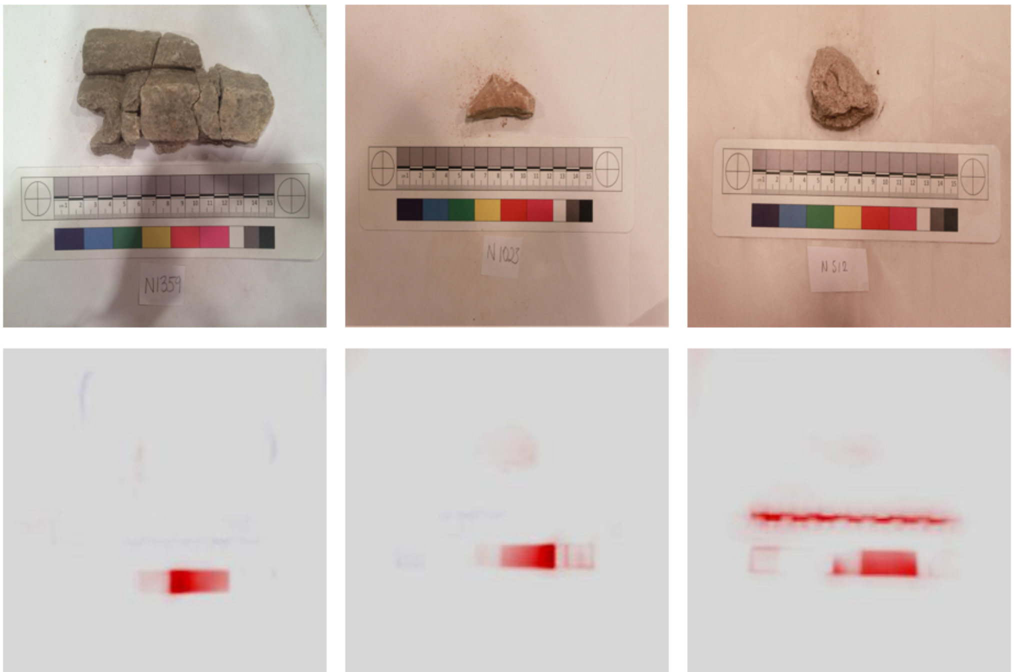


Figure 5.3: Examples of *carved stucco* images are shown in the first row. In the second row, their corresponding feature attribution maps obtained with the *Composite* LRP. Notice that red areas correspond to the input features contributing to the *carved stucco* class. Instead, blue areas correspond to features favouring other classes.

- In a second step, the art experts validated whether the differences between the features learned by the models with HR images with respect to the features learned by the models trained with low resolution (LR) images were coherent with the features the experts considered important.

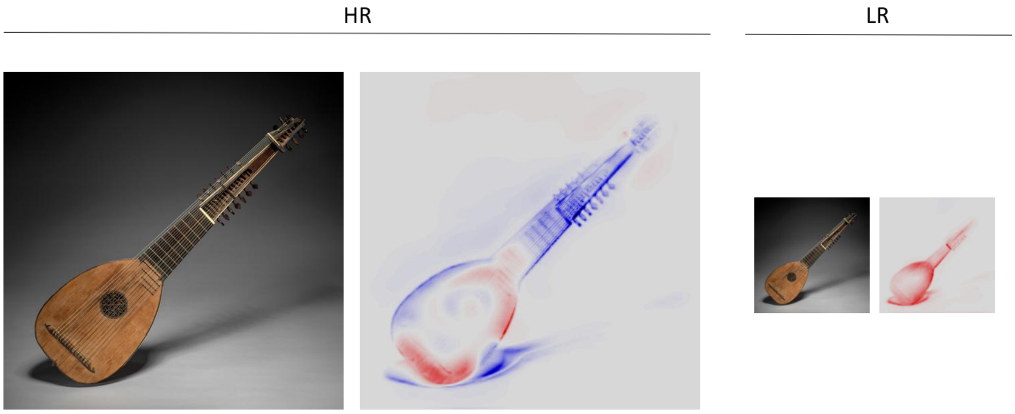


Figure 5.4: A *wood* class image in high-resolution (HR) and low-resolution (LR). Next to its corresponding LRP-based feature attribution map obtained using the model trained on HR and LR images.

Figure 5.4 shows an example of how different can be the explanations depending on the training resolution. According to the explanation, in HR, the body of the guitar contributes positively to the prediction of the *wood* class (see red areas of the explanation on the left). On the contrary, the neck of the guitar contributes negatively (blue areas), that is, it is characteristic of another class. On the other hand, in LR, according to the explanation, the whole image shape is in favour of the *wood* class (see the explanation on the right). This is likely to be caused by the loss of resolution, which makes details like guitar strings impossible to distinguish.

○ And last but not least, we discovered some shortcuts that the model had learned instead of the proper features of the class, therefore not allowing the

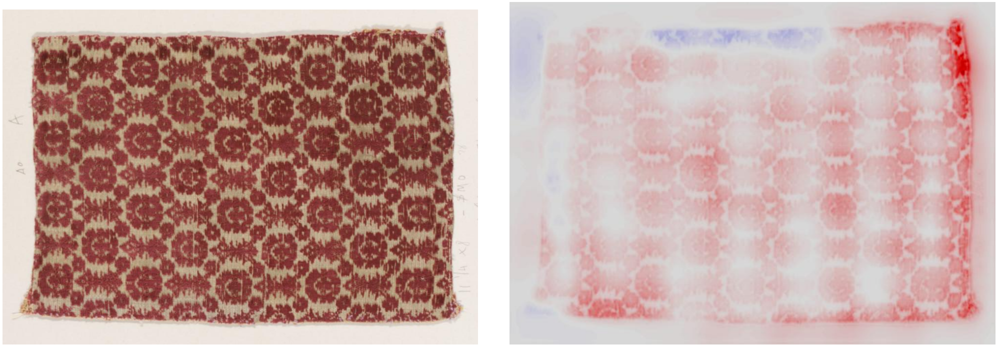


Figure 5.5: *Silk and metal thread* HR image and its feature attribution map. The ornamental motifs (red zones) positively contributed to that class.

model to generalize correctly. For example, according to experts, the *silk and metal thread* class can be distinguished from other textile fibres mainly through the glitter of its metallic threads. However, neither of the two models correctly differentiates these two classes. It seems that the models learned the ornamental motifs as a relevant feature of the *silk and metal thread* class. The experts got this from the explanations like the one shown in Figure 5.5.

5.1 | SUMMARY OF THIS CHAPTER

Performing explainability experiments gave us insights into this new dataset, the MAMe dataset. It allowed us to find some biases present in the data: we drop the *carved stucco* class because all the images had a ruler at the bottom and we found that the ornamental motifs could be a shortcut learned by the model to classify the *silk and meta thread* class. It also allowed to detect failures of the baseline models when discriminating between certain classes due to a lack of resolution. From the XAI perspective, at the end of this work several limitations were identified:

- Only one explainability method was used: the *Composite* LRP. However, many other feature attribution methods could better approximate the model behaviour (*i.e.*, XAI BIAS).
- We checked with experts about the alignment of the explanations with their expert knowledge. However, this could lead to HUMAN BIASES: reinforcing their beliefs (*i.e.*, confirmation bias) or over-relying on the explanations (*i.e.*, automation bias). And what is more, this human-centric evaluation may not be aligned with the model behaviour.
- The art experts' evaluation was time-consuming. The experts had to carry out an exhaustive analysis by checking multiple instances. This approach is not scalable in terms of expert hours. The case of the *carved stucco* class illustrates the importance of speeding up and facilitating the review of datasets and models for biases.

These issues highlight the importance of taking into account the noise introduced by the XAI methods (*i.e.*, XAI BIAS) as well as the biases introduced by humans (*i.e.*, HUMAN BIAS) while using explainability to detect DATA/MODEL BIASES. To rely on these explanations we first need to objectively assess the performance of these feature attribution methods and thus reduce the XAI BIAS. Only then, we could safely use these methods to detect biases present in the models and/or datasets. In addition, semi-automating the detection of these DATA/MODEL BIASES will reduce the inspection time required by experts as well as the noise introduced by them (reducing HUMAN BIAS).

6 | FOCUS FORMALIZATION

> Research is formalized curiosity. It is poking and prying with a purpose.
>
> *Zora Neale Hurston*

In this chapter, we introduce the *Focus* score and its formulation. *Focus* is an evaluation score for feature attribution methods we introduced in this work [8][1]. We first discuss the motivation for creating this new score (§6.1). Then, we introduce the elements needed to calculate the *Focus* score (§6.2) and its formalization (§6.3). Finally, we end the chapter with some experiments designed to test the correct behaviour as well as the robustness of the proposed methodology (§6.4).

6.1 | MOTIVATION

One of the main limitations identified at the end of the previous chapter, as a result of our work in the MAMe dataset is that of selecting a XAI method which is reliable and appropriate for the problem at hand. The challenge of quantitatively evaluating XAI methods lies in the absence of ground truth: we cannot be sure of what a DL method is doing unless we understand the model parametrization itself (at which point we would not need a XAI method). Nonetheless, we still want to approximate the *faithfulness* of XAI methods with respect to the underlying model, as this allows us to discern between accurate and misleading explanations.

Before evaluating the *plausibility*, we must evaluate the *faithfulness* of feature attribution methods. Since explanations that are apparently inappropriate (*e.g.*, the background of the object instead of the object itself) may be an accurate portrait of the model's behaviour, following a *bias* found and learnt from the data (*e.g.*, the zebra, tiger and dolphin example). As introduced in Chapter 4, the most widely used group of evaluation techniques is the perturbation category (*i.e.*, methods adding noise to the input instances). However, those disturbed images

[1] Part of this chapter can also be found in that work.

which fall outside of the original data distribution may reduce the reliability of the analysis because of the effect it may cause on the activations of the model (*i.e.*, are bad explanations caused by a bad method or by the corruption inserted into the samples?). On the other hand, localization techniques assume that the relevance must be on the object, an assumption that is not always true (*i.e.*, are bad explanations caused by a bad feature attribution method or by a shortcut learnt by the model?).

To circumvent these two problems (*i.e.*, the out-of-distribution problem and the localization assumption), we propose a novel evaluation score for feature attribution methods; we call it the *Focus*. First, instead of out-distribution noise, our input alteration approach induces in-distribution noise into samples, that is, alterations on the input but preserving the visual patterns found within the original data distribution. With this, we expect to minimize the exogenous noise added to the process. To do so, we modify the *context* of the sample instead of the *content*, leaving the original pixel values untouched. Second, instead of assuming that the relevance should fall on the object, we assume that the relevance should fall on the sample containing the object. That is, it provides a pseudo-ground truth for the localization of explanations. In practice, we create a new sample composed of samples of different classes: a *mosaic* image (see Figure 6.1).

Using mosaics as input has a major benefit: each input quadrant is an image from the original distribution, producing blobs of activations in each quadrant which are consequently coherent. Only the pixels forming the borders between images and the few corresponding activations may be considered out of distribution. By inducing in-distribution noise, mosaic images introduce a problem in which XAI methods may objectively err (*i.e.*, focus on something it should not be focusing on).

On those composed mosaics, we ask a XAI method to provide an explanation for just one of the contained classes and follow its response. In a sort of eye-tracking game, we measure how much of the explanation generated by the XAI is located in the areas corresponding to the target class, quantifying it through the *Focus* score. This score allows us to compare methods in terms of explanation precision, evaluating the capability of XAI methods to provide explanations related to the requested class.

6.2 | Methodology

The *Focus* computation involves three elements. First, one needs an explainability method to be evaluated. Second, the classification model to be explained. And the last elements needed for the *Focus* calculation are the mosaic samples. Let us start with these three ingredients:

1. The explainability method $\mathcal{A}$. This would be the specific technique the *Focus* is expected to evaluate (*e.g.*, any of the feature attribution methods introduced in Chapter 3).

2. A trained classification model θ. This will be the model the feature attribution method is trying to explain. The model will have been trained from a specific architecture on a dataset and using a certain training configuration.

3. And a set of mosaic samples. To formalize mosaics, let us remember the formalization of the classification problem introduced in the Introduction 1. Where a dataset $\mathbb{D}$ is composed by a set of images $\mathbb{I} = \{img_1, img_2, \ldots img_N\}$ and a set of classes $\mathbb{C} = \{c_1, c_2, \ldots c_K\}$, N being the number of total images and K the number of total classes. And every image in $\mathbb{I}$ has assigned a unique class from $\mathbb{C}$: $c(img)$. From here we build a set of mosaics $\mathbb{M} = \{m_1, m_2, \ldots m_T\}$ where T is the total number of mosaics in $\mathbb{M}$. A mosaic m is composed by J images $m = \{img_1, img_2, \ldots img_J\}$ and characterized by a target class $tc = c(m)$, the specific class the XAI method is expected to explain. While half of the images of the mosaic belong to the target class, the other half will be randomly selected from the rest of the classes. A toy example of a mosaic is shown in Figure 6.1: this mosaic comprises images from a three-class dataset (cats, dogs and rabbits). The size of this mosaic is four $J = 4$ (*i.e.*, the mosaic is composed of a two by two non-overlapping

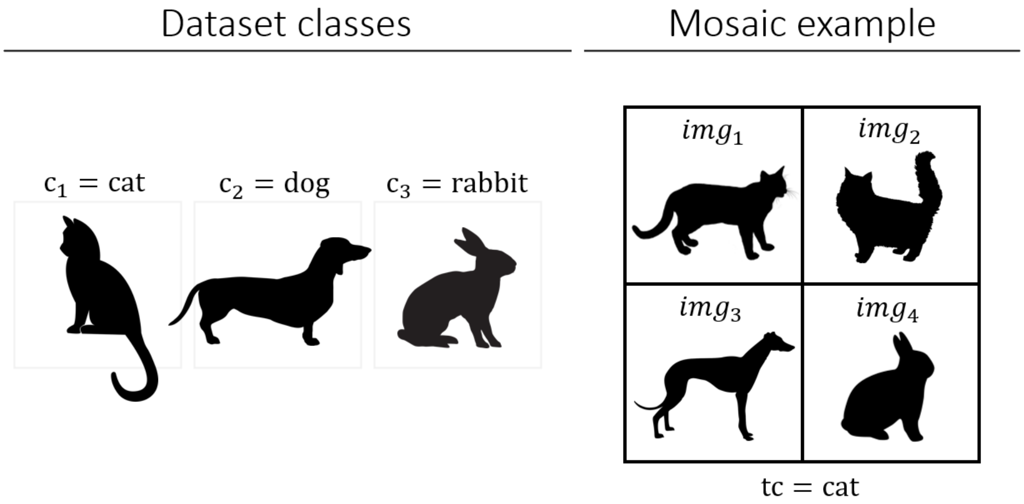


Figure 6.1: Toy mosaic example. This mosaic is composed of images from a three-class dataset: cats, dogs and rabbits. The *target class* of this mosaic is the cat class; therefore, the mosaic is composed of two cats, one dog and one rabbit.

grid of images). The *target class* of this mosaic is the cat class $tc = cat$. Therefore, the mosaic is composed of two images of cats $c(img_1) = c(img_2) = tc$ and the other two images correspond to a dog $c(img_3) \neq tc$ and a rabbit $c(img_4) \neq tc$.

At this point, we have already introduced the different elements needed for the *Focus* computation. Let us now present the *Focus* formulation.

6.3 | FOCUS SCORE

As previously introduced, when a feature attribution method is applied to an image to explain the model's prediction regarding a chosen class, it typically produces a mapping from pixels to real values. In the XAI field this is referred to as feature *relevance*. While some feature attribution methods also provide negative relevance (*i.e.*, the property of certain input features contributing against the prediction of the target class), this is not generalized. Therefore, for this first definition of the *Focus*, we only consider positive relevance. For XAI methods providing both positive and negative relevance (*e.g.*, LRP), only the positive relevance is used, while negative values are treated as zero.

Intuitively, the output of a method is reliable (but not necessarily understandable) when higher values of relevance lie on pixels of the image that are visual evidence toward the chosen class. We consider *visual evidence* any set of pixels used by the model to distinguish the chosen class from any other class of the task. To formalize this, we introduce a probability distribution $\mathcal{P}_{tc}$ over all possible pixels given a *target class* tc. The probability of sampling a pixel from $\mathcal{P}_{tc}$ is proportional to the pixel's relevance toward tc attributed by an explainability method $\mathcal{A}$ and a model θ. Then, we define the formal reliability $Re(\mathcal{A}, \theta, tc)$ as the probability that a pixel sampled from the distribution $\mathcal{P}_{tc}$ lies within visual evidence corresponding to tc.

The definition of $Re(\mathcal{A}, \theta, tc)$ over a method-model-class triplet can be extended to evaluate a method-model pair as $Re(\mathcal{A}, \theta)$. To do so, we take the expectancy of reliability over all classes $\mathbb{C}$: $Re(\mathcal{A}, \theta) = \mathbb{E}_{tc\in\mathbb{C}}[Re(\mathcal{A}, \theta, tc)]$. More accurate models and better feature attribution methods will result in $Re(\mathcal{A}, \theta)$ values closer to 1. The lower bound of $Re(\mathcal{A}, \theta)$ is the probability that any pixel lies within evidence, which is proportional to the number of pixels lying on visual evidence.

To obtain the $Re(\mathcal{A}, \theta)$ metric, we would require a ground truth of which pixels are evidence toward a class. A way to bypass this limitation is to take the assumption that evidence toward a class is more prevalent in images labelled with that class, this being the main assumption of the proposed approach. We thus define the *Focus* as an estimator of the reliability computed over a dataset. The *Focus* evaluates the expected probability that a pixel sampled from $\mathcal{P}_{tc}$ lies on an image of the *target class tc*. Notice the *Focus* underestimates the reliability, as evidence toward a class can be present on samples of a different class of the dataset. We leverage this to our advantage in §10.1, using it to detect *biases* in models and datasets (be they desirable or undesirable biases).

Since this new score only requires image labelling instead of pixel labelling, we transform the dataset into a set of mosaics as introduced in §6.2. As such, we compute *Focus* on subsets of J images (*i.e.*, each image composing the mosaic is labelled) to estimate the *Focus* of a method and a model on the whole dataset. In this context, the *Focus* score estimates the reliability of XAI method's output as the probability of the sampled pixels lying on an image of the *target class* of the mosaic. This is equivalent to the proportion of positive relevance lying in those images:

$$Focus_{\mathcal{A},\theta}(m) = \frac{R_{tc}(img_1) + \ldots + R_{tc}(img_{J/2})}{R_{tc}(m)} \tag{6.1}$$

where $R_{tc}(r)$ is the sum of positive relevance toward class c on the region of the mosaic r. And $\{img_1, img_2, \ldots img_{J/2}\}$ are the *target class* images withing the mosaic.

This probability can be interpreted as the precision of the relevance. In practice, using the *Focus* is analogous to asking the XAI method "*Why does mosaic m belong to the target class?*" on a mosaic m which contains both samples belonging and not belonging to the *target class*. Given the previous question and a good underlying model, a reliable feature attribution method should be able to concentrate most of its explanation relevance on the appropriated images of the mosaic.

As explainability becomes more reliable, the *Focus* will grow. As with reliability, the theoretical upper bound of the *Focus* score is 1, but this is unrealistic: *visual evidence* of a class appearing exclusively on images of that class is seldom true. On the other hand, in the case of uninformed relevance attribution (*i.e.*, unreliable explanations), the expected value of *Focus* is 0.5, since the probability of picking a pixel of the correct class is just the prior probability of picking one of the pixels of $\{img_1, img_2, \ldots img_{J/2}\}$, which amount to half of the total pixels in the mosaic.

6.4 | SANITY CHECKS

In this section, we introduce different experiments performed with two main objectives: first, to reach the final *Focus* design and mosaic construction proposal, and second, to test the robustness and consistency of the method.

6.4.1 | MOSAIC CONSTRUCTION

First, we conduct a randomization experiment to assess and decide the exact position of the *target class* images within the mosaic grid. This experiment uses GradCAM [89] on top of a VGG16 [97] model trained for the Dogs vs Cats[2] dataset (pre-trained on ImageNet [85]). We used mosaics of size four $J = 4$. The six possible configurations of the two by two grid were tested, plus a seventh for random positioning. For each configuration, 2,812 mosaics were created, using *cat* class as the *target class*. The resulting *Focus* distributions are shown in Figure 6.2. Clearly, the positioning of target samples affects the *Focus* distribution. Configurations where the two target class images (img_1 and img_2) are arranged contiguously tend to be better. While this may be partially the result of explanation relevance spilling over samples, it happens more prominently when correct samples are placed on top. Meanwhile, the left-right configurations show a smaller gain when placing the correct samples on the right. We hypothesize that such variance in *Focus* performance is independent of the underlying XAI method and is instead caused by particularities of the dataset and/or task. Since we cannot

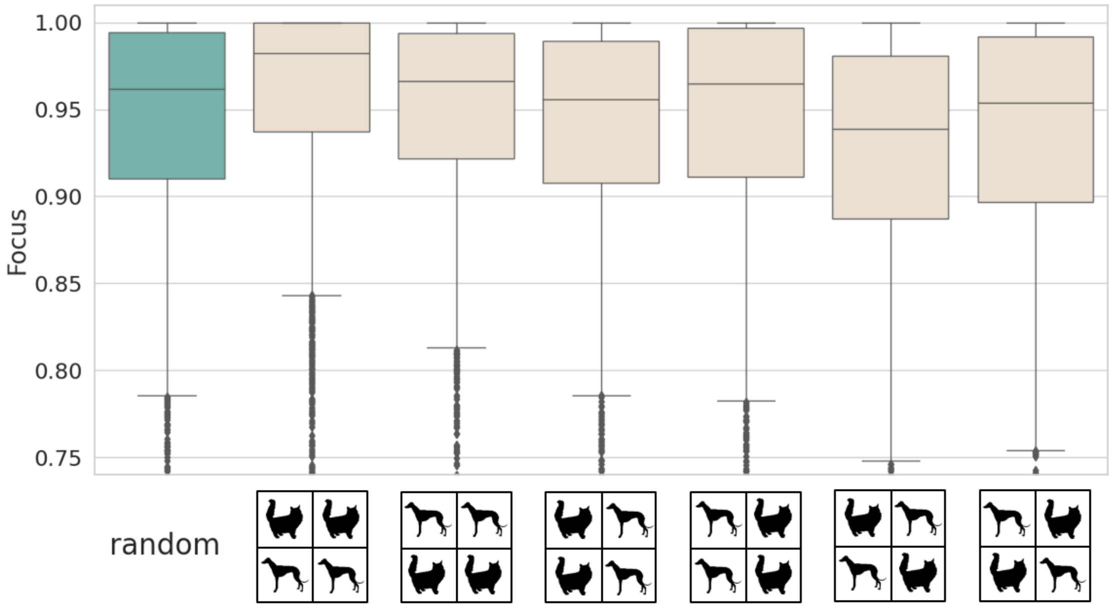


Figure 6.2: *Focus* obtained by GradCAM on a VGG16 trained for Dogs vs Cats dataset (pre-trained on ImageNet), using different mosaic configurations. Each box plot shows the distribution of *Focus* obtained from evaluating 2,812 samples for each configuration (the cat being the *target class*).

[2] https://www.kaggle.com/c/dogs-vs-cats/overview

guarantee that these properties will hold among target classes, datasets or models, we propose to use a sampling approach hereafter. The exact position samples within the composed grid are chosen randomly for every mosaic.

When building the mosaics, we must also set the number of *target class* images within each mosaic. We decided to build the mosaics with half of the images belonging to the target class and the other half not belonging to the *target class*. This construction decision was motivated to avoid a low *Focus* due to missing evidence in a particular instance. In other words, if the *target class* image of a specific mosaic does not contain clear evidence for the *target class*, we will be penalizing the evaluation of the *feature attribution* method, even though the XAI method is being faithful to the model. To test this hypothesis, we perform the following experiment. We created two mosaic variations of size four $J = 4$. The first version with two images belonging to the *target class* and the second with only one *target class* image. The results support our hypothesis (see Figure 6.3): the *Focus* decreases when using the one vs three format. To prevent the evaluation

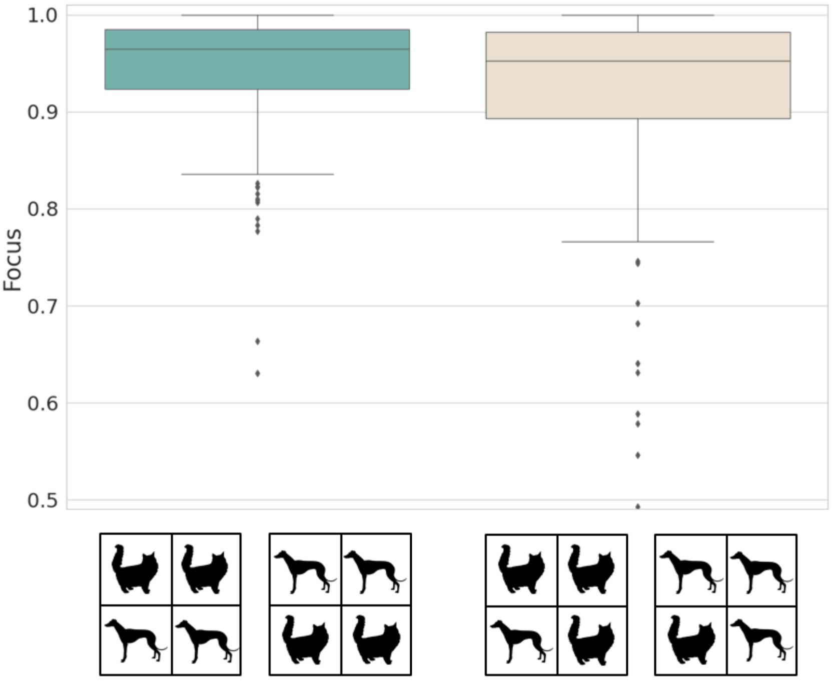


Figure 6.3: *Focus* obtained by GradCAM on top of a VGG16 model fine-tuned for Dogs vs Cats dataset. Two different mosaic variations are tested. On the left, two *target class* images vs two non *target class* images (*i.e.*, two dogs vs two cats). On the right one *target class* image vs three non *target class* images (*i.e.*, one dog vs three cats or one cat vs three dogs).

from being penalized due to a lack of evidence in some instances, we use more than one *target class* image within the mosaic.

6.4.2 | RANDOMIZATION TEST

In this test, we evaluated model randomisation's effect on the *Focus* score. For performing this experiment, we used two different models. A VGG16 pre-trained on ImageNet and then fine-tuned for the Dogs vs Cats dataset and a totally randomized VGG16 model. The experiment computes the *Focus* metric on the cat *target class* ($tc = cat$) for the 2,812 mosaics with the random layout. The distribution of *Focus* achieved by GradCAM on both models are shown as histograms in Figure 6.4. While the mean of the *Focus* obtained with the pre-trained model reaches a remarkable 0.94, the random model mean score is 0.49, roughly 50% of the relevance lies on the wrong class quadrants.

To take the randomization analysis further, we replicate the experiment of Adebayo *et al.* [2]. In it, the authors qualitatively pointed out how visual explanations can be compelling to the eye even when randomizing one or more layers of the underlying model. In this experiment, layers are randomized in cascade, starting with only the top layer, and increasingly randomizing more layers one by one until obtaining a fully randomized model. We use GradCAM on InceptionV3 [104]

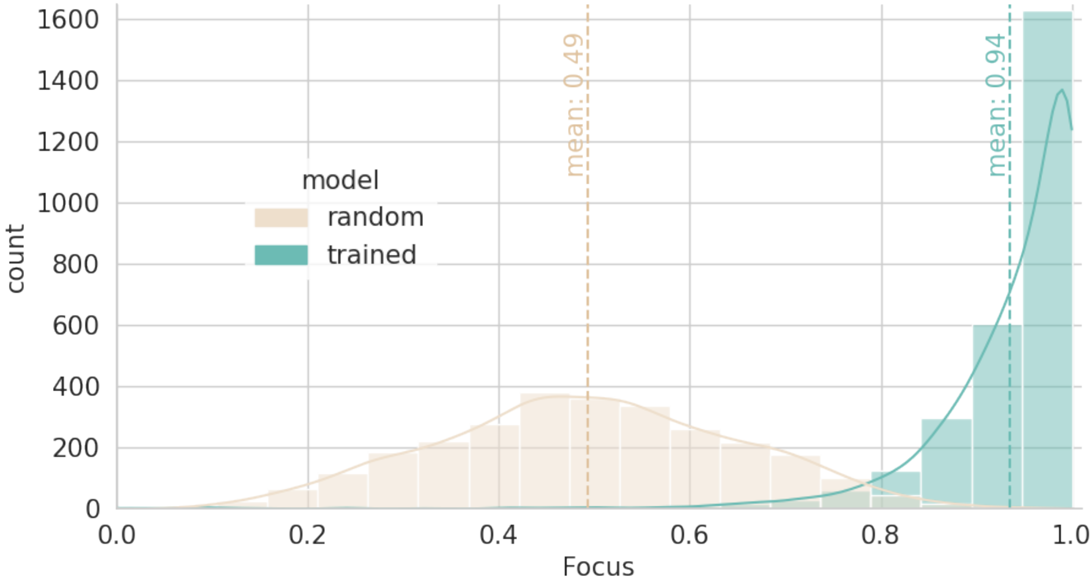


Figure 6.4: Histogram of *Focus* scores obtained by GradCAM from 2,812 mosaics, using a VGG16 trained on Dogs vs Cats and a randomized VGG16 model. The corresponding PDF estimation is represented by a contour line on top.

(like [2]) adding as well VGG16 and ResNet-18. Our results are straightforward: simply randomizing the top layer (or any other set of layers) makes the *Focus* drop to a 50% mean, the same score obtained by a purely random XAI method. This illustrates how resistant the *Focus* score is against misleading explanations.

6.5 | SUMMARY OF THIS CHAPTER

In this chapter, we introduced the *Focus*: a new score to assess the reliability of feature attribution methods. As already discussed, the difficulty of evaluating those XAI methods lies in the non-existence of ground truth. To overcome that barrier, we defined a pseudo-ground truth by constructing new instances from samples of different classes, we called them *mosaic images*. Given the mosaics, a model and a reliable feature attribution method, the *Focus* computes the proportion of the total explanation lying on the ground truth (*i.e.*, in the *target class* images). Mosaics allow us to assess the faithfulness of the feature attribution methods without disturbing the input image. In other words, the features found within the original data distribution are preserved. Finally, we also performed some sanity checks to verify the robustness of the *Focus*.

7 | FOCUSING ON XAI METHODS

> Science, for me, gives a partial explanation for life. In so far as it goes, it is based on fact, experience and experiment.
>
> *Rosalind Franklin*

Considering the problems we had identifying the most appropriate XAI method in the MAMe use case, in this chapter we tackle the assessment of different feature attribution methods following the definition of the *Focus* and mosaics presented in Chapter 6. This contribution was first introduced in [8][1].

7.1 | EXPERIMENTS DETAILS

As previously introduced, *Focus* requires an explainability method, a trained classification model and a set of mosaic samples. We detailed below the different explainability methods evaluated (§7.1.1), the models explained (§7.1.2) and a further explanation regarding the construction of the mosaics (§7.1.3).

7.1.1 | EXPLAINABILITY METHODS

Of all the feature attribution methods presented in Chapter 3, we select those more frequently used. The details on each one of them being as follows:

- GradCAM [89], based on the Gildenblat *et al.* implementation[2]. We compute the gradients of the logits of the class with respect to the feature maps of the final convolutional layer. That is the 5^{th} layer for AlexNet, the 13^{th} for VGG16 and the last layer from the 5^{th} block for ResNet-18 (also known as block E).

[1] Part of this chapter can also be found in that work.

[2] https://github.com/jacobgil/pytorch-grad-cam

- LRP [12], based on the implementation of Nam *et al.* [66]. On the first layer, we use the $z^B - rule$ [63], on fully connected layers, the $LRP - \epsilon$ [12], and on convolutional layers, the $LRP - \alpha\beta$ [12] with $\alpha = 1$ and $\beta = 0$.
- SmoothGrad [99], based on the implementation of Nakashima *et al.*[3]. Explanations are obtained by computing the gradient of the specific class score with respect to the input pixels and adding small perturbations on the input image (in our case Gaussian Noise).
- LIME [81], based on the implementation of Tulio *et al.*[4]. Each explanation is computed considering 1,000 samples, and the final explanation only includes the five top features, that is, the five most relevant superpixels.
- GradCAM++ [18], based on the implementation of Gildenblat *et al.*[2]. We use the last convolutional layer to compute the GradCAM++ explanations.
- IG [102], based on the implementation of Kokhlikyan *et al.* [49]. We use the black image as the baseline image and thirty steps to approximate the integral.

7.1.2 | MODELS

One needs a model to explain to run a XAI method. One generated from an architecture trained on a dataset through a specific training configuration. In these experiments, we use the following:

- **Architectures:** AlexNet[51], VGG16[97] and ResNet-18 [36].
- **Datasets:** the Dogs vs Cats[5], the MAMe [76], the MIT67 [78] and the ILSVRC 2012 [85] (hereafter ImageNet).

Training configurations: During training, data augmentation is performed and AMSGrad [80] is used to optimise weights. For the ImageNet dataset, we use the pre-trained models in the subpackage *torchvision.models*[6,7,8]. For Dogs vs Cats and MAMe datasets, we fine-tuned the ImageNet pre-trained models. Finally, in the case of the MIT67 dataset, we fine-tune the model pre-trained on Places365-Standard dataset [115] (models available in the official repository[9]).

[3] https://github.com/kazuto1011/grad-cam-pytorch

[4] https://github.com/marcotcr/lime

[5] https://www.kaggle.com/c/dogs-vs-cats/overview

[6] https://download.pytorch.org/models/alexnet-owt-4df8aa71.pth

[7] https://download.pytorch.org/models/vgg16-397923af.pth

[8] https://download.pytorch.org/models/resnet18-5c106cde.pth

[9] https://github.com/CSAILVision/places365

7.1.3 | MOSAICS

The last elements required to compute the *Focus* score are the mosaics. In these experiments, we set the size of the mosaics to four $J = 4$. Therefore, as defined in §6.4.1, half of the mosaic will belong to the *target class* and the other half to random classes. To maintain the resolution of visual patterns seen during their training, all XAI evaluation experiments use 448×448 mosaics. That is four times the size of the inputs the models were trained with. Also note that both AlexNet and VGG16 architectures were not input-agnostic when originally proposed, being limited by design to an input size of 224×224 pixels. Nowadays, these architectures employ an Adaptive Pooling Layer to circumvent this problem.

We built thousands of mosaics using the four datasets introduced in §7.1.2. Some examples are shown in Figure 7.1. Since the Dogs vs Cats dataset is a binary dataset, the mosaics will be composed of two cats and two dogs, see Figure 7.1 (**a**). In (**b**), we show a mosaic made up of images from the MIT67 dataset. The *target class* of this mosaic is the *airport* class, located in the top-left and bottom-right images. The other two are randomly chosen, in this case belonging to the *staircase* and *greenhouse* classes. In (**c**), the shown mosaic is from the ImageNet dataset, where the *target class* is the *lorikeet* class. The last example (**d**) is composed of images from the MAMe dataset, in this case, the *target class* is the *faience* class. Note that the exact position of the *target class* images is not always the same. As already said in §6.4.1, the position is chosen randomly for each mosaic.

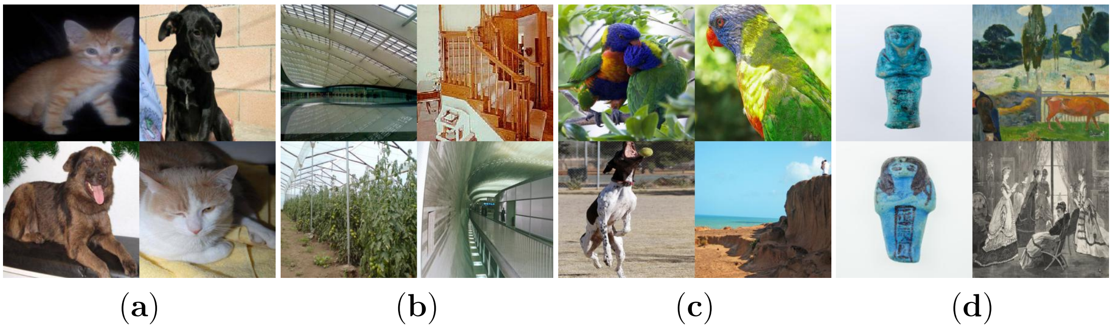


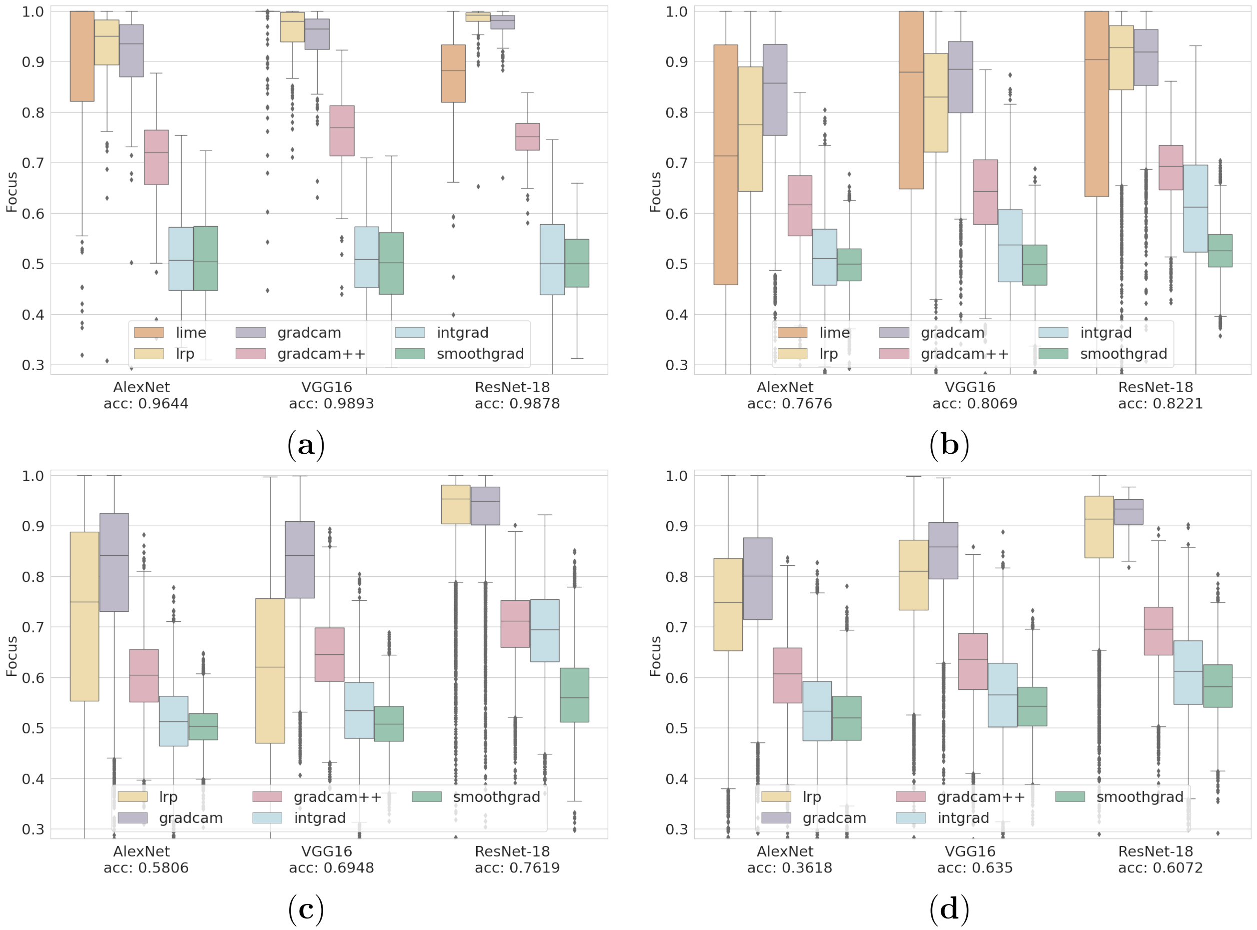
Figure 7.1: Mosaic samples used by the evaluation methodology, obtained for: (**a**) Dogs vs Cats (**b**) MIT67 (**c**) ImageNet and (**d**) MAMe dataset.

7.2 | RESULTS

Let us now put the *Focus* into practice. As previously introduced, we evaluate GradCAM, LRP, SmoothGrad, LIME, GradCAM++ and IG, using three architectures (AlexNet, VGG16 and ResNet-18) and four target datasets (Dogs vs Cats,

Table 7.1: Mean and standard deviation (in parenthesis) of the *Focus* distribution obtained by different XAI methods (columns) on architectures trained for different datasets (rows). The accuracy shown beside each model (*acc*) corresponds to the mean per class accuracy on the validation set. Best mean *Focus* per row in bold.

		GradCAM	LRP	SmoothGrad	GradCAM++	IntGrad	LIME
Dogs vs. Cats	AlexNet - acc: 0.9644	0.9101 (± 0.0903)	**0.9230 (± 0.1018)**	0.5092 (± 0.0840)	0.7041 (± 0.0872)	0.5113 (± 0.0858)	0.8883 (± 0.1797)
	VGG16 - acc: 0.9893	0.9446 (± 0.0577)	0.9526 (± 0.0877)	0.5035 (± 0.0854)	0.7574 (± 0.0777)	0.5108 (± 0.0849)	**0.9724 (± 0.1024)**
	ResNet-18 acc: 0.9878	0.9725 (± 0.0320)	**0.9741 (± 0.1018)**	0.4970 (± 0.0677)	0.7484 (± 0.0456)	0.5037 (± 0.0976)	0.9735 (± 0.0809)
MAMe	AlexNet - acc: 0.7676	**0.8292 (± 0.1346)**	0.7237 (± 0.2359)	0.4962 (± 0.0515)	0.6117 (± 0.0879)	0.5138 (± 0.0825)	0.6695 (± 0.2819)
	VGG16 - acc: 0.8069	**0.8556 (± 0.1123)**	0.7827 (± 0.2015)	0.4957 (± 0.0626)	0.6401 (± 0.0932)	0.5354 (± 0.1050)	0.7951 (± 0.2459)
	ResNet-18 - acc: 0.8220	**0.8941 (± 0.0938)**	0.8864 (± 0.1268)	0.5257 (± 0.0521)	0.6874 (± 0.0665)	0.6076 (± 0.1213)	0.7937 (± 0.2533)
MIT67	AlexNet - acc: 0.5806	**0.8133 (± 0.1401)**	0.6864 (± 0.2545)	0.5017 (± 0.0415)	0.6037 (± 0.0773)	0.5121 (± 0.0736)	—
	VGG16 - acc: 0.6948	**0.8230 (± 0.1088)**	0.6033 (± 0.1978)	0.5079 (± 0.0522)	0.6441 (± 0.0776)	0.5340 (± 0.0809)	—
	ResNet-18 - acc: 0.7619	**0.9248 (± 0.0818)**	0.9162 (± 0.1265)	0.5682 (± 0.0807)	0.7027 (± 0.0702)	0.6892 (± 0.0865)	—
ImageNet	AlexNet - acc: 0.3618	**0.7866 (± 0.1179)**	0.7345 (± 0.1442)	0.5194 (± 0.0644)	0.6018 (± 0.0797)	0.5342 (±0.0867)	—
	VGG16 - acc: 0.6350	**0.8426 (± 0.0881)**	0.7914 (± 0.1140)	0.5425 (± 0.0566)	0.6279 (± 0.0814)	0.5637 (± 0.0924)	—
	ResNet-18 - acc: 0.6072	0.8792 (± 0.0849)	**0.8814 (± 0.1068)**	0.5827 (± 0.0608)	0.6885 (± 0.0711)	0.6081 (± 0.0897)	—



(**a**) (**b**) (**c**) (**d**)

Figure 7.2: *Focus* distribution boxplot for different XAI methods applied to models trained for different datasets. The accuracy (*acc*) shown under each model corresponds to the mean per class accuracy on the validation set of the corresponding dataset. These datasets are (**a**) the Dogs vs Cats dataset, (**b**) the MAMe dataset, (**c**) the MIT67 dataset and (**d**) the ImageNet dataset. LIME is only present in (**a**) and (**b**).

MAMe, MIT67 and ImageNet). For the Dogs vs Cats dataset, the MAMe dataset and the MIT67 dataset we use 100 mosaics per *target class*, a total of 200, 2,900 and 6,700 mosaics, respectively. In the ImageNet experiments, a total of 10,000 mosaics are used (ten per target class). Since the LIME method is computationally expensive, we restricted the experiments with this method to the Dogs vs Cats (200 mosaics) and MAMe datasets (2,900 mosaics). For each experiment, Table 7.1 depicts the mean and the standard deviation of the *Focus* distribution. Figure 7.2 shows these distributions as box plots for further insides. Overall, *Focus* seems to be correlated with model accuracy. As models get better, the mean *Focus* goes up, and the standard deviation goes down. However, there are exceptions to this rule, as the ResNet-18 outperforms the *Focus* of others consistently. This indicates that certain architectures produce more precise explanations than others; we further analyze these relations in §7.2.1.

According to these experiments, GradCAM results are the best on average. Reaching a mean *Focus* above 81% in all experiments but one. And obtaining the best results in two-thirds of the conducted experiments. This feature attribution method is particularly robust to noisy models, performing competitively even with 36% accuracy models (AlexNet on ImageNet). GradCAM++ scores significantly lower in every experiment we conducted, being the third or fourth in the overall ranking. Still, its explanations are well above random behaviour.

LRP gets the second best *Focus* in eight of twelve experiments and the best in three of the remaining four. LIME performs very well on the high accuracy models of Dogs vs Cats, outperforming GradCAM. But on the other models, it is able to beat the mean of GradCAM only once, while suffering from a significantly larger variance. The worst results of LRP are obtained in the MIT67 experiment for the AlexNet and VGG16 models. Notice these models were pre-trained on the Places365-Standard dataset [115], which is noticeably narrower than ImageNet (434 vs 1,000 classes). Overall, these results indicate LRP is a very good methodology for explainability, particularly when applied to very accurate models.

LIME performs remarkably well for the Dogs vs Cats models, the ones with the highest accuracy (pre-trained with ImageNet), and the only two-class classification task. For lower accuracy models (AlexNet in this task, and all in MAMe task), LIME becomes less reliable. Its mean *Focus* drops, and its standard deviation becomes the largest of all XAI methods. The lack of hyperparameter tuning may have penalized the results for MAMe.

SmoothGrad generally obtains a *Focus* of around 50%, showing close to random precision in all experiments. Since this method uses the gradient of the output with respect to the input pixels, misleading attribution scores could be caused by discontinuous gradients or by saturation of gradients, as previously suggested [93]. The IG method tries to overcome these drawbacks and while its mean score is always better than the SmoothGrad, it remains quasi-random in general. The cause behind these noisy explanations may be the domination of gradients in saturated areas, as shown by Miglani *et al.* [60].

7.2.1 | FOCUS SCORE RELATIONS

The previous experiments show a strong relationship between the model performance and the *Focus* score. To further validate such correlation, we evaluate the evolution of the *Focus* metric and its corresponding accuracy during a model training process. In particular, we extract *Focus* after every training epoch, plotting the median of the corresponding distribution. The *Focus* metric is evaluated using the most stable explainability method according to our results, the GradCAM method. Regarding the model, we use the ResNet-18 architecture trained with the Dogs vs Cats dataset. This experiment is performed under two different setups: training from scratch and training on top of an ImageNet pre-trained model. Results of this experiment are shown in Figure 7.3, illustrating a strong correlation between *Focus* and model performance. For the training from a pre-trained model (top plot), the Pearson correlation coefficient is 0.9939 and, for the training from scratch case (bottom plot), the Pearson correlation coefficient is 0.9873. Additionally, the variance is shown as a shaded area around *Focus* score. Notice how the *Focus* metric reduces its variance while the model's training converges.

Beyond the relation between *Focus* and model performance, there are other factors affecting the *Focus* outcome. Among the ones we consider are architectures and datasets (having fixed GradCAM as the XAI method). To assess their impact, we look at all the experiments conducted in §7.2, plotting their median *Focus* scores with respect to the model performance in two plots (Figure 7.4): one plotting *Focus* versus accuracy (a) and another plotting *Focus* versus loss (b).

Let us first discuss model architecture. *Focus* typically grows with model performance. However, there are cases where a better performance does not entail a better *Focus*. The following examples illustrate such a scenario, always when switching from VGG16 to ResNet-18 architecture while using the same dataset. For the MAMe dataset (pink and grey circles), the accuracy slightly increases when moving from VGG16 to ResNet-18 (+1.52%) while *Focus* grows considerably (+3.39%). For the Dogs vs Cats dataset (pink and grey stars), the accuracy remains roughly equal (-0.15%) while *Focus* increases (+1.77%). Finally, and most remarkably, in the ImageNet dataset (pink and grey squares), the accuracy degrades (-2.78%) when moving from VGG16 to ResNet-18 while the *Focus* significantly improves (+4.03%). According to these results, when switching from VGG16 to ResNet-18 we can expect an improvement in *Focus*, even when such models are almost equally good at the underlying task. This particular experiment showcases the relevance of architecture for the *Focus*, and for XAI.

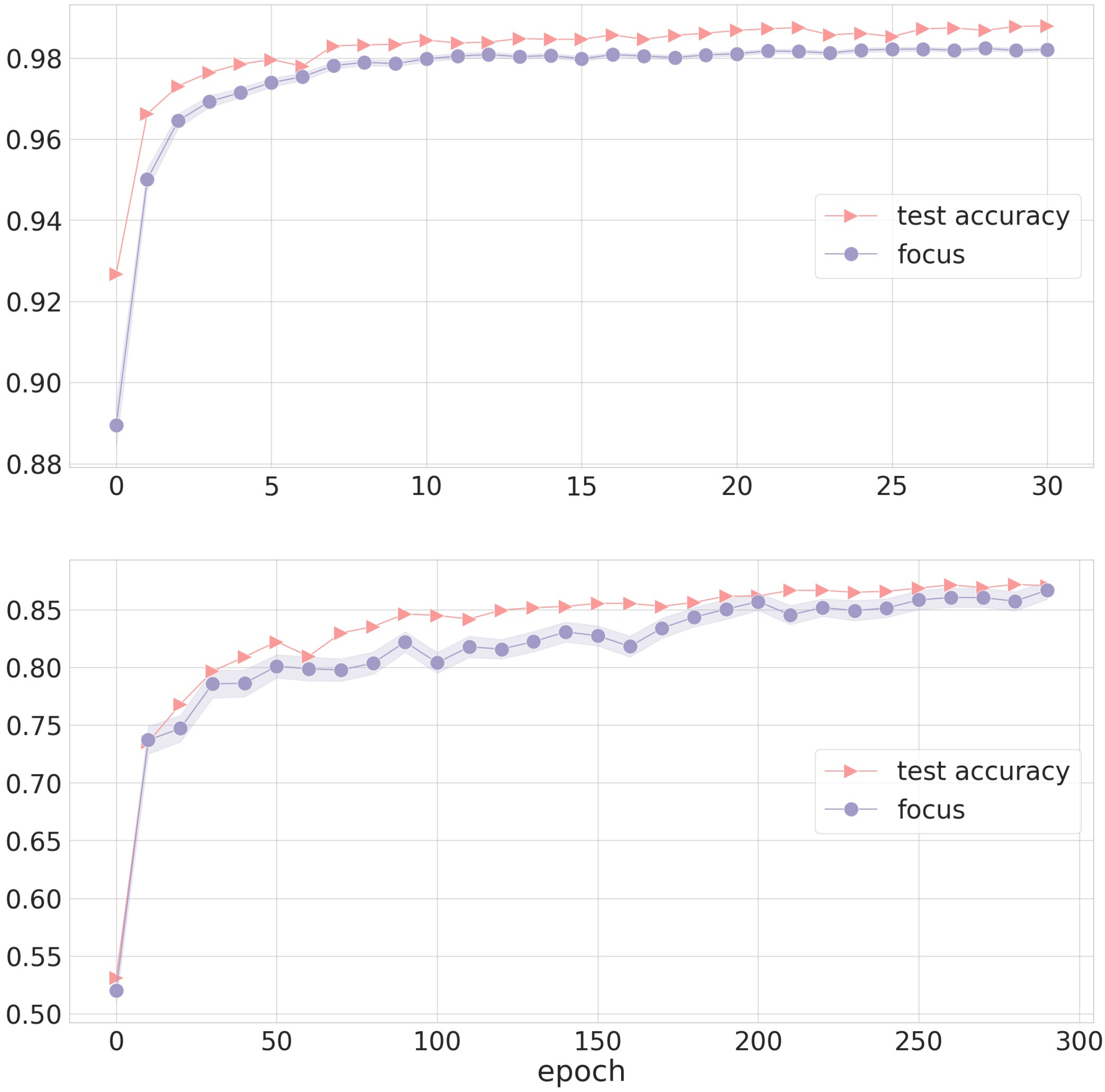
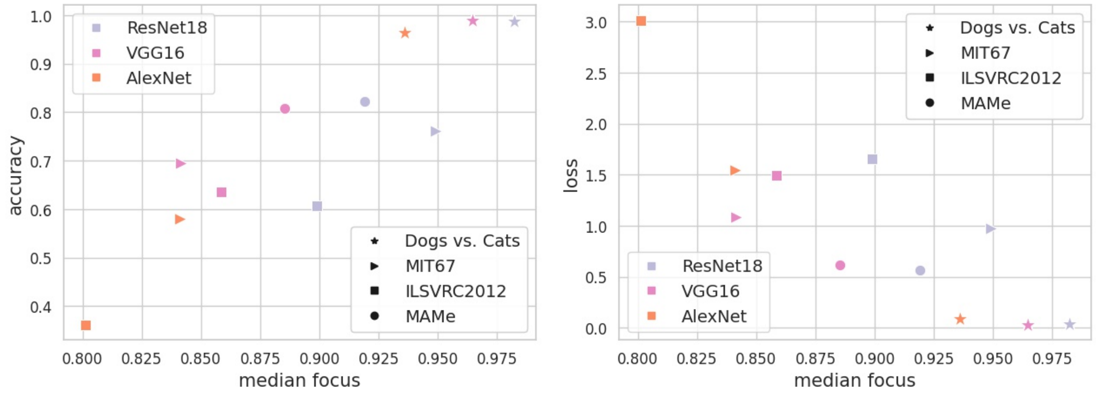


Figure 7.3: Model's training curves as pink triangles, the median of the *Focus* distribution in purple circles (variance as shaded area). Both curves correspond to a ResNet-18 trained on the Dogs vs Cats dataset. The top plot corresponds to a model pre-trained on ImageNet, while the bottom plot corresponds to a model trained from a randomized initialization.

The other main factor influencing the *Focus* score according to our experiments is the target task (*i.e.*, the dataset). There are two cases where having the same architecture but a different dataset, model performance degrades while *Focus* improves, showing the effect of the dataset itself. The first example is for ResNet-18 trained for either MAMe or MIT67 (grey circle and grey triangle). While for the



(**a**) (**b**)

Figure 7.4: Model accuracy (**a**) and model loss (**b**) vs median of the *Focus* distribution for different experiments. Each architecture is shown in a different colour, and each classification task is represented with a different marker.

MIT67 accuracy is worse (-6.02% acc), the same model outperforms the MAMe one in terms of *Focus* ($+2.98\%$). The second case involves VGG16 models trained for MIT67 and ImageNet (pink triangle and pink square), where, when comparing both models, the latter's accuracy degrades by a -6.98% while its *Focus* score improves a $+1.71\%$. These results illustrate the impact of the dataset on the *Focus* score. Several factors may be at play here, including the number of classes in the task, how fine-grained or varied these are, the pre-training used, and the training set size.

7.3 | SUMMARY OF THIS CHAPTER

In this chapter, we apply the *Focus* methodology introduced in Chapter 6. We use the *Focus* to evaluate six feature attribution methods, using different configurations (*i.e.*, different architectures and different datasets).

The main findings described in this chapter are the following. When applied to SmoothGrad or IG, *Focus* finds these methodologies as quasi-random in their explanations with respect to the model. On the contrary, LRP and GradCAM are both found to be consistently reliable methods. GradCAM performs well on all experiments conducted, even when the underlying model is not particularly well fit for the task. LRP performs very well for high-performing models, but it becomes more unreliable on less accurate models. This also seems to be the case of LIME, which suffers from an even larger variance. GradCAM++ performs better than random, but not as well as GradCAM and LRP.

The *Focus* results are rather consistent across tasks and architectures, providing empirical evidence of their performance. The consistency of *Focus* is likely related to the type of noise it induces. By altering the context and not the content of samples, *Focus* adds and exploits in-distribution noise. Unlike out-distribution noise, this is less prone to arbitrary model behaviour.

Notice that by selecting the method that best represents the model behaviour according to the *Focus* (*i.e.*, GradCAM in the case of the experiments conducted in this chapter), would allow to reduce the XAI BIAS present in the bias chain.

8 | ATTRIBUTION CONFUSION MATRIX

> I think that little by little I'll be able to solve my problems and survive.
>
> *Frida Kahlo*

In previous chapters, we introduced *Focus* and showed how it can help researchers choose the best feature attribution method according to this score, thus reducing the XAI BIAS. We also showed the relation between model performance and *Focus* score, which supports the reliability of the method. However, the *Focus* as previously proposed has two major limitations. On one hand, it only considers the positive relevances: some of the feature attribution methods also provide negative relevances (*i.e.*, features that provide evidence against the *target class*), see Figure 8.1. On the other hand, the *Focus* may suffer from numerical instabilities caused by zero divisions (*e.g.*, when all the attributions are negative). In this chap-

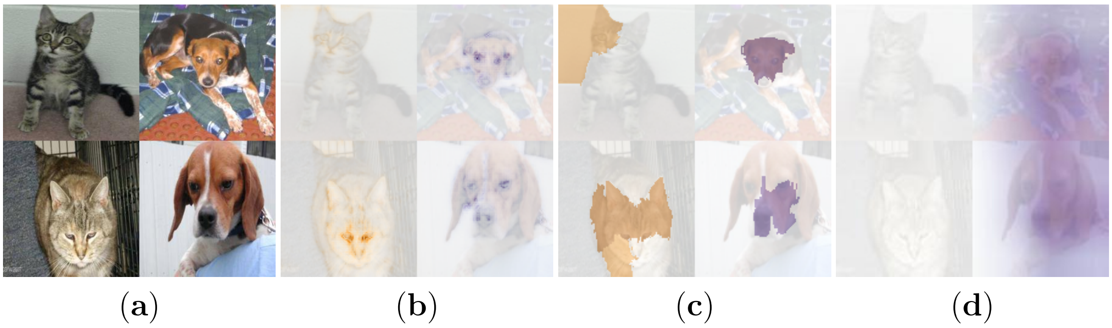


Figure 8.1: (**a**) Mosaic example made up of images from the Dogs vs Cats[5] dataset. On the right, the explanations for the target class *dog* are obtained with: (**b**) LRP (**c**) LIME and (**d**) GradCAM. Purple areas correspond to positive attributions and orange to negative ones. Notice that GradCAM only provides positive attributions. The model used was a ResNet-18 architecture pre-trained on ImageNet and fine-tuned on the Dogs vs Cats dataset.

ter we introduce an improved version of the *Focus* to overcome these limitations. This contribution was presented in [7][1].

8.1 | METHODOLOGY

By using mosaics, it is possible to generate a pseudo ground truth that allows us to go from attributions to classification scores, categorizing the input mosaic regions into *relevant* and *non-relevant*. Through the assumption that these labels align with the positive and negative ground truth, we can derive the most commonly used metrics in the evaluation of classification models. Let us first define T as the set of images belonging to the target class within the mosaic, N as the set of images not belonging to the target class, and α_i as the feature attributions. Therefore, for each mosaic, we define:

- True Positive evidence (TP) $= \sum_{i \in T} \max(0, \alpha_i)$
- False Positive evidence (FP) $= \sum_{i \in N} \max(0, \alpha_i)$
- True Negative evidence (TN) $= \sum_{i \in N} |\min(0, \alpha_i)|$
- False Negative evidence (FN) $= \sum_{i \in T} |\min(0, \alpha_i)|$

Having defined the TP, FP, TN and FN terms, the confusion matrix can be used as a performance measurement tool. We redefine the existing metrics for classification as Focus-X:

- Focus-Accuracy $= \frac{TP+TN}{TP+TN+FP+FN}$
- Focus-Precision $= \frac{TP}{TP+FP}$
- Focus-Recall $= \frac{TP}{TP+FN}$
- Focus-F1 $= \frac{2 \times TP}{2 \times TP+FP+FN}$

The use of these metrics, adapted from classification, enriches the evaluation of feature attribution models, detailing how different methods may hold greater relevance in certain scenarios. For instance, in certain medical applications, we could prioritize feature attribution methods which minimize the number of false positives (*i.e.*, high Focus-Precision) to avoid unnecessary treatments when these are highly invasive. Or prioritize attribution which minimizes false negatives (*i.e.*, high Focus-Recall) to avoid non-diagnosed pathological cases in a triage setting.

[1] Part of this chapter can also be found in that work.

This enables a more informed and adecuate decision from practitioners when identifying the most suitable XAI method for their particular requirements. It is worth noting that the original *Focus* metric is equivalent to Focus-Precision, thus inheriting its strengths and weaknesses.

8.2 | EXPERIMENT DETAILS AND RESULTS

Let us now use the metrics previously introduced to evaluate different feature attribution methods. The experiments in this section use similar setups to those detailed in Chapter 7 for the sake of consistency. We restricted the experiments to two of the three models described in §7.1.2 (*i.e.*, VGG16 and ResNet18) and to three datasets Dogs vs Cats, MAMe and MIT67 dataset. The feature attribution methods evaluated in this experiment are LIME, LRP, GradCAM and IG. The implementation of the last three is the same as the one described in §7.1.1. Notice that the LIME implementation has been modified to also generate negative relevance, this is the main difference in the experimentation setup. It is also based on the Tulio *et al.* implementation[4], however in this case, for each explanation, 1000 samples are used, and only the six superpixels with the largest attribution in absolute value are considered. Notice that while LIME, LRP and IG provide both positive and negative relevance, GradCAM only generates positive attributions. We used the GradCAM method as baseline given its top performance. The evaluation results of the feature attribution methods following the detailed metrics are shown in Table 8.1. For each target class: 100 mosaics were built for the Dogs vs Cats dataset (a total of 200), 100 mosaics for the MAMe (a total of 2,900) and 10 mosaics for the MIT67 (670 in total).

Among the methods obtaining positive and negative relevance (*i.e.*, LIME, LRP and IG), for the Dogs vs Cats task (high-performing models with accuracies of 98%) LIME consistently obtains the best scores on all measures. LRP gets the second position obtaining competitive Focus-Precision scores, but lower Focus-Recalls (thus making more false negative predictions). Lastly, IG gets random results on all metrics. Note that IG results may vary depending on the number of steps and the baseline image used (in this experiment we used 30 steps to approximate the integral and the black image as baseline). For the MAMe results (*i.e.*, models with lower accuracy, in the range 80-82%) LIME shows lower performance in all the metrics with respect to the simpler Dogs vs Cats task, probably due to model performance drop. This also affects the variance, which increases in all metrics, particularly in Focus-Precision (unreliable amount of false positives). Regarding the Focus-Recall scores, both LIME and LRP maintain high mean values. Finally, for the MIT67 task (*i.e.*, models with the lowest performance, in the range 69-76%) LIME performs better than LRP for all metrics, particularly in VGG16,

Table 8.1: Mean and standard deviation of the four metrics computed: Focus-Precision, Focus-Accuracy, Focus-Recall and Focus-F1. Each metric is shown grouped by column and each row shows the results for a combination of a feature attribution method, a specific architecture and a target task. For each model, the metric obtaining the highest mean is highlighted in bold.

			Focus-Precision	Focus-Accuracy	Focus-Recall	Focus-F1
Dogs vs Cats	VGG16 acc: 0.9893	LIME	**0.9935 (± 0.0724)**	**0.9913 (± 0.0435)**	**0.9863 (± 0.0746)**	**0.9855 (± 0.0859)**
		LRP	0.9526 (± 0.0877)	0.9343 (± 0.0835)	0.9011 (± 0.1707)	0.9141 (± 0.1290)
		IG	0.4973 (± 0.0912)	0.5038 (± 0.0011)	0.5039 (± 0.0015)	0.4963 (± 0.0471)
		GradCAM	0.9446 (± 0.0577)	-	-	-
	ResNet-18 acc: 0.9878	LIME	**0.9913 (± 0.0739)**	**0.9853 (± 0.0786)**	**0.9796 (± 0.1131)**	**0.9776 (± 0.1154)**
		LRP	0.9741 (± 0.1018)	0.9729 (± 0.1012)	0.9690 (± 0.1142)	0.9707 (± 0.1066)
		IG	0.4937 (± 0.0802)	0.5018 (± 0.0006)	0.5019 (± 0.0008)	0.4944 (± 0.0419)
		GradCAM	0.9725 (± 0.0320)	-	-	-
MAMe	VGG16 acc: 0.8069	LIME	0.7987 (± 0.2603)	**0.8048 (± 0.2373)**	**0.9490 (± 0.1757)**	**0.8359 (± 0.2333)**
		LRP	0.7827 (± 0.2015)	0.7913 (± 0.1967)	0.9103 (± 0.2200)	0.8311 (± 0.2001)
		IG	0.5354 (± 0.1050)	0.5043 (± 0.0023)	0.5065 (± 0.0035)	0.5152 (± 0.0512)
		GradCAM	**0.8665 (± 0.1123)**	-	-	-
	ResNet-19 acc: 0.8220	LIME	0.8020 (± 0.2520)	0.7987 (± 0.2422)	0.9632 (± 0.1508)	0.8443 (± 0.2205)
		LRP	0.8864 (± 0.1268)	**0.8913 (± 0.1237)**	**0.9866 (± 0.0786)**	**0.9292 (± 0.0989)**
		IG	0.6076 (± 0.1213)	0.5027 (± 0.0015)	0.5041 (± 0.0024)	0.5452 (± 0.0526)
		GradCAM	**0.8941 (± 0.0938)**	-	-	-
MIT67	VGG16 acc: 0.6948	LIME	0.7800 (± 0.2585)	**0.7823 (± 0.2319)**	**0.9390 (± 0.1823)**	**0.8218 (± 0.2280)**
		LRP	0.6012 (± 0.1918)	0.6132 (± 0.1898)	0.6886 (± 0.2231)	0.6367 (± 0.2022)
		IG	0.5262 (± 0.0809)	0.5076 (± 0.0043)	0.5118 (± 0.0057)	0.5157 (± 0.0401)
		GradCAM	**0.8248 (± 0.1076)**	-	-	-
	ResNet-18 acc: 0.7619	LIME	**0.9543 (± 0.1102)**	**0.9302 (± 0.1347)**	0.9611 (± 0.1282)	**0.9492 (± 0.1220)**
		LRP	0.9136 (± 0.1434)	0.9169 (± 0.1417)	**0.9736 (± 0.1240)**	0.9397 (± 0.1307)
		IG	0.6980 (± 0.0910)	0.5034 (± 0.0017)	0.5042 (± 0.0020)	0.5829 (± 0.0334)
		GradCAM	0.9302 (± 0.0749)	-	-	-

with only one exception (Focus-Recall for ResNet-18).

A consistent relevant finding across the experiments is the high Focus-Recall score of LIME and LRP (obtaining a mean greater than 0.9 in all experiments but one). That being said, underperforming models often yield lower precision scores than recall scores, indicating higher reliability of negative relevances with respect to positive relevances. However, in the case of LIME, this feature might be a consequence of the superpixels selection of LIME, since the explanation will only provide negative results, as long as these superpixels have high relevance in absolute value (being among the top 6 most attributed superpixels). This could be important for some case studies, and motivate their use in complement with methods which only provide positive relevance (*e.g.*, GradCAM).

GradCAM generates only positive relevances, so Table 8.1 displays Focus-Precision, because other metrics would be misleading (*e.g.*, since the false negatives always are zero by definition the Focus-Recall score would be always one). GradCAM performs well in all tasks, ranking as the top method in half of the experiments, also obtaining a small variance in general. However, in cases where negative relevance is important, GradCAM applicability is limited.

As stated before, the Focus-Precision (*i.e.*, the *Focus*) sometimes encounters numerical problems. This issue arises when all the attributions are negative, leading to a denominator of zero in Focus-Precision. Conversely, Focus-Accuracy only suffers from this issue when all the attributions are zero. This is reasonable, as the accuracy of an all-zero explanation remains ambiguous.

8.3 | SUMMARY OF THIS CHAPTER

In this chapter, we introduce an extension of the *Focus* score, where we use the widely used metrics in the classification field to evaluate the reliability of the feature attribution methods. The idea is to consider the *target class* images within the mosaic as the correct class and the *non-target class* images as the incorrect class. From this classification problem, we define:

- The TP as the amount of positive relevance that falls in the correct class.
- The FP as the amount of positive relevance that falls in the wrong class.
- The TN as the amount of negative relevance that falls in the wrong class.
- And FN the amount of negative relevance that falls in the correct class.

Then we use the Focus-Accuracy, Focus-Precision, Focus-Recall and Focus-F1 scores to compare and evaluate different existing feature attribution methods. In that regard, we found that among the approaches that produce both positive and negative attributions, LIME consistently achieves the highest scores. IG produces random-like results. And in general, LRP and LIME exhibit high Focus-Recall.

This framework improves some of the *Focus* drawbacks. First, the negative relevance provided by the feature attribution methods can be taken into account. Also, the Focus-Accuracy score overcomes the *Focus* numerical instabilities (*e.g.*, when only negative relevance is found in the explanations). And finally, this methodology can be more suitable in specific use cases where the relevance of false negatives and false positives is distinct (*e.g.*, systems that provide support in analyzing images in medical domains).

9 | TEXTFOCUS

> Language is the road map of a culture. It tells you where its people come from and where they are going.
>
> *Rita Mae Brown*

In previous chapters, we introduce the *Focus*, a score for assessing the reliability of feature attribution methods in the CV field and applied it to image data through the analysis of image classification models. However, the proposed methodology is versatile and applicable to a wider variety of domains. Due to some similarities among unstructured data (*e.g.*, images and text), we consider the extension of *Focus* to the NLP domain. This is what we call the *TextFocus*: an evaluation score for feature attribution methods applied to text classification models. This new score is introduced in [58][1].

9.1 | MODALITY SHIFT

TextFocus is similar to *Focus*, however, since the data type used for the former (*i.e.*, text) is different from the latter (*i.e.*, images), the methodology must be adapted to the properties of this new field. Let us start by formulating the text classification problem to understand the differences that arise when calculating the *TextFocus*. The datasets in this field, instead of being composed of images, will be composed of a set of sentences $\mathbb{S} = \{s_1, s_2, \dots s_N\}$ and a set of classes $\mathbb{C} = \{c_1, c_2, \dots c_K\}$, where $K < N$ and each sentence has a label assigned. Let us imagine we train a model for a text classification task like sentiment analysis (*i.e.*, positive/negative). If we feed the model with a sentence like BY FAR THE WORST MOVIE OF THE YEAR one would expect the model to classify it as a *negative* sentence. Following the proposed methodology, we can use a feature attribution method to understand which words are most relevant to the model's decision. For

[1] Part of this chapter can also be found in that work.

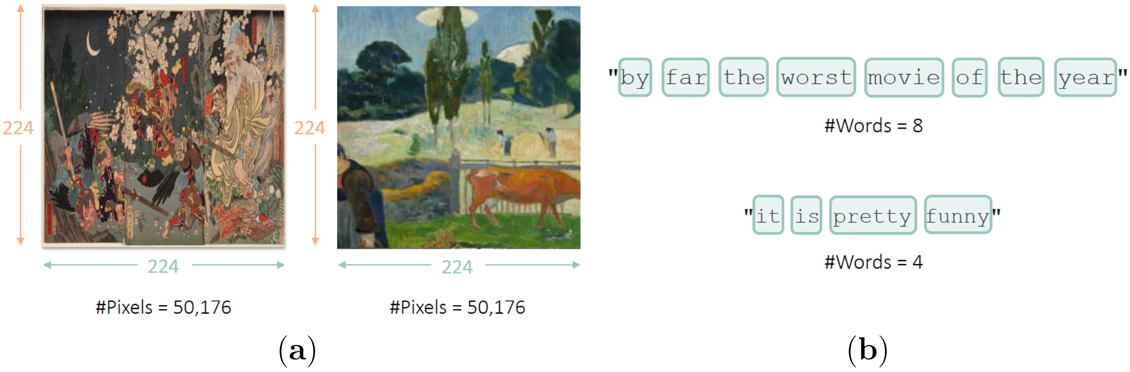


Figure 9.1: (**a**) Two images from the MAMe dataset reshaped to the same size (*i.e.*, same number of pixels). (**b**) Two sentences from the SST-2 dataset of different lengths. The top one is made up of eight words and the bottom one is of length four.

example, in the previous case, the word WORST will probably be highlighted since it will be favouring the *negative* class.

Up to this point, the problem seems analogous to image classification. However, one of the main differences is the variable length of instances. In the case of images, all the images in the training dataset are usually reshaped to the same number of pixels (*e.g.*, $img_1 = \{px_1, px_2, \dots px_P\}$ and $img_2 = \{px_1, px_2, \dots px_P\}$, therefore $|img_1| = |img_2| = \dots = |img_N|$). However, in the text datasets, sentences can be composed by a highly variable number of words $|s_1| \neq |s_2| \neq \dots \neq |s_N|$(*i.e.*, a variable number of tokens), see Figure 9.1. To work around this particularity, the *TextFocus* will be adapted as detailed in §9.3.

9.2 | METHODOLOGY

The elements involving the *TextFocus* computation are analogous to those of the *Focus*. First, the explainability method $\mathcal{A}$. Second, the trained classification model θ. And third, the mosaics.

Let us delve into the *textual mosaics* construction since they present some particularities with respect to the *image mosaics*. These mosaics are made up of J sentences $m = \{s_1, s_2, \dots s_J\}$, where half of them correspond to the *target class* $tc = c(m)$ and the other half did not. To separate the different sentences conforming the mosaic, we use a special token: the `[SEP]` token. And each mosaic begins with a `[CLS]` token and ends with a `[SEP]` token. We use this configuration since the text models used anticipate data in this format.

Examples of the *textual mosaics* structure are shown in Figure 9.2. The first row shows a mosaic of size $J = 2$; a mosaic composed of two sentences. The

SENTENCE 1 corresponds to the *target class* of the mosaic $c(\text{SENTENCE 1}) = tc$, unlike SENTENCE 2 $c(\text{SENTENCE 2}) \neq tc$. In the second row, an example of a mosaic of size $J = 4$ is shown. In this case, we see how two sentences belong to the target class (*i.e.*, $c(\text{SENTENCE 1}) = c(\text{SENTENCE 4}) = tc$) and the other two do not. Note that, same as for *image mosaics*, the position of the target class sentences is not always the same, for each mosaic the position of the target class sentences is chosen randomly.

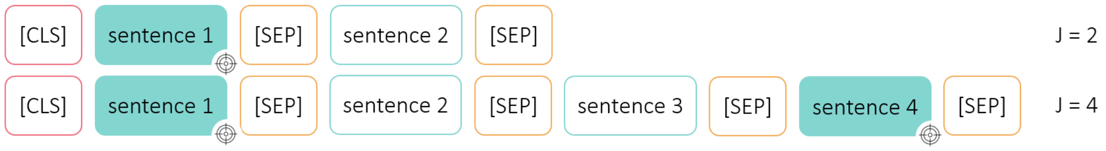


Figure 9.2: Example of the structure of two *textual mosaics*. The first row corresponds to a mosaic of size $J = 2$ and the second row shows a mosaic of size $J = 4$. Each sentence is separated from the next by a `[SEP]` token and each mosaic starts with a `[CLS]` token and ends with a `[SEP]` token. Notice that the target class sentences are highlighted in turquoise.

9.3 | TEXTFOCUS

The *TextFocus* is based on the *Focus* idea: it computes the proportion of positive explanation attribution falling on the target class sentences with respect to the total positive attribution of the mosaic. However, since the length of *textual mosaics* can vary, the score may be affected by these differences resulting in a potentially misleading evaluation.

To better illustrate this problem, we show in Figure 9.3 an example of a mosaic of size $J = 4$ with its corresponding attributions. This *textual mosaic* is composed of sentences from the Standford Sentiment (SST-2) dataset [100]. The target class of the mosaic shown in Figure 9.3 is the *negative* class and the target class sentences are highlighted in yellow: LACKS DRAMATIC PUNCH AND DEPTH and ONE OF THE MOST REPELLENT THINGS. The explanation's attributions are shown in the form of bars. On the right are the positive attributions (in this case tokens that favour the *negative* class): the token LACKS is highly attributed, which means that this token has high relevance for the prediction of the *negative* class. On the contrary, the bars on the left correspond to the negative attributions: the word HILARIOUS has assigned a high negative attribution, which is coherent since it is going against the *negative* class. In this case, since it is a binary classification problem, going against the negative class means favouring the *positive* class. This example illustrates how each sentence may have a different number of tokens, being

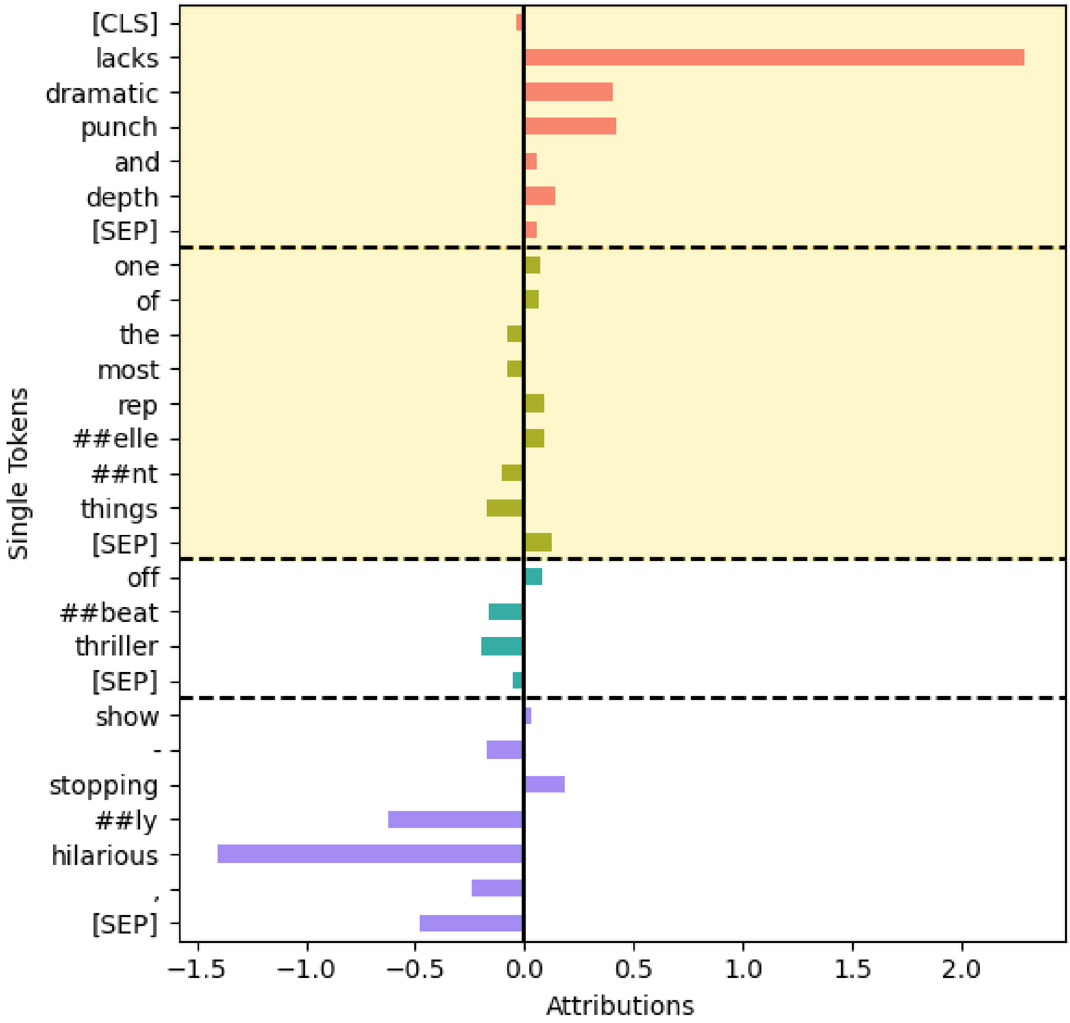


Figure 9.3: Example of a *textual mosaic* of size $J = 4$ being the target class the *negative* class. The two target class sentences are highlighted in yellow. And each sentence is separated from the next by a dashed line. The explanation attributions for each token are shown in the form of bars. The bars on the right represent the positive attributions (*i.e.*, in favour of the target class, the negative class in this example). On the left, the negative attributions (*i.e.*, the tokens against the target class, in this case against the *negative* class).

in this case the largest sentence of eight tokens and the shortest of three. To avoid the *TextFocus* score being biased by the difference in sentence length, we normalize the amount of attribution of each sentence.

$$\eta_i := \frac{R_{tc}(s_i)}{|s_i|} \tag{9.1}$$

where $R_{tc}(r)$ is the sum of positive relevance toward class c on the sentence of the mosaic r and $|s_i|$ corresponds to the size of the set s_i (*i.e.*, the sentence length). Using these normalized attributions we redefine the *Focus* formulation:

$$TextFocus_{\mathcal{A},\theta}(m) = \frac{\sum_{i \in T} \eta_i}{\sum_{i \in M} \eta_i} \tag{9.2}$$

where T is the set of sentences belonging to the target class, and M is the set of sentences of the whole mosaic.

9.4 | EXPERIMENTAL DETAILS

This section details the experimentation setup used for the evaluation of different feature attribution methods through *TextFocus*. First, the different XAI methods evaluated are presented (§9.4.1), and then, the models these XAI methods try to explain as well as the mosaics used are described (§9.4.2).

9.4.1 | EXPLAINABILITY METHODS

In these experiments, we use the XAI methods implemented in Captum (*i.e.*, an open source library for explainability) available for NLP models. Calculating gradients with respect to the input is essential for certain XAI methods, but this is not possible with NLP models that take discrete tokens. To make these methods work, token embeddings are used as inputs instead. The XAI methods evaluated are the following:

- Gradient, a simple call to the gradient of the target function propagated back to the input embeddings. Simonyan *et al.* [96] first used it to explain predictions in CV (see §3.2). Different choices on how to aggregate the attribution lead to different variants (*e.g.*, L1 [53], L2 [10]). We use the L2 variant since it was the one obtaining the best results for discovering one- or two-tokens shortcuts in NLP according to [14].

- Gradient X Activation [24] multiplies the gradient result by the input activation. This is done to represent the degree to which a signal is present or absent.

- IG [102], as introduced in §3.2 this method calculates the integral of gradients through the line joining a baseline x' and input x.

- DeepLIFT [93], the relevance is assigned with respect to a baseline by back-propagating a relevance score through the Rescale Rule (gradient formulation from [4]).
- Gradient SHAP (implementation inspired by the work of [56]) first adds multiple times Gaussian noise to each input instance. Secondly, a random point is chosen along the path between the input and the baseline. Then, the output gradients are computed with respect to the chosen points. Finally, the SHAP values correspond to the multiplication of the difference between the baselines and the inputs with the value of the expected gradients.
- LIME [81], as introduced in §3.1 this method produce perturbed samples of the original dataset. To do so, instead of turning off the superpixels of the input image, in the NLP field, random tokens are deleted from the instance to be explained. These new instances, plus their predictions, are used to train the linear model which approximates the original model locally. The model coefficients include the final attribution scores.
- Feature Ablation first replaces some input values with a baseline and then computes the difference induced in the output.

9.4.2 | DATASETS, MODELS AND MOSAICS

As previously introduced, we focus on the sentiment classification task. To perform the experiments we use two datasets widely known in the NLP field:

- The SST-2 [100] is a binary sentiment classification dataset. The SST-2 is composed of short sentences with a mean of 20 tokens per sentence, labelled as positive or negative. The sentences correspond to parts of movie reviews excerpted from `rottentomatoes.com` and labelled on Amazon Mechanical Turk [100].
- The Internet Movie Database (IMDB) [57] is also a binary sentiment classification dataset. The length of the sentences is longer with a mean of about 300 tokens per sentence. The sentences also correspond to movie reviews collected from `imdb.com`.

We use a DistilBERT model [87] fine-tuned on both the SST-2 dataset and the IMDB [100]. The former reaches an accuracy of 98.9% on the test set and the latter has an accuracy of 92.8%. The models are available in the official repository of Hugging Face[2,3].

[2] https://huggingface.co/distilbert-base-uncased-finetuned-sst-2-english

[3] https://huggingface.co/lvwerra/distilbert-imdb

For the mosaic construction, we set $J = 4$ for the mosaics built with the SST-2 dataset and $J = 2$ for the mosaics made up with the sentences from the IMDB dataset. This reduction in mosaic size is due to the limitation of DistilBERT models, allowing a maximum input size of 512 tokens. Since the IMDB sentences are on average of 300 tokens, we decided to reduce the size of the IMDB mosaics, to avoid truncating them, and select only those that had a length of less than 256 tokens.

9.5 | RESULTS

We conducted the evaluation of the seven feature attribution methods detailed in §9.4.1 using the proposed *TextFocus* score. For the XAI methods requiring baselines, we selected the special tokens that yielded the best results: the `[MASK]` token for Feature Ablation and the `[UNK]` token for IG, Gradient SHAP, and DeepLIFT. For the experiments with the SST-2 dataset, we analyzed a total of 1,746 mosaics with a size of $J = 4$, while for the IMDB, we used 7,014 mosaics with a size of $J = 2$. The results of *TextFocus* are depicted in Figure 9.4.

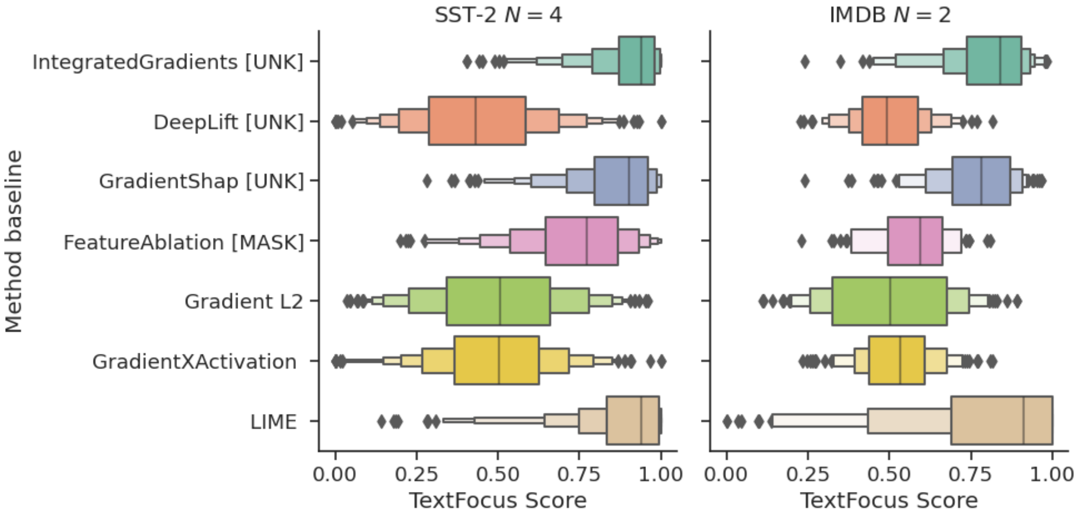


Figure 9.4: The *TextFocus* distributions for the seven XAI methods evaluated on 1,746 mosaics from the SST-2 dataset (left side) and 7,014 mosaics from the IMDB (right side). The square brackets indicate the baselines used.

In both experiments, IG achieves the highest result, with a mean *TextFocus* of 0.905 in the SST-2 case and 0.801 in the IMDB experiment. The LIME and Gradient SHAP methods get the second and third positions, respectively, with a closely mean *TextFocus*: 0.887 and 0.861 in the SST-2 dataset, and 0.789 and

0.762 in the IMDB dataset. Feature Ablation attains a mean *TextFocus* of 0.741 for the SST-2 experiment and 0.569 for the IMDB dataset, ranking fourth overall and being the last method to achieve a mean *TextFocus* above random.

Notice how the *TextFocus* results remain consistent across both datasets (*i.e.*, SST-2 and IMDB). However, a minor decrease in the mean *TextFocus* is noticeable when transitioning from the SST-2 to the IMDB experiment. This decline can be attributed to the disparity in performance between the two models. Specifically, the model fine-tuned on the SST-2 dataset achieves an accuracy of 98.9%, whereas the model fine-tuned on the IMDB dataset achieves a slightly lower accuracy of 92.8%.

In contrast, Gradient L2, GradientXActivation, and DeepLIFT exhibit a mean *TextFocus* of 0.5. This indicates that their explanations are unrelated to the labels assigned to the data, suggesting a lack of effectiveness in attributing relevance to the target class sentences.

9.6 | SUMMARY OF THIS CHAPTER

In this chapter, we introduce how the *Focus* methodology can also be applied in other domains different from the CV field. Specifically, we present how it could be adapted to the NLP domain. We show the main difference between the two data types (*i.e.*, images vs text) and how this affects both the construction of the *textual mosaics* as well as the score computation. We refer to this new score adapted to the characteristics of the NLP domain as *TextFocus*.

Finally, we evaluated seven feature attribution methods using the *TextFocus* score. Among the XAI methods assessed Gradient L2, DeepLIFT, and GradientXActivation demonstrate random-like behaviour. IG with the `[UNK]` baseline is the top-performing method according to *TextFocus*. LIME and Gradient SHAP (utilizing the `[UNK]` token as baseline) also offer faithful explanations. It is worth noting that LIME has an inherent characteristic where the explanations obtained, although reliable, are not deterministic. This means that the specific tokens removed during the computation can slightly alter the explanation. While this feature may not be a disadvantage for certain applications, it could be undesirable for others, especially those that require high reproducibility.

Note that this methodology can be further improved by following the same steps, as with *Focus*, described in Chapter 8. That is, considering the negative attributions and avoiding the possible instabilities generated by the eventual divisions by zero. Implementing these steps will contribute to the refinement and robustness of the methodology.

Part III.

Mosaics for Bias Detection

10 | BIAS IN DATA AND MODELS

> When your big data is corrupted by big silences, the truths you get are half-truths, at best.
>
> *Caroline Criado Perez*

The presence of *bias* in datasets and models is often inherent to their construction. Those biases can be desirable, useful and harmless, or they can be undesirable, useless, and harmful. To build reliable and *fair* models, we must develop tools that facilitate bias detection so that experts can decide if the found biases belong to the former (desirable) or latter (undesirable) category and then take the measures that they deem appropriate.

Suresh and Guttag distinguish seven sources of harm in ML [103]: historical bias, representation bias, measurement bias, aggregation bias, learning bias, evaluation bias and deployment bias. In the CV field, powered by ML, the most common sources are *representation* and *evaluation* biases. The first is due to the need for large datasets that often lack representativeness [73]. The latter is due to a lack of robust evaluations determining the model's capability to correctly generalise in real-world data.

Current data collection methods lead to non-random selection and make the data unrepresentative of the total population. This is not a new problem faced by DL techniques; a decade ago, Henrich *et al.* already claimed that most of the psychological and behavioural studies were based on the Western, Educated, Industrialized, Rich, and Democratic (WEIRD) societies [38]. And therefore, the outcomes from these studies only represented a subpopulation and cannot be generalised to humans worldwide. In this regard, many examples already exist in the DL and CV literature. Shankar *et al.* showed the lack of geo-representation within the ImageNet [85] dataset (*e.g.*, 45.5% of the images were from the United States [90]). Many examples exist of systems which are racially and gender biased such as the three commercial gender classifiers tested in [16], which performed better for white males. The worst performance was obtained when classifying black women (*i.e.*, belonging to two underrepresented populations: women and black). When we use these biased datasets in sensitive fields such as medicine, the consequences can be deadly, for example, diagnosing skin cancer in later stages in patients with dark skin tones [22].

The previous examples are also a symptom of *evaluation* biases since a proper assessment would have highlighted the biased behaviour of the model and prevented its release. As introduced before, this second bias is common in the ML-CV field. On the one hand, the model performance is usually evaluated in a test partition, different from the training and validation sets. However, typically a random split is obtained from the same distribution. Thus, the model is not being evaluated in generalization capability. On the other hand, as also pointed out in [103], the choice of the evaluation metrics could be another source of *evaluation* bias. For example, choosing the method with the best accuracy does not ensure that the method is capable of generalizing better to real-world data or that the method is less biased.

The combination of *representation* biases and *evaluation* biases results in unsafe models that contain an unknown amount of undesirable biases. And what's more, failure to detect these biases can lead models to perpetuate and/or exacerbate inequalities. To prevent that, we need methodologies for identifying and illustrating biases, which experts can use to search and select biases in CV models.

Motivated by these needs, the XAI field has acquired relevant attention in recent years, becoming a tool to provide insights into DL models' behaviour, as we have already introduced in the Introduction 1. However, these techniques present limitations when it comes to detecting biases. These biases typically need to be identified, found and verified by experts. Note that when we refer to experts, we mean experienced people in the task we are trying to solve since they have the available expert knowledge in that field. For example, in the case of the MAMe dataset, these experts would be the art experts. Or, in a medical imaging use case, the experts would be the medical practitioners. However, as we already anticipated in §5.1, even for experts in the field, the manual identification process is a time-consuming task that can induce subjective criteria and confirmation bias (*i.e.*, HUMAN BIAS). For this reason, automation of the bias identification process is needed, on the one hand, to save the expert reviewers time and, on the other hand, to prevent experts from overlooking unwanted biases.

10.1 | USING *FOCUS* TO AUTOMATE THE BIAS IDENTIFICATION PROCESS

As a first approach, we realized that *Focus* and mosaics together could be used to automate the *bias* identification process while providing visual validation to the

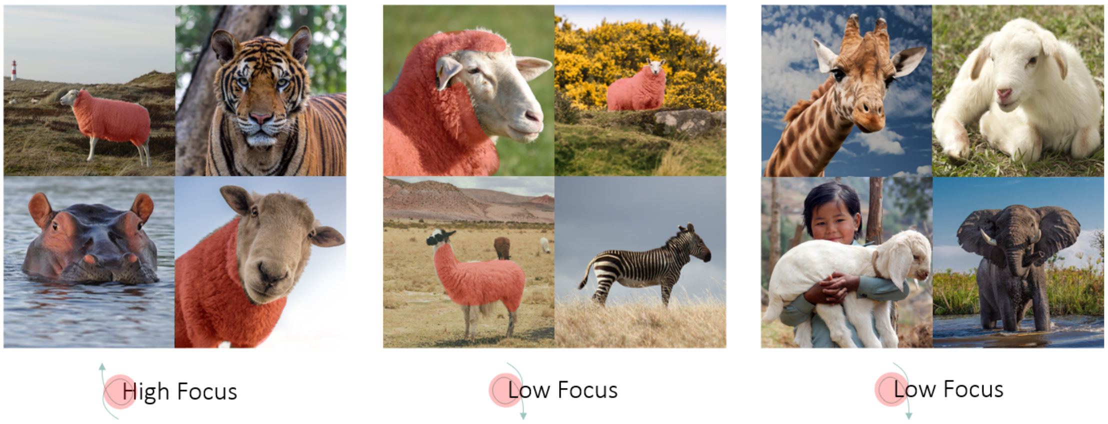


Figure 10.1: Possible explanations obtained with a model that has learned to differentiate the *sheep* class based on the presence of wool texture. The *target class* of these mosaics is the *sheep* class. Depending on whether or not the bias is in the target class images or if it is in the non-target class images or not, the *Focus* score will be higher or lower.

user. We introduced this approach in [8][1].

One could ask, how can the *Focus* help in this process? The main idea is straightforward: since mosaics induce in-distribution noise, confused attribution on the wrong regions may directly correspond to visual biases of the model. To better illustrate this scenario, let us return to the sheep example. Let us imagine we are analysing the mosaics shown in Figure 10.1, where the *target class* of those mosaics is the *sheep* class, and red areas correspond to the positive attribution towards the *target class*. If we compute the *Focus* score on those explanations, the first one will obtain a high *Focus* score since the bias (*i.e.*, the wool texture) is present in the sheep instances. The second mosaic will obtain a lower *Focus* since the bias is also in the non-target class image (*i.e.*, in the lama image). And the third mosaic will also obtain a lower *Focus* since the bias is neither present in the target class images (*i.e.*, lamb images) nor in the non-target class images.

Therefore, this inherent feature of mosaics and *Focus* can be used to automate the selection process of those instances that may contain potential biases. This process will depend on the specific use case we are interested in. For instance, if we want to analyze the biases between two specific classes, we might prefer to create 2×1 mosaics. Or, we might be interested in looking for data-driven biases by using images outside the original data distribution. Thus, we could build mosaics by mixing images from the training dataset and images from other datasets.

[1] Part of this chapter can also be found in that work.

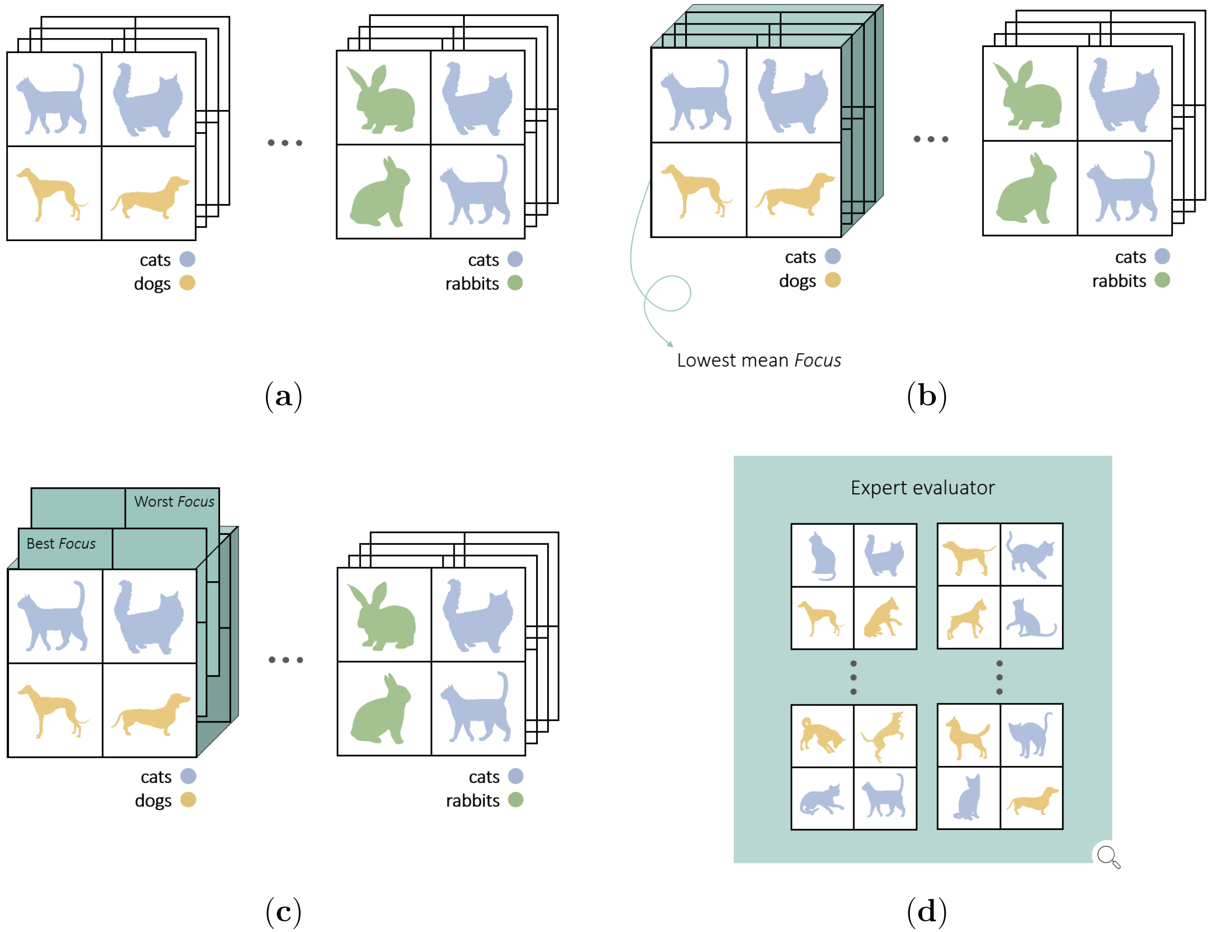


Figure 10.2: Steps of the proposed approach. (**a**) First, compute pair-wise *Focus*. (**b**) Pick those pairs with the lowest mean *Focus*. (**c**) From those pairs, get the mosaics with the highest and the lowest *Focus*. (**d**) Finally, present the mosaics containing potential biases to the human evaluator.

To illustrate this idea, we present a possible approach designed to find potential biases between pairs of classes in any dataset or model (see Figure 10.2).

1. First, we use mosaics with two classes to better detect biases between pairs of classes. Therefore, in the mosaics used for this experiment, samples different from the *target class* actually belong to the same class: $c(img_3) = c(img_4) \neq tc$.

2. We concentrate on the most relevant biases by finding the pairs of classes obtaining the lowest mean *Focus* in their joint mosaics.

3. Then, for each pair of classes, the approach extracts the mosaics with highest and lowest *Focus*.

4. And finally, those mosaics are presented to a human evaluator who must review the produced explanations. The evaluator's role is to interpret the rationale behind the explanations (both correct and incorrect) and the degree of generalization for the task. Based on that assessment, corrective measures can be implemented, as later discussed.

We follow this approach to identify potential biases in the widely known ImageNet dataset. We use the GradCAM method and the ResNet-18 architecture for this experiment, a particularly robust configuration according to our experiments presented in §7.2 (with this choice we also reduce the XAI BIAS). An example is shown in Figure 10.3, the top mosaic corresponds to a high *Focus* mosaic and the bottom one to a low *Focus* mosaic. We can see how the model is able to correctly attribute relevance to the *peacock* images on the upper mosaic, while, for the bottom mosaic, some of the relevance incorrectly falls on the head of the *common iguana*. The fact that most of the incorrect relevance in the *common iguana* falls in the subtympanic shield (*i.e.*, the characteristic circle in its jowl) seems to be related to its visual similarity with the ocellus of the *peacock* (*i.e.*, the circular spot in the feathers). Notice the iguana's subtympanic shield is hardly visible in the top mosaic.

After identifying biases and assessing their impact, one could try to mitigate their relevance for the model. For example, with more images of the target class without the characteristic pattern found in the outer class could be added to the training set (*e.g.*, *peacocks* images where the ocellus is not visible). Or, more images of the outer class where the characteristic pattern is present (*e.g.*, *common iguana* images where the subtympanic shield is visible) could be added. In either case, the dependency of the target class with respect to the bias would be reduced, increasing the robustness of the model.

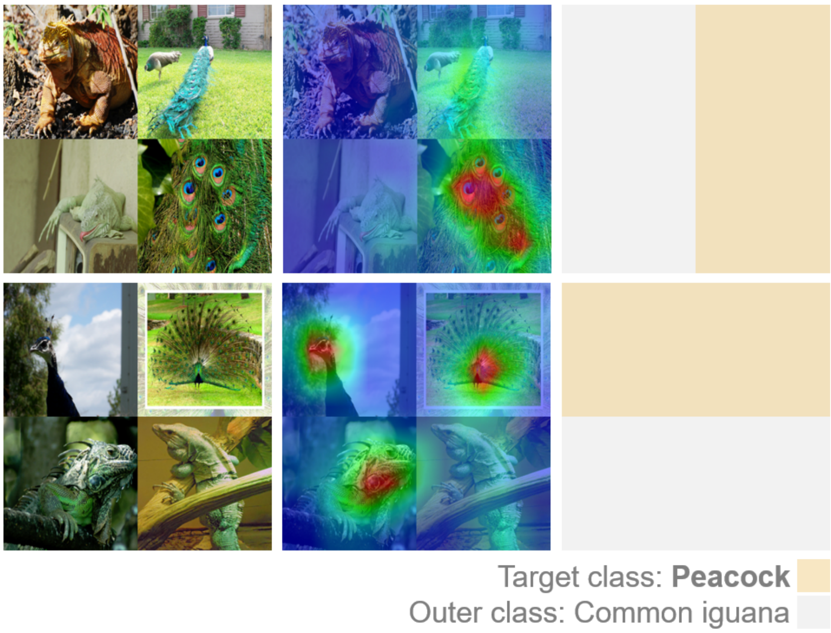


Figure 10.3: In the first column, two examples of mosaics are depicted. The second column shows the corresponding GradCAM explanations obtained with a ResNet-18 architecture trained on ImageNet. In this case, the target class is the *peacock* class, and the outer class is the *common iguana* class. The third column specifies the positions of the classes within the mosaic. The mosaic above obtains a high *Focus* score (0.818), and the one below a low *Focus* score analogous to a random explainer (0.494).

This experiment shows how the *Focus* score seems a promising tool for the selection of samples containing potential unwanted biases. However, in the model of the previous example (*i.e.*, trained on the ImageNet dataset), we had no control over the existing biases. Instead, we decide to build a model trained on a dataset in which we can control the level of unwanted correlations and therefore quantify the bias introduced using the *Focus*. This approach was introduced in [9]. Let us first train the biased model.

10.2 | BUILDING A BIASED MODEL

To train the biased model, we first created a biased dataset §10.2.1. Then, we train the model on that biased dataset §10.2.2. And finally, we perform some *sanity* checks to verify whether we managed to introduce an unwanted correlation into the model §10.2.3.

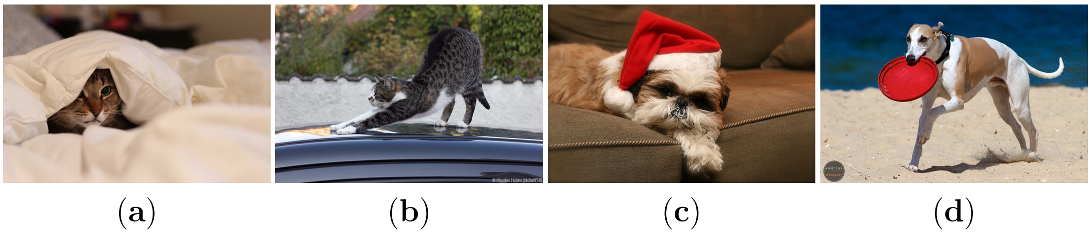

(a) (b) (c) (d)

Figure 10.4: Examples of indoor/outdoor images: (**a**) cat-indoor (**b**) cat-outdoor (**c**) dog-indoor (**d**) dog-outdoor.

10.2.1 | DATASET CREATION

The dataset creation is motivated by the need to have control over some of the dataset biases. To do so, we use the MetaShift [54] to induce a correlation that we can quantify and control. This work clusters the images according to metadata. An annotated graph is created where each node represents a class in a specific context (*e.g.*, *dog frisbee*). The distance between nodes represents the similarity between those contexts (*e.g.*, *dog frisbee* will be closer to *dog grass* than *dog books*). The more contexts are shared within a class, the closer the nodes will be. Using the construction proposed by [54], we create a dataset composed of two classes (cat and dog) with two subclasses (indoor and outdoor), see Figure 10.4 for details.

We built the dataset with images from two well-known datasets, both providing contextual information: the Common Objects in Context (COCO) dataset [55] and the Visual Genome dataset [50]. Tables 10.1 and 10.2 show the exact contexts used

Table 10.1: Contexts included in each category for the Visual Genome dataset. The first and second column corresponds to the cat-outdoor and cat-indoor category. And the third and fourth column to the dog-outdoor and dog-indoor categories respectively.

☀ 🐈	💡 🐈	☀ 🐕	💡 🐕
car, fence, grass, roof, bench, bird, house	speaker, computer, screen, laptop, computer mouse, keyboard, monitor, desk, sheet, bed, blanket, remote control, comforter, pillow, couch, books, book, television, bookshelf, blinds, sink, bottle, faucet, towel, counter, curtain, toilet, pot, carpet, toy, floor, plate, rug, food, table, box, paper, suitcase, bag, container, vase, shelf, bowl, picture, papers, lamp, cup, sofa	house, grass, horse, fence, cow, sheep, dirt, car, motorcycle, truck, helmet, snow, flag, boat, rope, trees, frisbee, bike, bicycle, sand, surfboard, water, fire hydrant, pole, skateboard, bench, trash can	screen, shelf, desk, picture, laptop, remote control, blanket, bed, sheet, lamp, books, pillow, curtain, container, table, cup, plate, food, box, rug, floor, cabinet, towel, bowl, television, carpet, sofa

Table 10.2: Contexts included in each category for the COCO dataset. The first column corresponds to the outdoor contexts and the second to the indoor ones.

☼	💡
bicycle, car, motorcycle, airplane, bus, train, truck, boat, traffic light, fire hydrant, stop sign, parking meter, bench, frisbee, skis, snowboard, sports ball, kite, baseball bat, baseball glove, skateboard, surfboard, tennis racket	bottle, wine glass, cup, fork, knife, spoon, bowl, chair, couch, potted plant, bed, dining table, toilet, tv, laptop, mouse, remote, keyboard, cell phone, microwave, oven, toaster, sink, refrigerator, book, clock, vase, scissors, teddy bear, hair drier, toothbrush

for the construction of the indoor and outdoor subclasses for both datasets, the Visual Genome dataset and the COCO dataset respectively.

10.2.2 | MODEL

Next, we train a model using only samples from cats-indoor and dogs-outdoor. In this way, we expect to introduce an unwanted correlation, which could, in fact, occur in a real scenario: dog-outdoor images are more likely than cat-outdoor images.

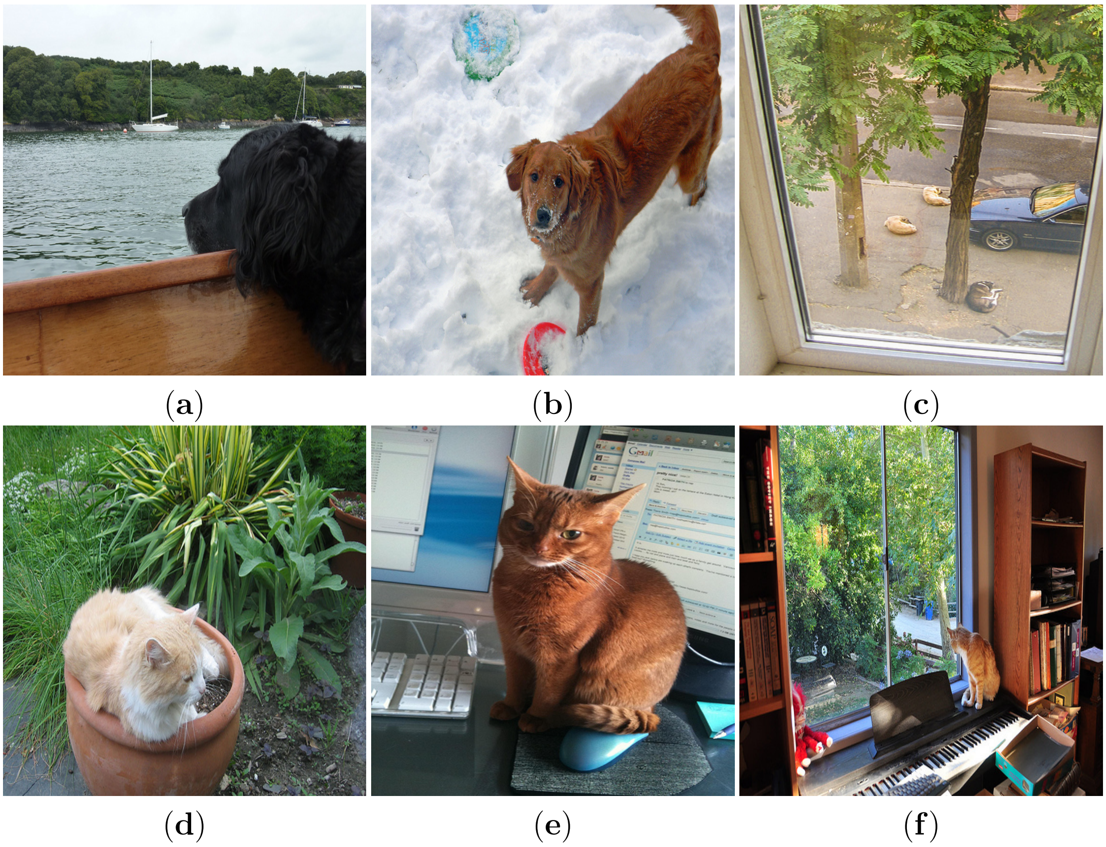
For training the model, we use a total of 1,060 images per class (cats-indoor versus dogs-outdoor). Where 960 images per class were used for training and 100 for validation. We use the ResNet-18 [36] architecture, the AMSGrad [80] to optimize weights and we perform data augmentation during training: random rotation ([-30, 30] degrees), random crop and random horizontal flip with a chance of 50%. We reach a mean per class accuracy on the validation set of 87%, corresponding to the model with the minimum validation loss. From here, we will call this model: the *biased model*.

We also train a model which avoids those unwanted correlations for comparison purposes. We use the same training size (1,060 images per class), but in this case, both cats and dogs will be equally present in both contexts, 50% outdoors and 50% indoors. We reach a mean per class accuracy on the validation set of 60.5%, using the model with the minimum validation loss. Notice that the performance obtained is much lower, indicating that the induced context was a successful shortcut to the model. Without this added bias, the high variability (different breeds) and the low quality (mislabeled samples or partially occluded animals) of the dataset limits the model's performance, which fails to learn to distinguish the two classes robustly. From now on, we will refer to this second model as the *non-biased model*.

10.2.3 | SANITY CHECKS

To empirically prove that the previous model, trained for the cats (indoor) and dogs (outdoor) classification task, is indeed biased (*i.e.*, it has managed to learn the context instead of cat and dog characteristics patterns), we perform the following experiment. Starting from the hypothesis that images predicted with low probability or that are predicted as the opposite class (in the case of a binary classification problem) are likely to be those that have patterns of the opposite class, we selected the three dog images with the lowest prediction and the three worst cat image predictions, see Figure 10.5.

Before continuing with the hypothesis evaluation, it is worth mentioning how samples predicted with the least certainty significantly differ between cats and dogs. While for dogs, the lowest probability corresponds to 56.58% and the third lowest to 82.11% (both of which account for a correct classification in a binary prob-


(**a**) (**b**) (**c**)

(**d**) (**e**) (**f**)

Figure 10.5: Examples of the worst predictions of the validation images set. Worst dog predictions: (**a**) dog: 0.5658, (**b**) dog: 0.7948 and (**c**) dog: 0.8211. Worst cat predictions: (**d**) cat: 0.0038, (**e**) cat: 0.4627 and (**f**) cat: 0.4729.

lem), for cats, these probabilities drop to 0.38% the lowest, and the third lowest to 47.29%. This is likely due to the prevalence of dog-related biases (*i.e.*, outdoors patterns) in the validation set with respect to cat-related biases. As shown in Figure 10.5 (and mentioned before), the worst predicted cat sample seems to be a labelling mistake (labelled as indoor when it seems to be outdoor). We do not correct this mistake for the sake of methodological consistency. These results show a higher performance when classifying *dogs-outdoor* than *cats-indoor*, suggesting that the model has learnt to focus more on outdoor than indoor patterns. This may be due to the fact that outdoor patterns are less variant and more frequent, being a perfect visual pattern to discriminate between both classes.

Following the previous hypothesis: images predicted with a low probability are likely to contain a pattern of the opposite class. We build a set of 2×1 mosaics by combining those pairs of images (cats versus dogs) to apply a feature attribution method. As we have already introduced in the previous chapter, using feature attribution methods on top of the mosaics enhances the detection of biases.

For this experiment, we use the GradCAM attribution method. According to the *Focus* results presented in Chapter 7, GradCAM is the method obtaining better performance and therefore minimizing the noise introduced into the pipeline. Results for both *target classes* are shown in Figure 10.6 and Figure 10.7. In all mosaics with the *target class* being the dog (see Figure 10.6), the GradCAM attribution focuses on areas where trees, leaves, or plants are present. Regardless of whether these patterns appear in the cats' or dog squares. Based on this, we hypothesize that the model has learnt to detect vegetation instead of discriminating between cats and dogs. Indeed, it seems reasonable to think that most dogs in an outdoor context will be on meadows, fields or mountains (with a prominent presence of vegetation), while indoor cats will lack such a pattern. This situation would have made it easier for the model to distinguish between dogs and cats by only learning the green context instead of learning the characteristic patterns of these two mammals. Similarly, the attribution in Figure 10.7, with the cat being the *target class*, falls on the wood or the brown areas (*e.g.*, first column of Figure 10.7). Although to a lesser extent than the vegetation, this pattern seems to be learnt by the model as a characteristic pattern of the cat class.

In order to corroborate that the model has learnt to identify *vegetation* as a characteristic pattern of the *dog* class and *wood* as characteristic of the *cat* class, we perform another sanity check. We fed the model with the hand-selected images shown in Figure 10.8, obtained from external sources. Image (**a**) is an image of only grass, which is predicted as a dog with a probability of 99.98%. On the contrary, Image (**b**) is a wood image which is predicted as a cat with a probability of 96.30%. In the case of Image (**c**), both patterns are present, although the green pattern is more prominent. This image is predicted as a dog with a probability of

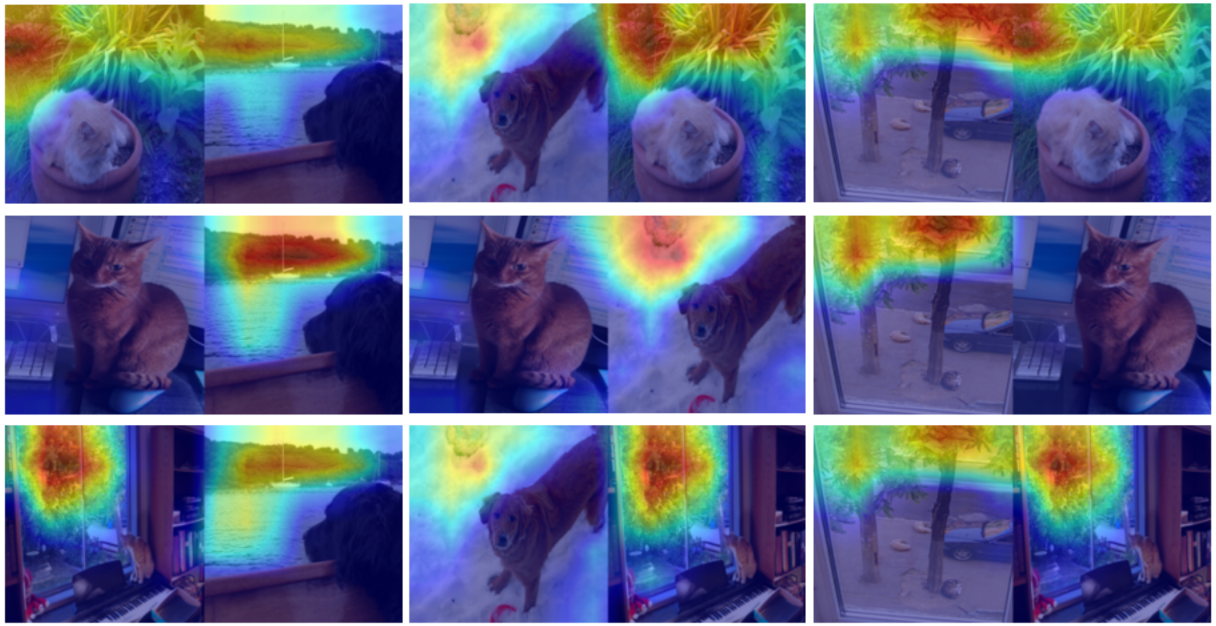


Figure 10.6: Feature attribution maps obtained by GradCAM on the *bias model* (the *dog* being the target class).

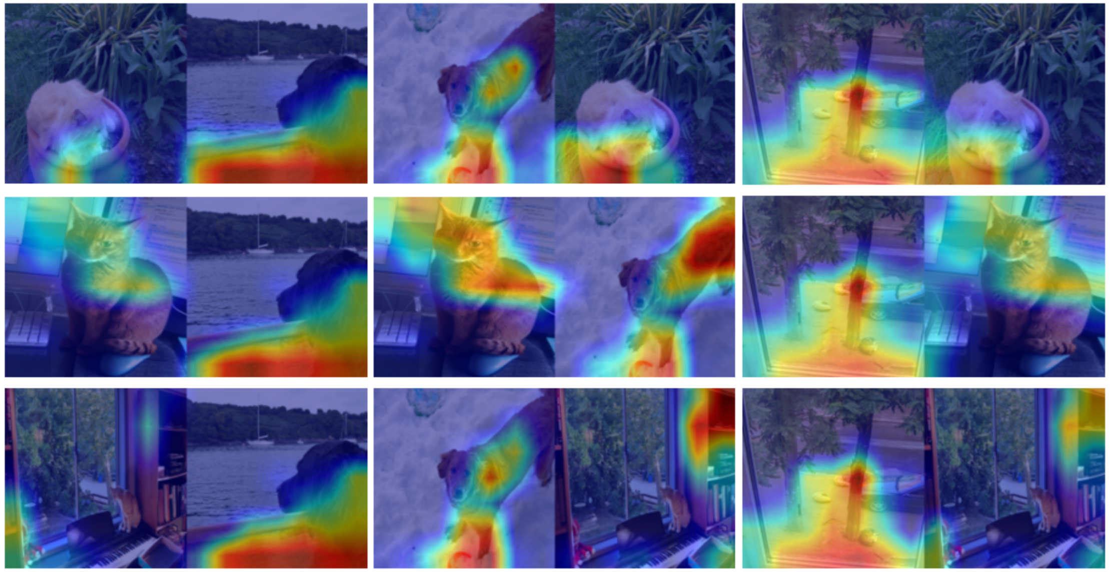


Figure 10.7: Feature attribution maps obtained by GradCAM on the *bias model* (the *cat* being the target class).

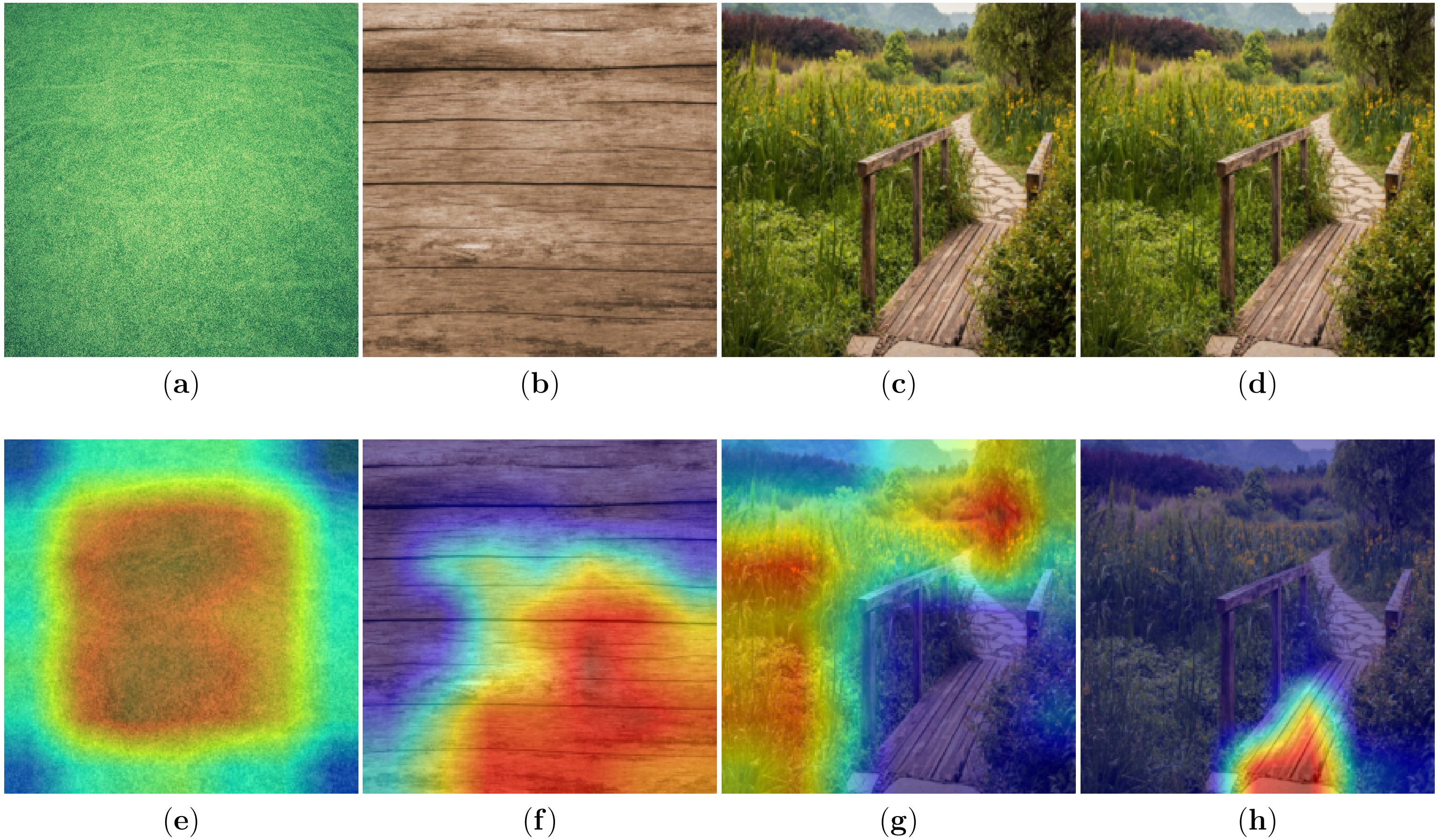


Figure 10.8: First row: hand-selected samples predicted by the *biased model* as (**a**) dog: 0.9998 (**b**) cat: 0.9630 (**c**) and (**d**) dog: 0.9944. Second row: feature attribution maps obtained by GradCAM for the images of the first row being the *target class*: (**e**) the dog class, (**f**) the cat class, (**g**) the dog class and (**h**) the cat class.

99.44%. Notice how the attribution, being the *target class* the dog class (see Image (**g**)), falls on the green part around the path. However, when we ask for the attribution of the cat class (see Image (**h**)), the relevance focuses on the wooden bridge.

These results validate our hypothesis: vegetation is the primary pattern learnt by the model as characteristic of the *dog* class, and the wood pattern is learnt as characteristic of the *cat* class. At this point, we can confirm that the model is clearly skewed, it has learnt to differentiate the two classes mainly by context and not by the animal, and furthermore, we are aware of the principal patterns enabling such distinction.

10.3 | FOCUS ON A BIASED MODEL

This section evaluates the *Focus* behaviour when applied to the *biased model* and the *non-biased* model. As previously introduced, we use the GradCAM method. And for the mosaics, we build four sets of 2×1 mosaics, following all possible combinations. Each set contains the same amount of mosaics (10,000):

1. **cat-indoor versus dog-outdoor**: Combines 100 cat-indoor images and 100 dog-outdoor images. Note that this set follows the same distribution used for training the *biased model.*

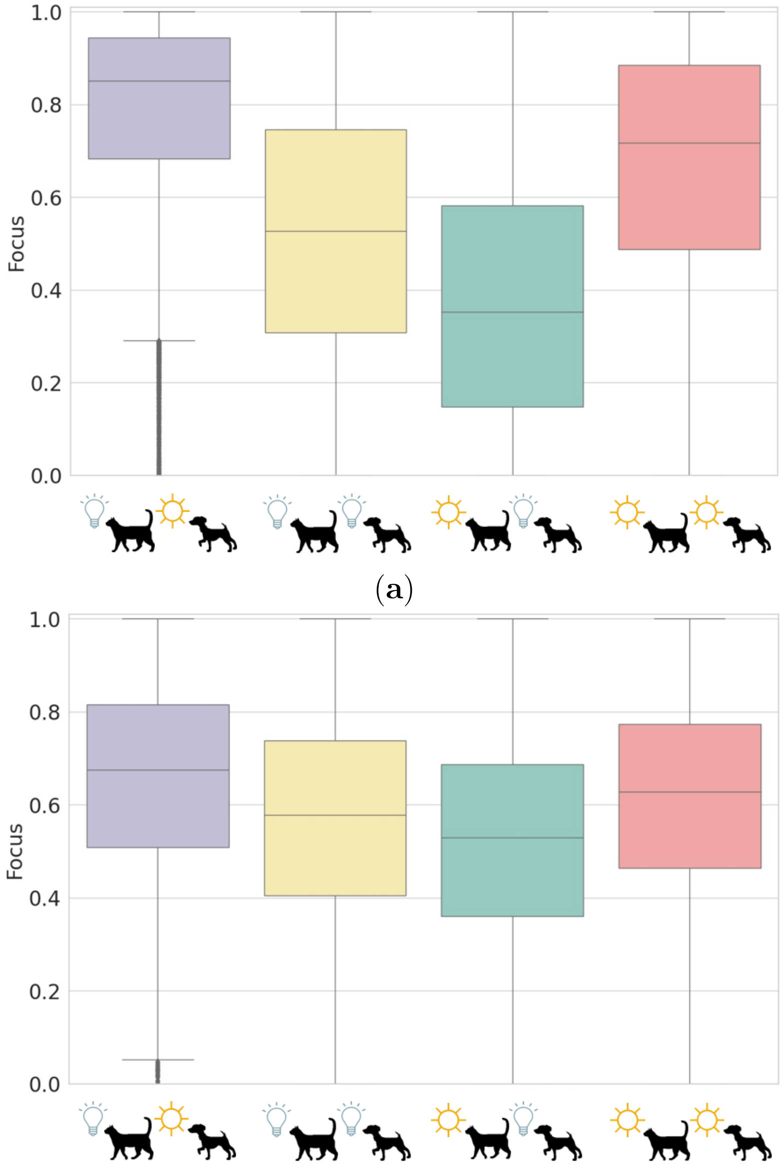
2. **cat-indoor versus dog-indoor**: Combines 100 cat-indoor images and 100 dog-indoor images.

3. **cat-outdoor versus dog-indoor**: Combines 100 cat-outdoor images and 100 dog-indoor images. Note this set corresponds to a distribution complementary to the one used for training the *biased model.*

4. **cat-outdoor versus dog-outdoor**: Combines 100 cat-outdoor images and 100 dog-outdoor images.

Note that none of these sets corresponds to the distribution used for training the *non-biased model* in which samples of all sets are used (cats and dogs equally sampled from indoor and outdoor contexts). At this point, we can now compute the *Focus* obtained by each of the two models on each of the four mosaic sets. The resulting *Focus* distributions (including the 10,000 samples per set) are shown in Figure 10.9.

In the experiments with the *biased model*, the highest *Focus* is expected to be obtained with Set 1 since the images within this set follow the same distribution in which the model has been trained. On the other hand, the *Focus* obtained with Set 3 should be the lowest since the images correspond to the completely inverse



(**a**)

(**b**)

Figure 10.9: Each box plot shows the *Focus* distribution for a different validation set (evaluating 10,000 mosaics per set). The purple box plots correspond to the cat-indoor and dog-outdoor set (Set 1). The yellow box plots correspond to the cat-indoor and dog-indoor sets (Set 2). The green box plots to the cat-outdoor and dog-indoor set (Set 3). And the red box plots to the cat-outdoor and dog-outdoor set (Set 4). (**a**) *Focus* distributions obtained by GradCAM on the *biased model* (**b**) *Focus* distributions obtained by GradCAM on the *non-biased model.*

distribution. In this case, the mean *Focus* is expected to be between 0 and 0.5 since the learnt biases may be found on the non *target class* squares.

In the experiments with the *non-biased model*, we expect the *Focus* distributions to be similar to one another. The training distribution of this model avoids biases regarding indoor and outdoor, which should prevent the model from focusing on these properties. Thus, the four sets become analogous if the context is not a factor.

Results follow our hypothesis as seen in Figure 10.9. The context (indoor-outdoor) plays a significant role in the *biased model* and has a much weaker impact on the results of the *non-biased model*. For the *biased model*, a mean *Focus* greater than 0.8 is obtained when using the same context as in training (Set 1, see first box plot in Figure 10.9 (**a**)). However, when the complementary distribution is used, Set 3, the mean *Focus* falls below 0.4. As hypothesized before, this low *Focus* is most likely due to the model finding patterns in the image of the opposite *target class*. Finally, the two sets having at least one correct context (Set 2 and Set 4) obtain a mean *Focus* in between the two mentioned above (see the second and the fourth box plot in Figure 10.9 (**a**)).

We hypothesize that a significant amount of label noise is found (particularly in the cat outdoor class, incorrectly labelling indoor cat images as outdoor samples). This would explain the fact that outdoor cats and dogs (red box plot of Figure 10.9 (**a**)) obtain a higher *Focus* than indoor cats and dogs (yellow box plot of Figure 10.9 (**a**)) as well as why the inverse distributed set (green box plot of Figure 10.9 (**a**), mean *Focus* of 0.3532) is not complementary of the equally distributed set (purple box plot of Figure 10.9 (**a**), mean *Focus* of 0.8507).

In contrast, the *Focus* distributions obtained with the *non-biased* model have a mean *Focus* close to each other. The mean *Focus* obtained with Set 1 is still the highest, as shown in Figure 10.9 (**b**), and the mean *Focus* obtained with Set 3 is slightly the lowest. This is likely to be caused by label noise induced by the natural predominance of cats to be indoors and dogs to be outdoors.

10.4 | SUMMARY OF THIS CHAPTER

In this chapter, we first introduce the sources of harm that lead to biases in the data and models. Focusing on the most common in the CV area: *representation* and *evaluation* biases.

As previously introduced, when using explainability to detect biases in data and models other sources of biases can arise (*e.g.*, HUMAN BIASES). That is why, in this section, we explain the importance of providing tools that help experts use explainability to find biases, minimizing both the XAI BIASES (*i.e.*, choosing the XAI methods providing more faithful explanations) and HUMAN BIASES (*i.e.*, semi-automating the detection task). To do so, we illustrate how the *Focus* together with mosaics can be a powerful tool for automating the bias detection process. Then, we propose an approach for finding biases in models and datasets by utilizing the *Focus* and mosaics and we apply it to the widely known ImageNet dataset.

To better analyse the *Focus* behaviour when applied to a biased model, we train a biased model to which we induce a controlled correlation: we only use cats-indoor and dogs-outdoor. In this way, the model is forced to learn a bias, in this case, the context. Then we perform a set of sanity checks to verify that this model is indeed biased. Following that, an explainability method (GradCAM) is used on top of mosaics. The nature of mosaics allows us to easily identify the biases found within the model: the model learnt vegetation patterns as characteristic of the *dog* class, while brown and wood patterns are learnt as characteristic of the *cat* class. We use this *biased model* to analyze the behaviour of the *Focus* when applied to the biased setting. Additionally, for comparison purposes, we also train a *non-biased model* as a baseline. To perform this experiment, we use four mosaic sets: cat-indoor vs dog-outdoor (Set 1), cat-indoor vs dog-indoor (Set 2), cat-outdoor vs dog-indoor (Set 3) and cat-outdoor vs dog-outdoor (Set 4). Our findings show how the presence of a shared bias is clearly reflected in the *Focus* distribution. The *Focus* decreases when the context learnt by the model is present in both classes within the mosaics. This shows again the potential of the *Focus*, together with the mosaic structure, for detecting potential biases in datasets and models.

11 | MOSAICS FOR CONTEXT BIASES

> Taken out of context I must seem so strange.
>
> *Angela Maria DiFranco*

In the previous chapter, we used mosaics in conjunction with *Focus* to detect DATA and MODEL BIASES. In this chapter, we propose to exploit mosaics for the same purpose but without using explainability. In this way, we avoid both XAI BIASES and HUMAN BIASES. The main idea of this approach is to use mosaics built by combining images of the original data distribution with images of potential biases and then explore the models by directly observing their outputs. This contribution was introduced in [6][1].

11.1 | EXPERIMENTAL DESIGN

First, the new synthetic datasets created for this work are presented (§11.1.1), then the training configurations used (§11.1.2) and finally, the generalization capabilities of all trained models are evaluated (§11.1.3).

11.1.1 | DATASET

An image diffusion model was used to create the datasets employed in these experiments. With diffusion models, one can specify what to generate and guide it to produce realistic images. The model used was a text-to-image diffusion model [82]: from a text prompt, the model generates realistically looking images which are, at the same time, faithful to the text. We generate three different datasets (see Figure 11.1), each one composed of four classes: *bench*, *fire hydrant*, *plane*, and *mug*. All of these are publicly available, and these are the details for its generation:

1. Context (C): This dataset is composed of images corresponding to the four objects in a typical context, according to the model's representation. The

[1] Part of the content of this chapter can be found in that work.

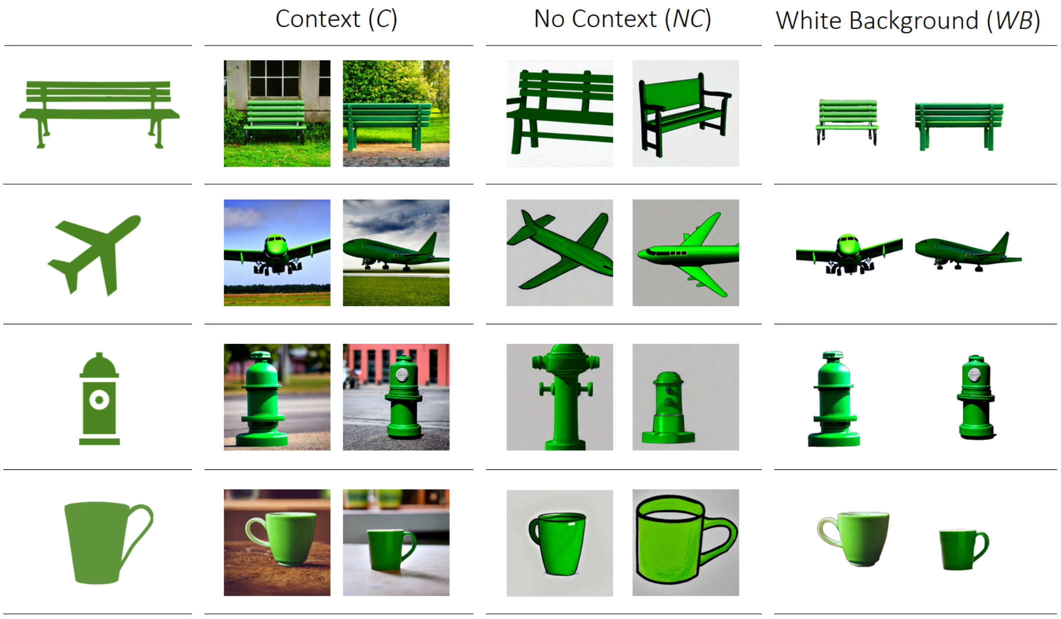


Figure 11.1: Sample instances by class and dataset. Each dataset is shown in a different column (from left to right): context (C), no context (NC) and white background (WB) dataset. Class examples are separated by row (from top to bottom): bench, plane, fire hydrant and mug.

exact prompt used to generate these images was: A GREEN <u>CLASS</u> ON THE FOREGROUND. TYPICAL BACKGROUND. For each class, the word CLASS is replaced by the object: bench, fire hydrant, plane or mug.

2. No Context (NC): This dataset contains images of the four same objects but without a background. To that end, we slightly changed the prompt and asked for a sketch of the object with uniform background. The exact prompt used to generate these images was: NO BACKGROUND. SIMPLE SKETCH OF A GREEN <u>CLASS</u>.

3. White Background (WB): For creating this dataset, we manually removed the background of the C dataset images using a tool designed and provided by Adobe[2]. Therefore, this dataset is composed of the same images as the C dataset, but setting the background to white.

Notice that for each class of each dataset, 150 images were created. And in order to prevent the model from learning to differentiate these classes by their

[2] https://www.adobe.com/express/

recurring colours (*e.g.*, most fire hydrants are red) or by texture (*e.g.*, benches are often made of knotted wood), we set the colour of the four objects to green.

11.1.2 | TRAINING SETUP

We train six different models using the three datasets introduced before. Due to the simplicity of the datasets, we use the AlexNet [51] architecture, a shallow architecture that can fit our data. Each dataset is used to train two models: one from scratch and one pre-trained on ImageNet [85] and then fine-tuned for it. For the pre-trained models, we use the AlexNet model available in the *torchvision.models* subpackage[3]. We use a total of 100 images per class for training, 25 for validation and 25 for the test. To avoid confusion, we will refer to them as follows:

1. **model-*C***: model trained from scratch on the *C* dataset.
2. **model-*NC***: model trained from scratch on the *NC* dataset.
3. **model-*WB***: model trained from scratch on the *WB* dataset.
4. **pt-*C***: model pre-trained on ImageNet and fine-tuned on the *C* dataset.
5. **pt-*NC***: model pre-trained on ImageNet and fine-tuned on the *NC* dataset.
6. **pt-*WB***: model pre-trained on ImageNet and fine-tuned on the *WB* dataset.

11.1.3 | CROSS EVALUATION

Potential biases that may appear in the previously trained models actually originated in the diffusion model, then recreated in the dataset and finally learnt by the models. If we evaluate these models in a random split of the same dataset in which it has been trained, we will probably obtain a high performance even though these models have not learned the features of the four objects. However, if we test the models in a partition of the other two datasets (both having the same four classes), we will be able to evaluate the model's generalization capabilities.

To do so, we cross-evaluate all six models with all three datasets. The accuracies and cross-accuracies obtained with the different models are illustrated in Figure 11.2. Each histogram (*i.e.*, group of three bars) corresponds to one of the six models. Each bar corresponds to the accuracy obtained by using a different test set. And each set is shown with a different colour: yellow corresponds to the *NC* set, grey to the *C* and, green to the *WB* set.

First of all, models trained from a random state (first three histograms of Figure 11.2) perform more poorly on all test settings than pre-trained models

[3] https://download.pytorch.org/models/alexnet-owt-4df8aa71.pth

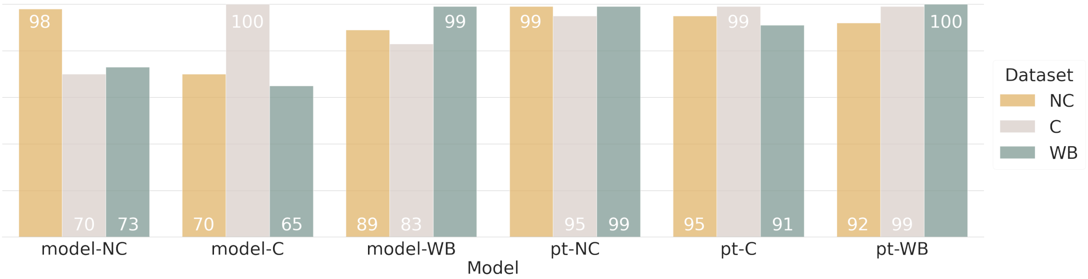


Figure 11.2: Accuracies obtained with the six models (model-NC, model-C, model-WB, pt-NC, pt-C and pt-WB) on the three test sets (NC, C, WB). Each set is represented in a different colour and the results for each model are grouped in the form of a histogram (group of three bars).

(three last histograms of Figure 11.2). The best accuracies of those non-pre-trained models (model-NC, model-C and model-WB) are achieved when using the same distribution as for training (*i.e.*, 98%, 100% and 99% respectively). However, those models struggle to correctly distinguish the classes when using cross-tests.

On the contrary, pre-trained models manage to generalize much better than the models trained from scratch: the difference between the bars of histograms fourth, fifth and sixth are less prominent than the differences seen in the first three histograms. As expected, using pre-trained models prevents the model from learning patterns, or better-called shortcuts, that are present in the small training dataset, but they are not the patterns expected to be learned by the model. Also note that the plane, the mug and the bench are also classes of the ImageNet dataset (check here), which means that the pre-trained models knew the visual features needed to identify the different classes before the fine-tuning process.

An interesting finding in this first analysis is the relevant role played by the context. The only model that performs consistently well on all distribution shifts for non-pre-trained models is the one trained with WB. Obtaining an accuracy of 89% when using the NC and 83% when using the C. This lower performance when using the C set, despite being the same exact objects but with white background, is most likely due to the large distribution shift that the presence of context (*i.e.*, patterns surrounding the objects) supposes for a model not trained with a background. The same is observable for the models trained with NC where the worst performance is obtained with the C set (first and fourth histogram of Figure 11.2).

On the other hand, the model trained from scratch with context images is the one generalizing the worst, obtaining a performance of 70% when tested with NC and a 65% accuracy when using the WB. The C model may have learnt contextual biases. And thus, the model does not maintain the performance when those context features are not present (*e.g.*, within the NC and WB sets).

11.1.4 | CONTEXT BIASES AND CONTEXTUALIZED MOSAICS

As previously introduced, in this work, we focus on studying *context* biases: if each context is specific to each class and it is not found in the rest of the classes, the model will learn those contexts as *shortcuts*. As we empirically stated in the previous section, these learned shortcuts lead to biased models that are not able to generalize correctly.

To formalize the model predictive behaviour as a desirable causal model [30], we build a Directed Acyclic Graph (DAG) to represent the problem. Each node in this DAG represents the object present in the images (O), the context (C) and the predicted class (Y). The desirable setting representing the relationship between these nodes would be the one shown in Figure 11.3 (**a**). However, the graph learned by the model is probably more similar to the one shown in Figure 11.3 (**b**). To assess the relationship between C and Y, that is, the relationship between the context and the predicted class by the model (*e.g.*, the relation between the vegetation context for the bench class), we perform an intervention fixing C=c, with four possible alternatives (*i.e.*, one context per object): $do(C = c_1)$, $do(C = c_2)$, $do(C = c_3)$ and $do(C = c_4)$. The new graph after the intervention will be the one shown in Figure 11.3 (**c**). To perform the intervention, we construct what we call the *contextualized* mosaics.

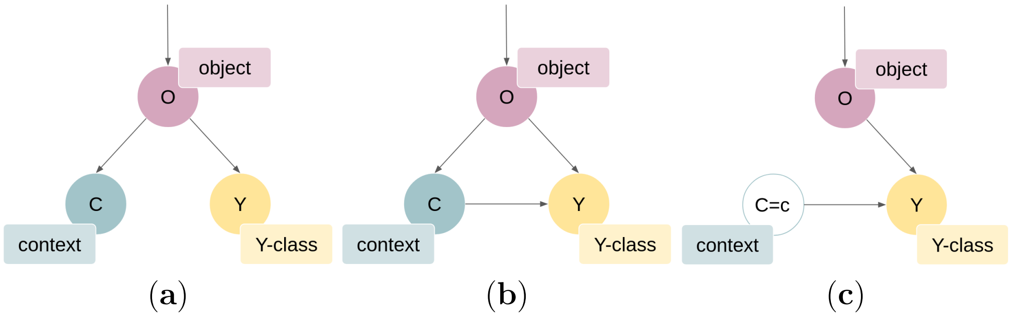
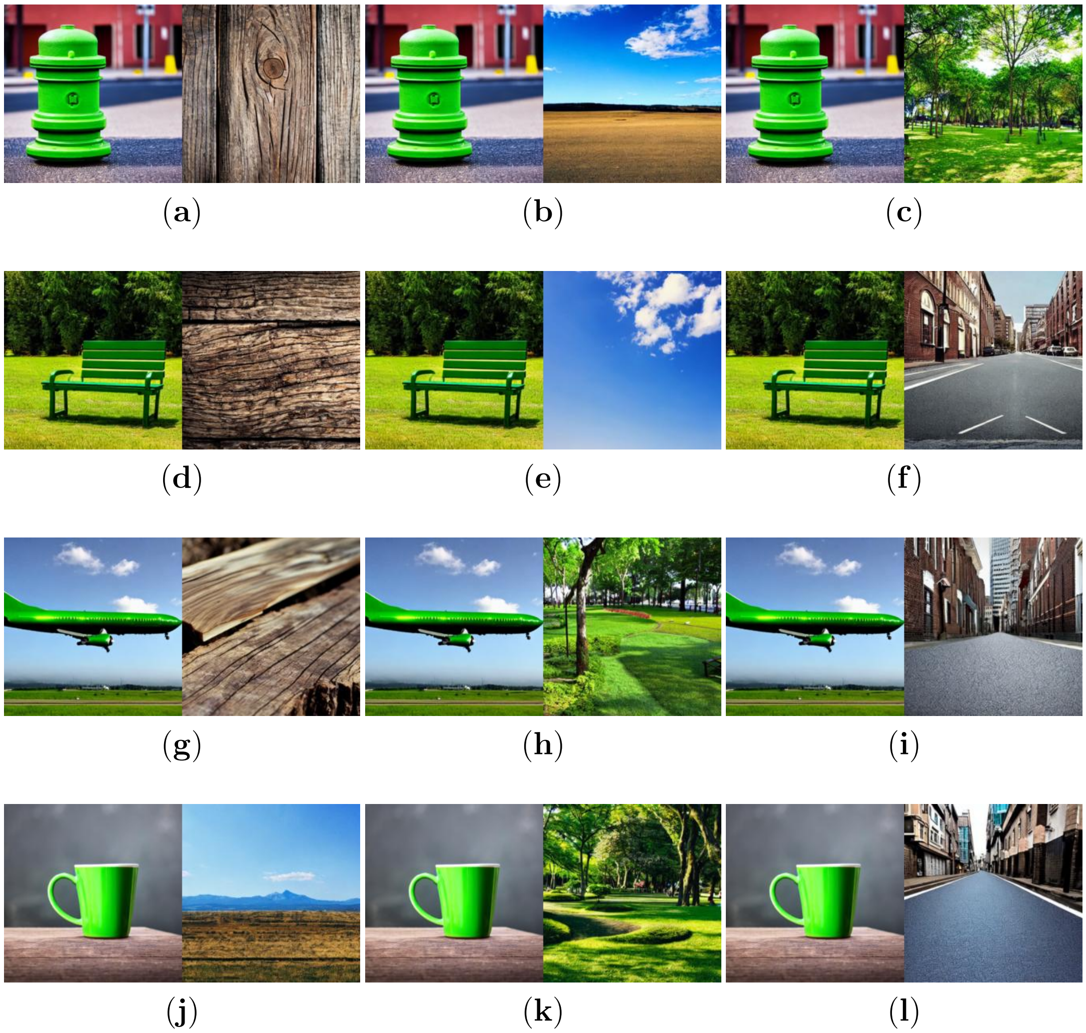


Figure 11.3: (**a**) Desired causal model representing the relationships between the object within the images (O), the context (C) and the predicted class (Y) (**b**) Actual graph learned by the model. (**c**) The modified graph after the intervention (fixing C).

For the *contextualized* mosaics, apart from the set of images $\mathbb{I} = \{img_1, img_2, \ldots img_N\}$ and the set of classes $\mathbb{C} = \{c_1, c_2, \ldots c_K\}$, we will also have a set of contexts $\mathbb{X} = \{ctx_1, ctx_2, \ldots ctx_K\}$, each context composed of a set of context images. Note, that there will be the same number of contexts as classes, that is, one potential bias context per class. For the mosaic construction, we fix the mosaic size to $J = 2$ (mosaics of size 1×2), where each mosaic will be made up of an image and a context $m = \{img, ctx\}$ where $c(img) \neq c(ctx)$. In other words, the



(**a**) (**b**) (**c**)

(**d**) (**e**) (**f**)

(**g**) (**h**) (**i**)

(**j**) (**k**) (**l**)

Figure 11.4: *Contextualized* mosaic examples of size 1×2 composed of object images with context images. Each row shows mosaics built from the same object image with different contexts. The first row corresponds to a fire hydrant image combined with a (**a**) wood context, (**b**) sky context and (**c**) park context. The second row corresponds to mosaics built with a bench image along with a (**d**) wood context, (**e**) sky context and (**f**) road context. Third-row mosaics are made up of a plane image with a (**g**) wood context, (**h**) park context and (**i**) road context. In the last row, mosaics of a mug image and a (**j**) sky context, (**k**) park context and (**l**) road context are shown.

class assigned to the context has to be different from the class of the image (*e.g.*, a mug will be combined with a sky, the sky being assigned the plane class since it is a typical context of the plane and not of the mug). Examples of *contextualized* mosaics are shown in Figure 11.4.

Let us delve into how we build these *contextualized* mosaics. The mosaic design is based on the assumption that target biases are known beforehand. This is a realistic case, as the domain expert should have prior knowledge of the possible biases that could exist. In our experiment, we can replace expert knowledge by analyzing the text prompt used during the C dataset generation, the text prompt included the sentence: TYPICAL BACKGROUND. From here, we observe the typical contexts where the four objects (*i.e.*, bench, plane, fire hydrant and mug) are usually found according to the generative model used. And we use the same diffusion model to obtain the context images with which we generated the training datasets. The selected context per object and the prompts used to generate those images are the following:

- A park for the *bench* class: A PARK WITH VEGETATION.
- Sky for the *plane* class: A CLEAR BLUE SKY.
- A road for the *fire hydrant* class: A REALISTIC TARRED ROAD IN A CITY.
- A piece of wood for the *mug* class: A PIECE OF WOOD.

Once these context images have been generated (see Figure 11.5), we build the mosaics by combining the original images of the different objects from the test set with the different contexts within a 1×2 grid. For each of the 25 object images, we combined them with five different samples obtained for each of the three contexts not belonging to that class. That is, for an image of a plane, we will combine it with five park images, five road images and five wood images. This results in a total of 1,500 mosaics. Examples of these mosaics are shown in Figure 11.4.

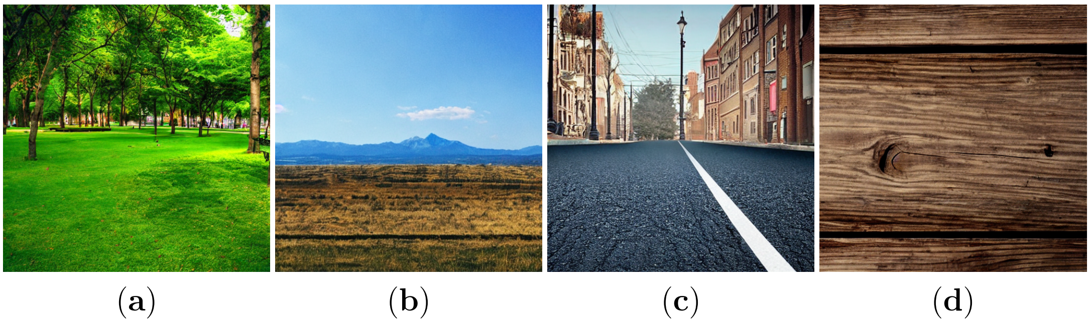


Figure 11.5: Examples of context samples generated for each class. (**a**) A park for the *bench* class (A PARK WITH VEGETATION.) (**b**) Sky image for the *plane* class (A CLEAR BLUE SKY.) (**c**) A road for the *fire hydrant* class (A REALISTIC TARRED ROAD IN A CITY.) (**d**) And a piece of wood for the *mug* class (A PIECE OF WOOD.)

11.2 | RESULTS

Let us analyse the results obtained using the *contextualized* mosaics to assess the relevance of the context in the predictions of model-C, the one that may have learnt context biases. When building the mosaics combining the original images with the context images, we induce a source of confusion for the model. We assess the impact of this *noise* (*i.e.*, context noise) by comparing the model's output of the class object image along with the model's output produced by the same image when composed in a mosaic with a context image. Results of this analysis are shown in Figure 11.6. For each possible combination of <class, context>, as long as $c(img) \neq c(ctx)$, we show a 1D and 2D histogram. The 2D histogram colour intensity represents frequency. The colour code used is green for bench images or mosaics with a park context, orange for the mug images or mosaics with a wood context, blue for the plane images or mosaics with a sky context and grey for the fire hydrant images or mosaics with road context. We observe the change in model probabilities induced by attaching a given context to an image of a given class. The larger this change is, the stronger the bias. To obtain a better understanding of Figure 11.6, we represent, in Figure 11.7, examples of images and mosaics utilized to obtain the first row results in Figure 11.6 (*i.e.*, using the sky context). The rectangle colors in Figure 11.7 were also aligned to the colors used in Figure 11.6. Let us now analyze the results in Figure 11.6 by context (*i.e.*, by row).

Sky. According to these results, the sky seems to be a relevant feature learned by the model. When the bench images are combined with the sky context, the sky confuses the model, obtaining higher logits for the *plane* class than for the *bench* class in 44 mosaics out of 125 mosaics (*i.e.*, 35.2%). In the case of the *fire hydrant*, the sky context also increases the logits of the *plane* class. However, the *fire hydrant* image features seem to be more relevant for the model, maintaining in 107 mosaics the highest logit values for the *fire hydrant* class. When combining the mug with the sky, the model is completely confused, being the logits for the *plane* class higher than the *mug* ones in more than half of the mosaics (64%).

Park. The vegetation also appears to be an important feature for the *bench* class, even to a greater extent than the sky context for the *plane* class. Although the lowest impact is for the mosaics with *fire hydrant* images, the park context still has a huge influence being the *bench* logits higher than the *fire hydrant* logits in 50 mosaics. The park context also impacts the mug results, managing to drastically shift the predictions towards the *mug* class in 115 mosaics (*i.e.*, 92%). In the plane case, 58 mosaics out of 125 obtain higher logits for the *bench* class (*i.e.*, 46.4%).

Road. This context does not seem to be as decisive as the two previous contexts (*i.e.*, sky and park context). Combining the *road* context with the three classes (*i.e.*, bench, mug or plane) has a lesser impact on the prediction: less than 12% of the mosaics are predicted as *fire hydrants*.

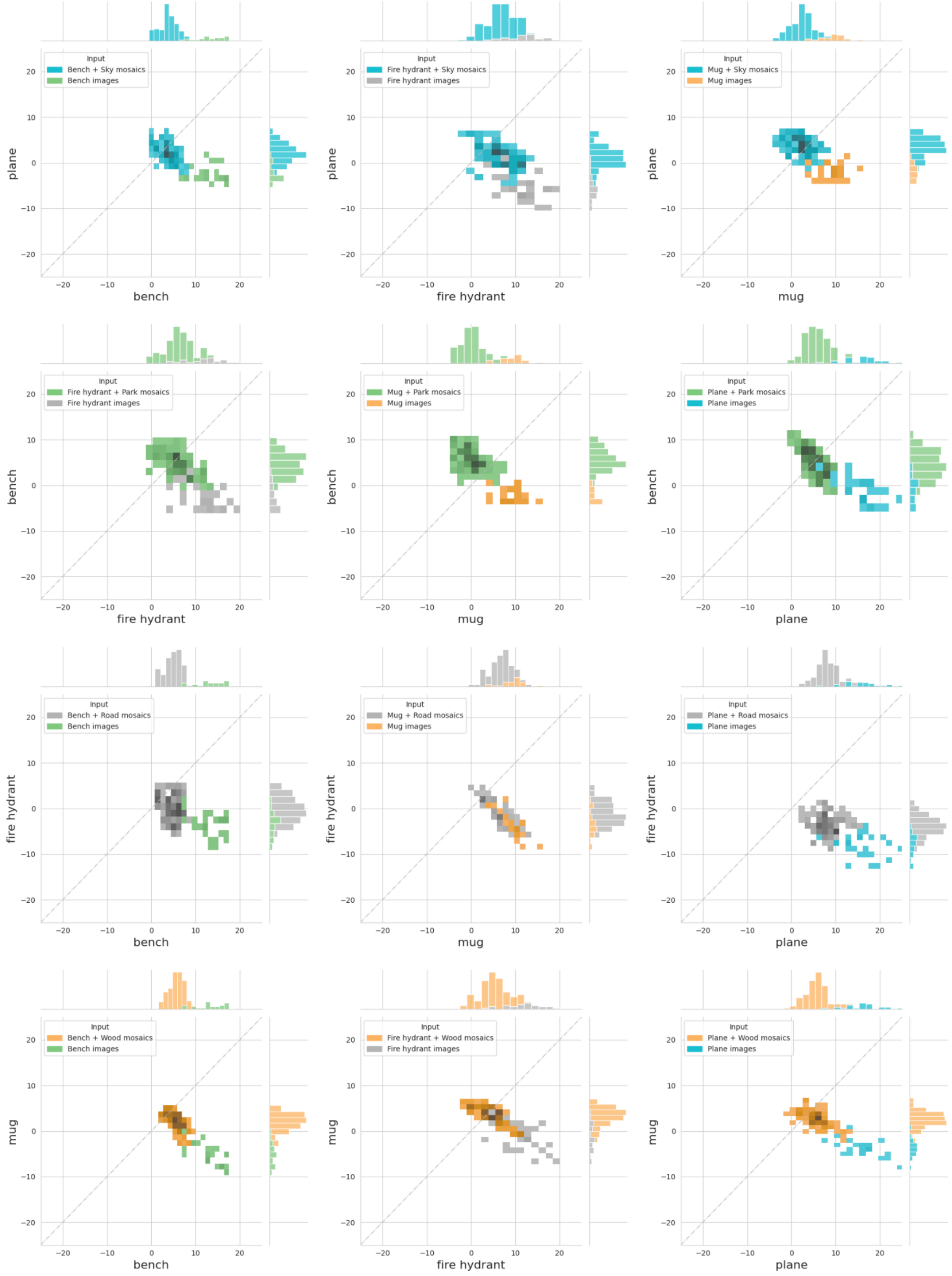


Figure 11.6: The logits obtained with the original images with respect to those obtained with the images combined with the different contexts are shown as histograms. In each row, the results are displayed for the mosaics composed of sky images (in the first row), park images (in the second row), road images (in the third row) and wood images (in the fourth row). The colour code used is the following: blue for mosaics with sky images and for plane images, green for mosaics with park images and for bench images, grey for wood mosaics and for fire hydrant images and orange for mosaics with wood context and mug images.

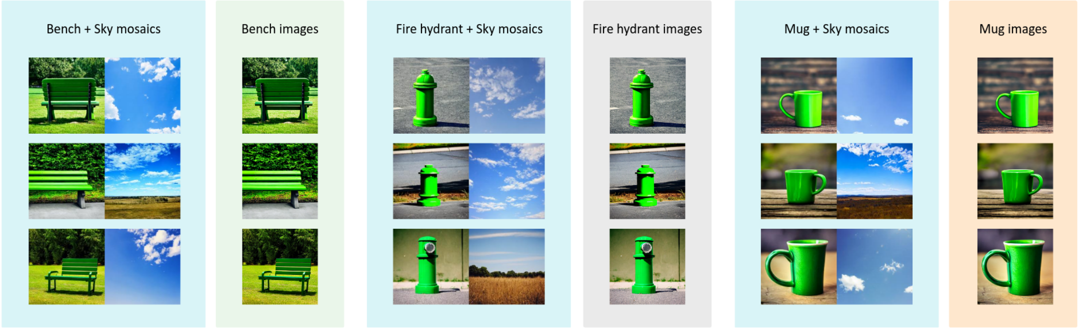


Figure 11.7: Examples of images and mosaics used to obtain the results of the first row of Figure 11.6: that is, using the sky context. The rectangle colors are the same as those used in Figure 11.6. As these mosaics were built with the sky context images (typical of the *plane* class), they are represented in blue. The single images of benches are in green, those of fire hydrants are in gray and those of mugs are in orange.

For the *bench* class, the road presence slightly reduces the confidence towards the *bench* class, obtaining higher logits for the *fire hydrant* in 14 out of 125 mosaics. In the mug case, only 9 mosaics obtained the *fire hydrant* logits greater than the *mug* logits. Finally, the road context combined with plane images does not change the prediction of any mosaic. This is likely due to the presence of the sky (*i.e.*, which is a very weighty feature for the model) in the plane images, and thus, favouring the *plane* class.

Wood. Although not as influential as the sky and the park contexts, the wooden context does seem to have a greater impact than the road context. When combined with bench images, the wooden context increase the logits value towards the *mug* class, obtaining 22 out of 125 mosaics greater logits for the *mug* class than for the *bench*. The impact is higher when combined with the *fire hydrant* class getting higher logits on the mug class in 38 mosaics (*i.e.*, 30.4%). And 28 mosaics combined with *plane* images get higher logits on the *mug* class.

In short, the main findings of these results are the following. The sky seems to be a relevant feature for the model when predicting the *plane* class. The park context (*i.e.*, the vegetation) is clearly a relevant characteristic for the prediction of the *bench* class: this could be a shortcut learned by the model. The road context is a characteristic that favours the *fire hydrant* class, but it does not seem to be so determinant for its prediction. Finally, the wood context is influential in the prediction of the *mug* class, although not as relevant as the sky for the *plane* class or the park for the *bench* class. We also observed that depending on the object class with which the contexts are combined, they have a greater or lesser effect. We observed how when contexts are combined with fire hydrant images the bias

Table 11.1: Euclidean distance of the distributions means to the diagonal for each of the combinations shown in Figure 11.6. The diagonal corresponds to the point where the logit values for the two classes coincide. For better interpretation, when the mean is located on the right of the diagonal, we present the results as positive. Conversely, if the mean of the logits distribution is located on the left of the diagonal, we show the results as negative. Mosaics with benches are highlighted in green. In grey the mosaics with fire hydrants and in orange the mosaics with mugs.

Bench + Sky mosaics	0.7165	Fire hydrant + Sky mosaics	2.5436	Mug + Sky mosaics	-0.7512
Bench images	7.9297	Fire hydrant images	8.7015	Mug images	5.6778
Fire hydrant + Park mosaics	0.7191	Mug + Park mosaics	-2.8234	Plane + Park mosaics	0.3372
Fire hydrant images	7.3755	Mug images	5.8023	Plane images	8.6055
Bench + Road mosaics	2.3406	Mug + Road mosaics	3.7252	Plane + Road mosaics	5.6887
Bench images	8.4977	Mug images	6.4769	Plane images	12.0303
Bench + Wood mosaics	1.7755	Fire hydrant + Wood mosaics	0.8831	Plane + Wood mosaics	1.4188
Bench images	9.2526	Fire hydrant images	6.6443	Plane images	9.8776

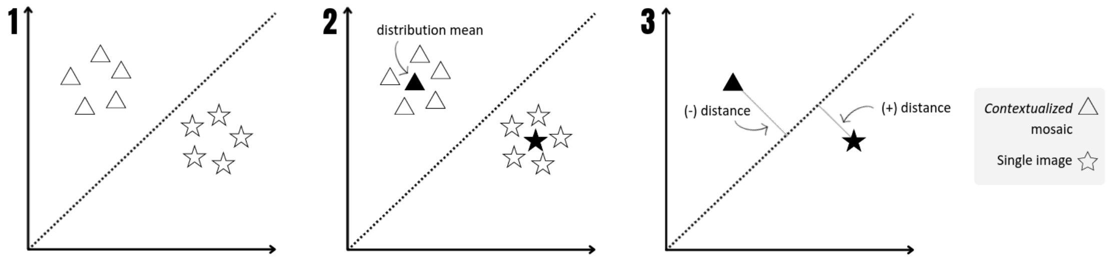


Figure 11.8: Steps to compute the distances shown in Table 11.1 and Figure 11.9. In the first step, the distributions of the single images and the mosaic distributions were calculated (*i.e.*, the same as those shown in Figure 11.6). In the second step, the distribution mean was obtained (filled triangle and star). Finally, the Euclidean distance from the mean to the diagonal was computed.

effect is lower. Or for example, the sky present in the plane images within the mosaics continue to favour the *plane* class.

Another way to reach these findings can be by analyzing the results of the *contextualized* mosaics using Table 11.1. This table shows the Euclidean distance of the means of the logit distributions of Figure 11.6 to the diagonal (see Figure 11.8, for a better comprehension of the steps followed). The diagonal is the point where the logits towards the two classes coincide. To better understand the results, we show the distances corresponding to the means in the part of the object class as positive. And as negative when the mean is in the part of the context class.

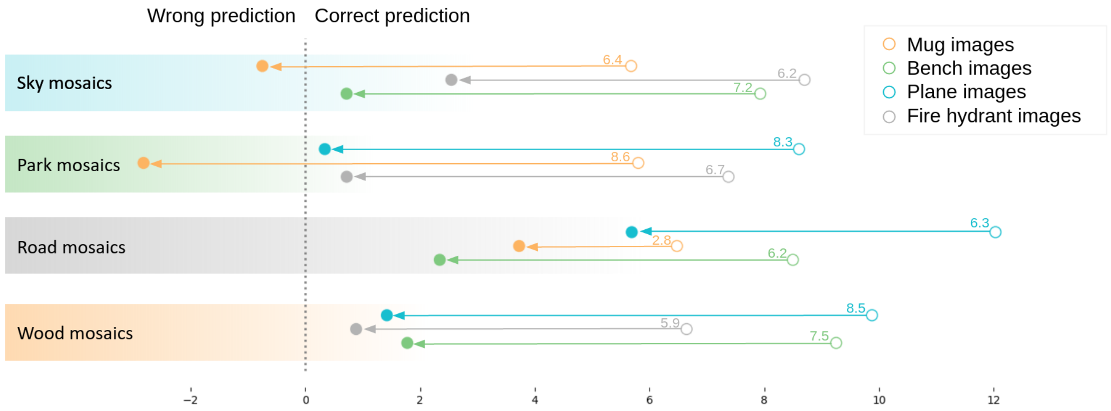


Figure 11.9: Visualization of the results of Table 11.1. The filled dots represent the mosaics. The empty dots correspond to the results of the single images. The diagonal is depicted by a dashed vertical line. The horizontal lines represent the difference between the two distances (*i.e.*, that of the single images to the diagonal and that of the mosaics to the diagonal). Greater values indicate a stronger influence of the context.

That is, the more negative the value, the more predictions there will be towards the context class. An illustration of these results is also shown in Figure 11.9.

As we already anticipated, if we look at the fire hydrant class (highlighted in grey in Table 11.1), we can observe that this is the class where contexts have the least influence. See how the distance is always positive and the greatest (among the other objects) in the case of the sky and park contexts (*i.e.*, 2.5436 and 0.7191). This could be because the model has learned some pattern of the object itself (or perhaps a bias from the fire hydrant context that we have not detected), and therefore in most cases, the model continues to predict the mosaics as *fire hydrants*. We can see that both the sky and the park context when combined with mug images (highlighted in orange) manage to move the mean of the distribution towards the prediction of the class of the context (*i.e.*, -0.7512 and -2.8234). That is, mosaics of mugs combined with sky images will be predicted mostly as *planes*, and mosaics of mugs with parks will be predicted mostly as *benches*, see filled orange points on the left side of Figure 11.9. On the other hand, the road context combined with any object obtains a mean distribution still far from zero (third row). In other words, the road context does not manage to move the distribution towards the prediction of the *fire hydrant* class.

In this section, we show two ways to analyze and visualize the results of the *contextualized* mosaics with the aim of evaluating the influence of context biases. After this first analysis, measures could be applied to mitigate the effect of context biases in this model-C if the domain expert considers it so.

11.2.1 | RESULTS CHECKS

Taking advantage of the availability of the *WB* dataset, in this section, we perform another experiment to further analyze the importance of the context and verify whether the results are consistent with those obtained with the *contextualized* mosaic methodology. To do so, for each context (*i.e.*, park, sky, road and wood), we paste an object of the remaining three classes, creating a total of 25 images per context-object pair. And we calculate the performance of the model for those new images. Notice that this is another way of performing the intervention explained in §11.1.4, that is fixing the four contexts intending to analyze the relationship between context C and the predicted class Y.

The results for each class are shown in the form of bar plots. For example, Figure 11.10 shows the number of images predicted as *plane*: (**a**) using 25 only context images, (**b**) 25 mug instances superposed to the sky context, (**c**) 25 fire hydrant images with sky context and (**d**) 25 bench images superposed to the sky context. The colour code of the following figures is green for the bench images and the park context, orange for the mug images and the wood context, grey for the fire hydrants and the road context, and blue for the plane images and the sky

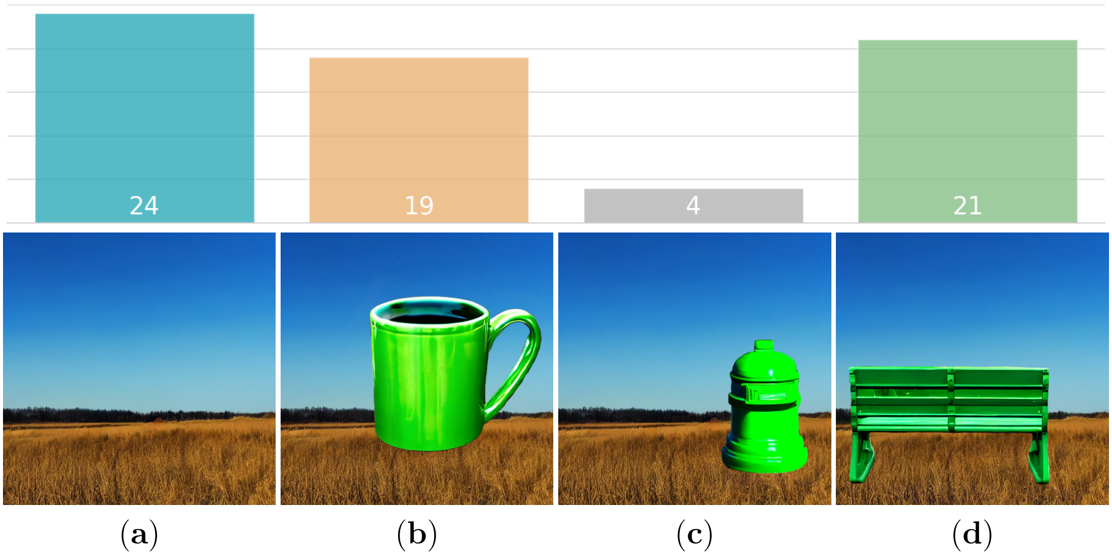


Figure 11.10: Number of images classified as *plane* with the sky context. The blue bar corresponds to results with only sky images, orange to the mug with sky context images, grey to a fire hydrant and green to the bench with sky context images. Instance examples used for each bar plot are shown at the bottom of the image (**a**) sky context image, (**b**) image of a mug superimposed on the sky context, (**c**) a fire hydrant (**d**) and a bench also superimposed on the sky image.

context. Let us now analyze the importance of context for each class.

Sky. These results confirm the findings obtained with the mosaics. It can be seen in Figure 11.10 how 24 sky images out of 25 were classified as planes. This means that the blue sky is an important feature for the *plane* class. Nevertheless, with only that experiment, we cannot affirm that this is an unwanted shortcut. However, 19 sky images with mugs (out of 25) were classified as *planes* and 21 with benches were also classified as *planes*. This, on the contrary, does confirm that this correlation learned by the model is not the intended behaviour. Notice that in the case of the fire hydrant, only 4 sky images with fire hydrant objects were classified as *planes*. This is consistent with previous results obtained with the mosaic analysis: the fire hydrant features are more relevant to the model than the presence of the sky.

Park. Also consistent with the mosaic results, vegetation is shown to be important to the *bench* class; see Figure 11.11. 25 images of only parks were predicted as *benches*. Also, the 25 mug objects superimposed on parks, 22 fire hydrants and 24 out of 25 planes over parks were classified as *benches*. That is to say, regardless of the object present in the park images, almost all of them were predicted as *benches*.

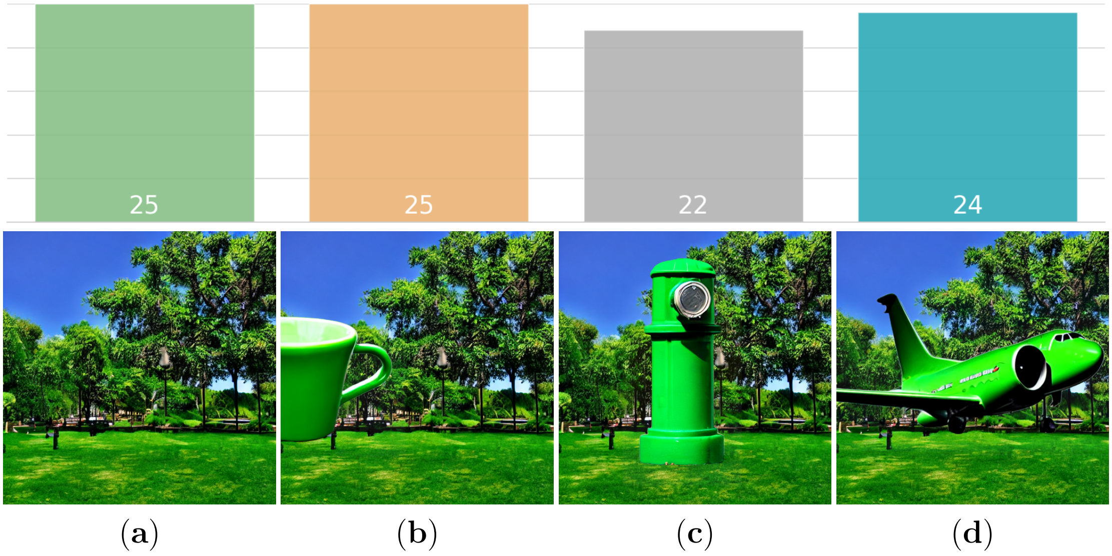


Figure 11.11: Number of images classified as *bench* with the park context. The green bar corresponds to results with only context images, orange to the mug with park context images, grey to a fire hydrant with park and blue to the plane with park context images. Same as before, instance examples used for each bar plot are shown at the bottom of the image.

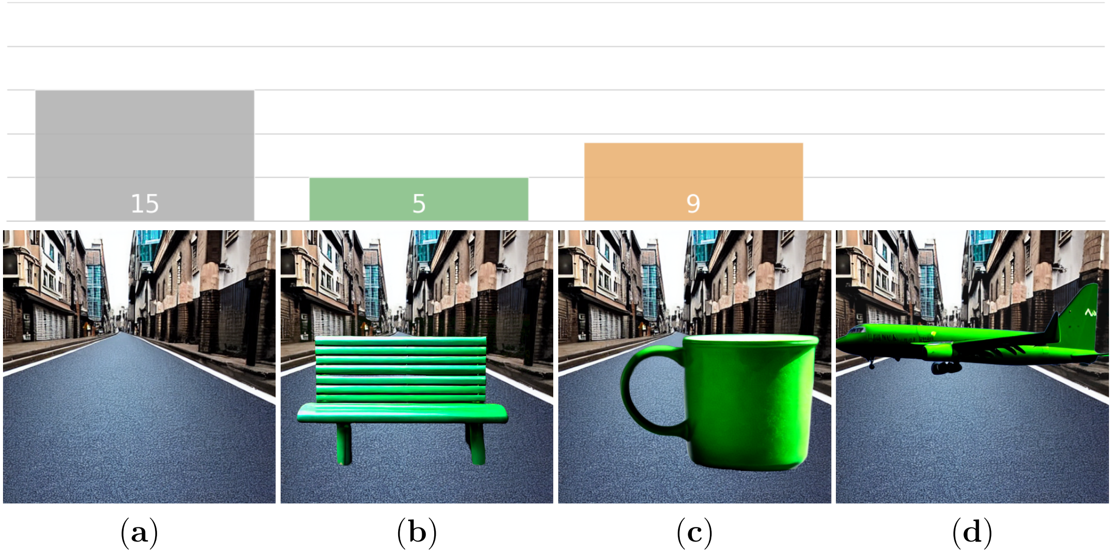


Figure 11.12: Number of images classified as *fire hydrant* with the road context. The grey bar corresponds to results with only road images, green to bench images and orange to mug images. The bar corresponding to the plane with road context obtains an accuracy of zero.

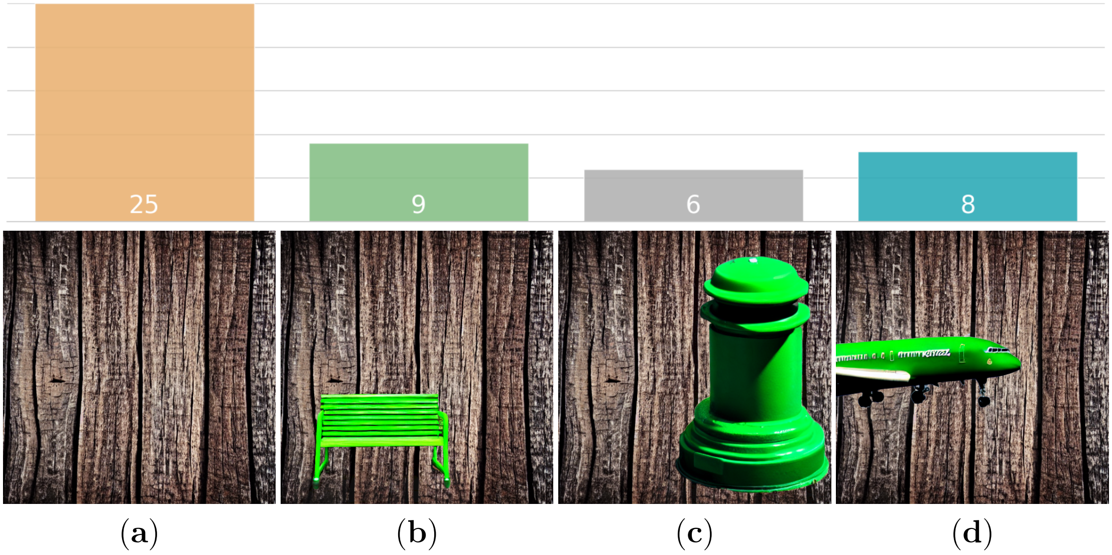


Figure 11.13: Number of images classified as *mug* with the wood context. The orange bar corresponds to results with only wood images, green to bench images, grey to fire hydrant images and blue to plane images.

Road. As already anticipated with the *contextualized* mosaics, the road context is not a relevant context for the *fire hydrant* class, or at least the model did not rely solely on the context. This can be simply observed in the grey bar plot in Figure 11.12: only 15 road images were predicted to be *fire hydrants*. In the case of the benches, only 5 of them, for the mugs 9 and in the case of the planes, none of the images were predicted as *fire hydrants*.

Wood. This context favours the *mug* class since as all the wooden images were classified as *mugs* (see Figure 11.13). However, when the different objects were pasted, the model was confused. The number of images classified as mugs decreased drastically: 9 bench images, 6 fire hydrant images and 8 plane images out of 25 were classified as *mugs*.

These findings are consistent with the *contextualized* mosaic outcomes, demonstrating their reliability and usefulness. The difficulty of having the segmented objects and being able to build the cross-context images makes this second experiment not viable in a real use-case scenario, instead, the mosaic creation is straightforward. Therefore, as long as we have identified possible sources of unwanted biases, we can build mosaics and use the proposed methodology to analyze their impact.

11.3 | SUMMARY OF THIS CHAPTER

In this chapter, we introduce a new way of using mosaics to analyze the impact of DATA/MODEL BIASES. In the experiment previously detailed, we focus on assessing the *context* influence on the model-C decisions. Unlike previous chapters, where we use XAI methods to asses biases, here we only use the model's output, avoiding the noise introduced by those XAI methods, and consequently also avoiding the bias introduced by humans (*i.e.*, HUMAN BIAS).

First, we presented three new datasets generated with a diffusion model: Context, No Context and White Background dataset. Then we introduce the *contextualized mosaics* methodology to analyse the relevance of *context* biases. The different steps followed are detailed in Figure 11.14. We start by identifying four potential *context* biases: the sky for the *plane* class, the park context for the *bench* class, the road for the *fire hydrant* class, and the wood for the *mug* class. Then we build the *contextualized* mosaics by combining those contexts with the original images. And finally, after analyzing the impact produced on the output these are the main findings. The park context was identified as a potential shortcut learned by the model to predict the *bench* class. To mitigate this shortcut, one could try to add more benches without a vegetation context, thus forcing the model to learn the bench characteristics and rely less on the vegetation when predicting the *bench* class. The sky also turned out to be an element that favoured the *plane* class. To

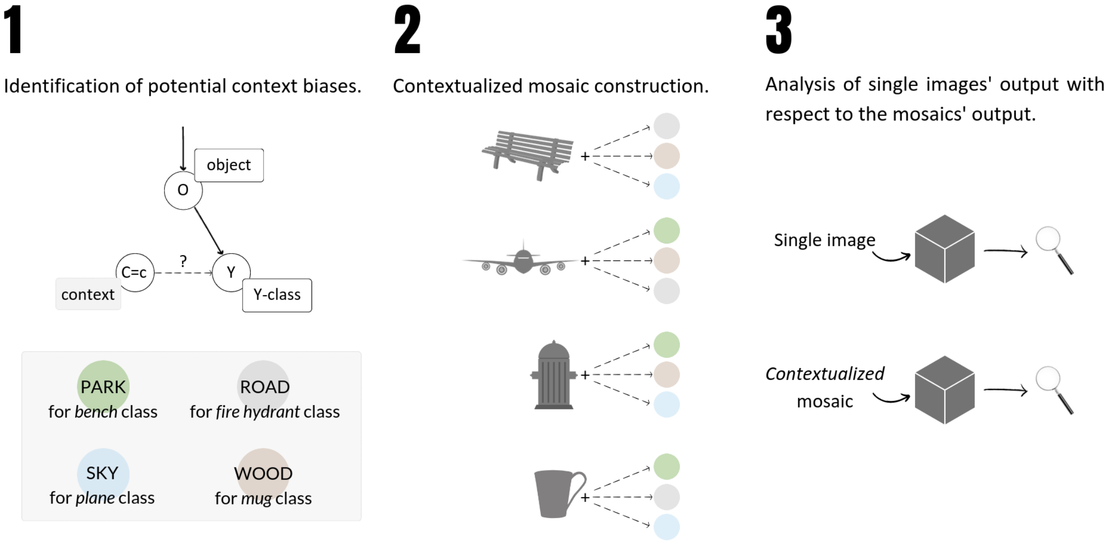

1
Identification of potential context biases.
object
O
C=c
?
Y
context
Y-class
PARK
for bench class
ROAD
for fire hydrant class
SKY
for plane class
WOOD
for mug class
2
Contextualized mosaic construction.
3
Analysis of single images' output with respect to the mosaics' output.
Single image
Contextualized mosaic

Figure 11.14: Steps of the proposed methodology. Firstly, four potential biases are identified: park for the *bench* class; road for the *fire hydrant* class; sky for the *plane* class; and wood for the *mug* class. Next, we construct the *contextualized* mosaics by combining each object with the three remaining contexts. Finally, the influence of the context is analyzed, by comparing the output of the single images with that of the mosaics.

prevent any object from being identified as a plane when having a sky in the background, one could add more images of the other classes with blue sky (*i.e.*, benches with a blue sky or mugs with a blue sky behind). The road does favour the fire hydrant prediction since all the fire hydrants are on the street, nevertheless, this bias could be considered not dangerous. This assessment of whether or not a bias can be harmful must be decided ultimately by the domain expert. Finally, the wood context does not seem to be a determinant shortcut in the prediction of the *mug* class. Being these context biases mitigated, the model will learn the characteristics of each object. Therefore, the model will generalize better, obtaining a high performance when tested with images from outside of the distribution (*e.g.*, when tested with the *NC* or *WB* sets).

To conclude, while this experiment focused on analysing the impact of *context* biases using mosaics, this methodology can be extended to examine other types of biases. For example, if textures could be a source of bias, mosaics could be created by combining the objects with the textures identified as potential biases.

Part IV.

Wrap-up

12 | CONCLUSIONS

> I have learned over the years that when one's mind is made up, this diminishes fear; knowing what must be done does away with fear.
>
> *Rosa Parks*

Throughout this thesis, we addressed the research question outlined in the Introduction 1 by presenting various methodologies for identifying, visualizing, and evaluating biases in datasets, models, and XAI methods. At the core of these methodologies lies the construction of mosaics, which serves as a technique to introduce a source of confusion. Leveraging this approach, we are able to assess the presence of different biases and their impact. Throughout each chapter, we already summarized the different conclusions of this thesis. Now, we aim to delve deeper into them and discuss both our future work and that of the community.

We group the outcomes into three subsections. Firstly, we outline our conclusions regarding the direction in which the explainability field should progress and the advancements that are essential to achieve the desired level of explainability. Secondly, we detail the conclusions regarding the evaluation of explainability methods. In order to reduce the noise introduced by these methods, we must evaluate them from two perspectives. Firstly, we must assess how effectively the XAI methods reflect the model's behaviour. Then, we must consider the level of interpretability they offer to the end user. Although in this thesis we only focus on the evaluation of faithfulness, in this chapter, we will also detail what are some of the results of the literature in the area of plausibility to have a broader perspective of the current state of this subfield, which is needed for the advancement of explainability in the right direction. Lastly, we will explore various conclusions related to biases and their detection. By understanding and mitigating biases effectively, we can strive for fair and unbiased models in their decision-making processes.

12.1 | EXPLAINABILITY TRENDS

Explainability is a growing area of research and development. The motivations behind this field are diverse, including compliance with legislation, the aim of building reliable and fair models, or the detection of biases and undesired behaviours in models. However, regardless of the underlying motivation, if the goal of the community is to build trustworthy models, from now on, explainability will be a requirement for any deployed model. Therefore, XAI should be integrated into various stages of the model development process to enhance transparency and interpretability: in the model design (*i.e.*, choosing models which are more explainable), in the training phase (*i.e.*, using XAI metrics to guide training) and in the selection process (*i.e.*, choosing the model also based on the quality of their produced explanations).

In Part I, we explored different families of *post-hoc* explainability methods. However, a key finding of this thesis is that relying solely on feature attribution methods is insufficient. In fact, alternative techniques to feature attribution methods have recently gained popularity among the explainability community. This trend reinforces the previous outcome.

As already discussed in Chapter 2, a way to enable us to gather more comprehensive insights into the model's behaviour could be done by integrating different types of *post-hoc* explanations (*e.g.*, feature-based, concept-based, counterfactual, etc.). In this direction, a recent work proposes a novel approach called CRAFT [26]. This method not only identifies the important concepts but also shows where these concepts are located in the input images.

This thesis focuses on *post-hoc* explainability, where explainability methods are applied once the model is already trained. Post-hoc methods have been extensively used in the literature since interpretable models were not as performant as black boxes. However, the number of interpretable-by-design models with comparable performance to opaque models is rising in the current landscape. Examples of these approaches are models that use prototype representations [19, 68, 67] or the Concept Bottleneck Models (CBM) [47, 88, 109] that map the input pixels to some concepts and the concepts to the target classes.

To summarize, we first proposed that an advantageous approach towards explainability would involve combining *post-hoc* techniques to attain more comprehensive and interpretable explanations. Additionally, instead of choosing between *post-hoc* explanations or inherently interpretable models, a beneficial approach could involve the combination of both methodologies.

12.2 | XAI ASSESSMENT

As outlined in this document, explainability methods have the potential to introduce noise into the pipeline, making it essential to assess whether these methods accurately represent the model's behaviour. Or on the contrary, if the explanations are misleading and cannot be utilized to comprehend the model's behaviour or identify biases in the data/models.

We first showed how the combination of *Focus* and mosaics serves as a tool to evaluate the faithfulness of these explainability methods. We also demonstrated the utility of this methodology not only in the CV domain but also across other modalities. If a XAI technique gets a random *Focus*, this technique can be deemed non-trustworthy. Nonetheless, to achieve a complete faithfulness assessment, we propose integrating various evaluation methodologies, such as localization techniques (*e.g.*, *Focus*), randomization methods, and axioms, among others. This integrated approach would contribute to standardizing evaluation metrics.

In this thesis, our primary focus lies in the initial part of the evaluation, which involves assessing the faithfulness of the XAI methods. However, as discussed earlier in this document, once we establish the technique's reliability, the subsequent step is to evaluate its plausibility. In other words, even if a method accurately reflects the model's behaviour, if it fails to help the end user comprehend the model, it becomes useless in terms of interpretability. However, these two features must be evaluated separately and do not have to be correlated. Indeed, existing reliability metrics exhibit a weak correlation with plausibility metrics [70, 45].

Although it is not the focus of this thesis, there are already studies in the literature that examine how useful explainability techniques are for the end user. For instance, in this work [45], the authors show the existence of confirmation bias when providing the users with explanations. Specifically, users tend to perceive predictions as correct when they are presented with explanations. In the same direction, this work [92] shows how the visual explanations tested did not help end users comprehend the model failures in the image classification task. Indeed, users exhibited better accuracy in predicting the incorrect class when the explanations were not provided compared to when they were provided. This shows that there is still room for improvement to achieve better XAI techniques.

To strike a balance between faithfulness to the model's behaviour and user usefulness, it is essential to work towards standardizing faithfulness and plausibility metrics. This standardization will provide a common framework to ensure more reliable approaches while facilitating end users' comprehension of the model.

12.3 | BIASED MODELS

All models exhibit biases. However, some of them are dangerously biased. Relying solely on a model's performance can be dangerous since DL models can often discover shortcuts that deviate from the intended solution.

As previously discussed in this document, the sources of these biases are diverse. It may be due to biases in the data since the model tends to inherit biases present in the training data. Algorithm biases can also emerge during model training, influenced by the chosen parameters and methodologies. Additionally, contextual biases arise from the attempt to create universal models, disregarding the distinct societal contexts in which these models are trained and deployed. Assuming a model trained in one context will apply to a completely different one can be ambitious and, in some cases, unrealistic.

These are just a few examples of sources of biases. With this understanding, the objective of the community is to identify and address these biases, aiming to mitigate their impact on model decisions. To achieve this, there is a need for bias localization and illustration methods that facilitate scalable and ongoing analysis of models. Consequently, bias identification will provide us with a clearer understanding of the factors and attributes influencing the model's decision-making process. Additionally, it could open the door to improving data quality.

In this thesis, we first address bias detection using explainability. Since one of the purposes of explainability is to uncover undesired biases. To achieve this, we leverage the *Focus* technique in conjunction with mosaics. The main objective is to semi-automate the bias detection process, minimizing human intervention until the final step, where the decision on whether the bias is desired or unwanted is made. We could improve the proposed automation techniques by trying to translate those potential biases into concepts (*e.g.*, through clustering) that are more human-understandable. This would enable humans to make informed decisions regarding the harmfulness of those biases.

We also explore an alternative approach to detect biases, circumventing the use of explainability. Instead, we utilize mosaics by combining potential biases with the original images to assess their impact on the final prediction. Although this method requires prior knowledge of possible biases, its main advantage lies in avoiding the noise introduced by explainability methods. In future work, we could improve this method by automatically detecting possible biases and then conforming the mosaics to analyze their relevance to the model.

Bias detection is crucial to achieve fair machine learning models and prevent them from having negative and illegal consequences. In fact, the widespread irruption, adoption and deployment of the Large Language Model (LLM) have brought the bias issue to the forefront of discussion. For instance, language models can perpetuate and strengthen social biases, foster polarization by creating echo chambers, and contribute to the spread of misinformation, negatively impacting society by deceiving public perception. As a result, the field of bias detection in AI, particularly in DL, is an active and rapidly evolving area of research.

12.4 | SUMMARY OF THIS CHAPTER

This chapter provides an overview of the key conclusions drawn from this dissertation. We first highlight the importance of explainability in developing trustworthy models, emphasizing its integration at different stages.

While this thesis focuses on *post-hoc* methodologies, particularly feature attribution techniques, we suggest combining various approaches for more comprehensive insights. We also note in this chapter the rise of interpretable models with performance comparable to black boxes.

Then, we discuss the potential noise introduced by explainability methods and how we tackle this by proposing an evaluation approach using *Focus* and mosaics to assess the faithfulness of the explanations. However, we also emphasize the importance of striking a balance between explainability techniques that faithfully mirror the model's behaviour and those that facilitate user understanding.

Finally, in this chapter, we devote a part to biases, recognizing their diverse sources and potential dangers. We explain how we leverage *Focus* and mosaics to semi-automate the process of bias detection. And how we explore an alternative method involving mosaics also to evaluate bias impact. Lastly, the significance of bias detection, particularly in the context of language models, is also highlighted due to their potential societal impact.

Overall, this chapter underscores the role of explainability and bias detection in developing reliable and fair AI models while emphasizing the ongoing evolution and future work of these research areas.

13 | RELATED CONTRIBUTIONS

> Women have an important contribution to make.
>
> *Margaret Mead*

The different works related to this thesis have been introduced throughout this document. However, this chapter is devoted to listing the publications and briefly explaining the contribution made.

13.1 | THE MAME DATASET [PUBLISHED]

In this work, we introduce a new dataset composed of images of artworks extracted from different museums. This new dataset—the Museum Art Medium (MAMe) dataset—comprises 29 medium classes (*i.e.*, techniques and materials). We train different models to check the feasibility and complexity of the proposed task and the impact of high-resolution and variable-shaped images. And finally, we apply explainability to evaluate the coherence between the characteristics considered important by the model and the characteristics that art experts consider important when differentiating the different classes. My main contribution to this work was the explainability part: planing the assessment, choosing the explainability method, adapting and implementing it and executing the experiments. I also contributed to the training of the models and the writing, visualization and reviewing part.

> Ferran Parés, Anna Arias-Duart, Dario Garcia-Gasulla, Gema Campo Francés, Nina Viladrich, Eduard Ayguadé, and Jesús Labarta. The MAMe dataset: On the relevance of High Resolution and Variable Shape image properties. *Applied Intelligence*, pages 1–22, 2022.

13.2 | THE FOCUS [PUBLISHED]

This paper presents the *Focus* score: an evaluation score for feature attribution methods. We empirically prove the robustness of the methodology by performing some sanity checks. We compare and evaluate different explainability techniques using the *Focus* score. And finally, we introduce the *Focus* application for bias detection. To facilitate the use of the *Focus* score we also added it to the Quantus [37] toolkit. Quantus is a public library where different XAI evaluation metrics have been implemented.

> Anna Arias-Duart, Ferran Parés, Dario Garcia-Gasulla, and Victor Giménez-Ábalos. Focus! Rating XAI Methods and Finding Biases. In *2022 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 1–8. IEEE, 2022.

13.3 | FOCUS AND BIAS [PUBLISHED]

This work analyses the *Focus* behaviour when applied to a biased model. To train the biased model, we first create a biased dataset. A dataset of cats and dogs in two contexts, outdoor and indoor. To train the biased model, we only use dogs outdoors and cats indoors. Once verified that the model is biased (*i.e.*, it learns to differentiate the classes by the context and not by the animal) we apply the *Focus* on top of that biased model. The results show how the *Focus* score decreases with the presence of the context bias.

> Anna Arias-Duart, Ferran Parés, Dario Garcia-Gasulla and Victor Giménez-Ábalos. Focus and Bias: Will it Blend? In *Artificial Intelligence Research and Development*, pages 325–334. IOS Press, 2022.

13.4 | A CONFUSION MATRIX [PUBLISHED]

In this work, we improve two of the main *Focus* limitations. On the one hand, the *Focus* presents some numerical instabilities due to divisions by zero (*e.g.*, when there is no positive relevance in the explanation). And secondly, the *Focus* does not consider the negative relevance of those methods that provide them. In this approach, we transform the evaluation problem into a classification problem, and we leverage to redefine the classification metrics, such as Focus-Accuracy, Focus-Recall, etc., to evaluate the feature attribution methods.

> Anna Arias-Duart, Ettore Mariotti, Dario Garcia-Gasulla and Jose Maria Alonso-Moral. A Confusion Matrix for Evaluating Feature Attribution Methods. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 3708-3713, June 2023.

13.5 | ASSESSING BIASES THROUGH VISUAL CONTEXTS [PUBLISHED]

In this paper, we propose a methodology to analyze context biases. For this purpose, we use mosaics, but in this case, we avoid using explainability methods. First, the datasets used in this work are generated using a diffusion model. All three datasets are publicly available: Context dataset, No Context dataset and White Background dataset. Then, we present the so-called contextualized mosaics, which we use to illustrate and visualize the relevance of context biases to the model. Finally, we also provide a public notebook tutorial for educational purposes available on Kaggle.

> Anna Arias-Duart, Victor Gimenez-Abalos, Ulises Cortés, and Dario Garcia-Gasulla. Assessing Biases through Visual Contexts. *Electronics*, 12(14): 3066, 2023.

13.6 | TEXTFOCUS [NOT YET SUBMITTED]

In this work, we extend the *Focus* score to the NLP domain, and we call this new score: the *TextFocus*. First, we present how we build the *textual mosaics*. Then, we introduce the modifications that must be considered when calculating the *TextFocus*, due to the requirements of *textual mosaics* (*e.g.*, variable length). Finally, we compare different explainability methods applied to text classification models. My contribution to this work took place at various stages of the process. I participated in conceptualising the methodology, part of the implementation and in the text writing and revision part.

> Ettore Mariotti, Anna Arias-Duart, Michele Cafagna, Dario Garcia-Gasulla and Jose Maria Alonso Moral. TextFocus: Assessing the Faithfulness of Feature Attribution Methods in Natural Language Processing. [not yet submitted]

BIBLIOGRAPHY

[1] Amina Adadi and Mohammed Berrada. Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI). *IEEE access*, 6:52138–52160, 2018. 3, 5, 17

[2] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity Checks for Saliency Maps. *Advances in neural information processing systems*, 31, 2018. 33, 35, 52, 53

[3] Naveed Akhtar. A survey of Explainable AI in Deep Visual Modeling: Methods and Metrics. *arXiv preprint arXiv:2301.13445*, 2023. 30

[4] Marco Ancona, Enea Ceolini, Cengiz Öztireli, and Markus Gross. Towards better understanding of gradient-based attribution methods for Deep Neural Networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30-May 3, 2018, Conference Track Proceedings*. OpenReview. net, 2018. 33, 35, 76

[5] Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. Machine bias. *ProPublica*. 1

[6] Anna Arias-Duart, Victor Gimenez-Abalos, Ulises Cortés, and Dario Garcia-Gasulla. Assessing Biases through Visual Contexts. *Electronics*, 12(14):3066, 2023. 99

[7] Anna Arias-Duart, Ettore Mariotti, Dario Garcia-Gasulla, and Jose Maria Alonso-Moral. A Confusion Matrix for Evaluating Feature Attribution Methods. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 3708–3713, June 2023. 66

[8] Anna Arias-Duart, Ferran Parés, Dario Garcia-Gasulla, and Victor Giménez-Ábalos. Focus! Rating XAI Methods and Finding Biases. In *2022 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 1–8. IEEE, 2022. 45, 55, 83

[9] Anna Arias-Duart, Ferran Parés, Víctor Giménez-Ábalos, and Dario Garcia-Gasulla. Focus and Bias: Will It Blend? In *Artificial Intelligence Research and Development*, pages 325–334. IOS Press, 2022. 86

[10] Leila Arras, Ahmed Osman, Klaus-Robert Müller, and Wojciech Samek. Evaluating Recurrent Neural Network Explanations. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 113–126, Florence, Italy, 2019. Association for Computational Linguistics. 75

[11] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador García, Sergio Gil-López, Daniel Molina, Richard Benjamins, et al. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information fusion*, 58:82–115, 2020. 3, 17

[12] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation. *PloS one*, 10(7):e0130140, 2015. 25, 27, 33, 35, 56

[13] Nicholas Baker, Hongjing Lu, Gennady Erlikhman, and Philip J Kellman. Deep convolutional networks do not classify based on global object shape. *PLoS computational biology*, 14(12):e1006613, 2018. XI, 7

[14] Jasmijn Bastings, Sebastian Ebert, Polina Zablotskaia, Anders Sandholm, and Katja Filippova. "Will You Find These Shortcuts?" A Protocol for Evaluating the Faithfulness of Input Salience Methods for Text Classification, November 2022. arXiv:2111.07367 [cs]. 75

[15] Francesco Bodria, Fosca Giannotti, Riccardo Guidotti, Francesca Naretto, Dino Pedreschi, and Salvatore Rinzivillo. Benchmarking and survey of explanation methods for black box models. *Data Mining and Knowledge Discovery*, pages 1–60, 2023. 17

[16] Joy Buolamwini and Timnit Gebru. Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification. In *Conference on fairness, accountability and transparency*, pages 77–91. PMLR, 2018. 81

[17] Diogo V Carvalho, Eduardo M Pereira, and Jaime S Cardoso. Machine learning interpretability: A survey on methods and metrics. *Electronics*, 8(8):832, 2019. 17

[18] Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In *2018 IEEE winter conference on applications of computer vision (WACV)*, pages 839–847. IEEE, 2018. 26, 27, 33, 35, 56

[19] Chaofan Chen, Oscar Li, Daniel Tao, Alina Barnett, Cynthia Rudin, and Jonathan K Su. This looks like that: Deep learning for interpretable image recognition. *Advances in neural information processing systems*, 32, 2019. 120

[20] Brian Christian. *The alignment problem: Machine learning and human values*. WW Norton & Company, 2020. 3

[21] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and Andre Van Schaik. EMNIST: Extending MNIST to handwritten letters. In *2017 international joint conference on neural networks (IJCNN)*, pages 2921–2926. IEEE, 2017. 21

[22] Roxana Daneshjou, Kailas Vodrahalli, Weixin Liang, Roberto A Novoa, Melissa Jenkins, Veronica Rotemberg, Justin Ko, Susan M Swetter, Elizabeth E Bailey, Olivier Gevaert, et al. Disparities in dermatology ai: Assessments using diverse clinical images. *arXiv preprint arXiv:2111.08006*, 2021. 82

[23] Jeffrey Dastin. Amazon scraps secret AI recruiting tool that showed bias against women. *Reuters*. 1

[24] Misha Denil, Alban Demiraj, and Nando de Freitas. Extraction of Salient Sentences from Labelled Documents, 2015. arXiv:1412.6815 [cs]. 75

[25] Amit Dhurandhar, Pin-Yu Chen, Ronny Luss, Chun-Chen Tu, Paishun Ting, Karthikeyan Shanmugam, and Payel Das. Explanations based on the missing: Towards contrastive explanations with pertinent negatives. *Advances in neural information processing systems*, 31, 2018. 21

[26] Thomas Fel, Agustin Picard, Louis Bethune, Thibaut Boissin, David Vigouroux, Julien Colin, Rémi Cadène, and Thomas Serre. Craft: Concept recursive activation factorization for explainability. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2711–2721, 2023. 120

[27] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020. 3

[28] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations*, 2018. 7

[29] Amirata Ghorbani, James Wexler, James Y Zou, and Been Kim. Towards automatic concept-based explanations. *Advances in Neural Information Processing Systems*, 32, 2019. 19, 20

[30] Madelyn Glymour, Judea Pearl, and Nicholas P Jewell. *Causal inference in statistics: A primer.* John Wiley & Sons, 2016. 103

[31] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020. 21

[32] Bryce Goodman and Seth Flaxman. Eu regulations on algorithmic decision-making and a "right to explanation". *AI Magazine*, 38, 06 2016. 5

[33] Yash Goyal, Ziyan Wu, Jan Ernst, Dhruv Batra, Devi Parikh, and Stefan Lee. Counterfactual visual explanations. In *International Conference on Machine Learning*, pages 2376–2384. PMLR, 2019. 21

[34] Mara Graziani, Lidia Dutkiewicz, Davide Calvaresi, José Pereira Amorim, Katerina Yordanova, Mor Vered, Rahul Nair, Pedro Henriques Abreu, Tobias Blanke, Valeria Pulignano, et al. A global taxonomy of interpretable AI: unifying the terminology for the technical and social sciences. *Artificial intelligence review*, pages 1–32, 2022. XI, 3, 4

[35] Riccardo Guidotti. Counterfactual explanations and how to find them: literature review and benchmarking. *Data Mining and Knowledge Discovery*, pages 1–55, 2022. 21

[36] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 56, 88

[37] Anna Hedström, Leander Weber, Daniel Krakowczyk, Dilyara Bareeva, Franz Motzkus, Wojciech Samek, Sebastian Lapuschkin, and Marina Marina M.-C. Höhne. Quantus: An Explainable AI Toolkit for Responsible Evaluation of Neural Network Explanations and Beyond. *Journal of Machine Learning Research*, 24(34):1–11, 2023. 30, 126

[38] Joseph Henrich, Steven J Heine, and Ara Norenzayan. The weirdest people in the world? *Behavioral and brain sciences*, 33(2-3):61–83, 2010. 81

[39] High-Level Expert Group. *Ethic Guidelines for Trustworthy AI.* European Union, 2019. 4

[40] Sara Hooker, Dumitru Erhan, Pieter-Jan Kindermans, and Been Kim. A benchmark for interpretability methods in deep neural networks. *Advances in neural information processing systems*, 32, 2019. 34, 35

[41] Alon Jacovi and Yoav Goldberg. Towards Faithfully Interpretable NLP Systems: How Should We Define and Evaluate Faithfulness? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4198–4205, 2020. 30

[42] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873):583–589, 2021. 1

[43] Eoin M Kenny and Mark T Keane. On generating plausible counterfactual and semi-factual explanations for deep learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 11575–11585, 2021. 21

[44] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, and Rory Sayres. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav), 2017. 19

[45] Sunnie SY Kim, Nicole Meister, Vikram V Ramaswamy, Ruth Fong, and Olga Russakovsky. Hive: Evaluating the human interpretability of visual explanations. In *European Conference on Computer Vision*, pages 280–298. Springer, 2022. 121

[46] Pieter-Jan Kindermans, Sara Hooker, Julius Adebayo, Maximilian Alber, Kristof T Schütt, Sven Dähne, Dumitru Erhan, and Been Kim. The (un)

reliability of saliency methods. *Explainable AI: Interpreting, explaining and visualizing deep learning*, pages 267–280, 2019. 32, 35

[47] Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *International conference on machine learning*, pages 5338–5348. PMLR, 2020. 120

[48] Maximilian Kohlbrenner, Alexander Bauer, Shinichi Nakajima, Alexander Binder, Wojciech Samek, and Sebastian Lapuschkin. Towards best practice in explaining neural network decisions with LRP. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. IEEE, 2020. 31, 35

[49] Narine Kokhlikyan, Vivek Miglani, Miguel Martin, Edward Wang, Bilal Alsallakh, Jonathan Reynolds, Alexander Melnikov, Natalia Kliushkina, Carlos Araya, Siqi Yan, and Orion Reblitz-Richardson. Captum: A unified and generic model interpretability library for PyTorch, 2020. 56

[50] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123(1):32–73, 2017. 87

[51] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012. 56, 101

[52] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 21

[53] Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the Loss Landscape of Neural Nets. *arXiv:1712.09913 [cs, stat]*, November 2018. arXiv: 1712.09913. 75

[54] Weixin Liang and James Zou. Metashift: A Dataset of Datasets for Evaluating Contextual Distribution Shifts and Training Conflicts. In *International Conference on Learning Representations*, 2021. 87

[55] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 87

[56] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017. 24, 27, 76

[57] Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. Learning Word Vectors for Sentiment Analysis, 2011. 76

[58] Ettore Mariotti, Anna Arias-Duart, Michele Cafagna, Dario Garcia-Gasulla, and Jose Maria Alonso-Moral. TextFocus: Assessing the Faithfulness of Feature Attribution Methods in Natural Language Processing. In *[not yet submitted]*, 2023. 71

[59] David Martens and Foster Provost. Explaining data-driven document classifications. *MIS quarterly*, 38(1):73–100, 2014. 21

[60] Vivek Miglani, Narine Kokhlikyan, Bilal Alsallakh, Miguel Martin, and Orion Reblitz-Richardson. Investigating Saturation Effects in Integrated Gradients. *arXiv preprint arXiv:2010.12697*, 2020. 27, 61

[61] Tim Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial intelligence*, 267:1–38, 2019. 3

[62] Grégoire Montavon, Alexander Binder, Sebastian Lapuschkin, Wojciech Samek, and Klaus-Robert Müller. Layer-wise relevance propagation: an overview. *Explainable AI: interpreting, explaining and visualizing deep learning*, pages 193–209, 2019. 25, 40

[63] Grégoire Montavon, Sebastian Lapuschkin, Alexander Binder, Wojciech Samek, and Klaus-Robert Müller. Explaining nonlinear classification decisions with deep taylor decomposition. *Pattern Recognition*, 65:211–222, 2017. 56

[64] Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. Methods for interpreting and understanding deep neural networks. *Digital signal processing*, 73:1–15, 2018. 3

[65] W James Murdoch, Chandan Singh, Karl Kumbier, Reza Abbasi-Asl, and Bin Yu. Definitions, methods, and applications in interpretable machine learning. *Proceedings of the National Academy of Sciences*, 116(44):22071–22080, 2019. 3

[66] Woo-Jeoung Nam, Shir Gur, Jaesik Choi, Lior Wolf, and Seong-Whan Lee. Relative attributing propagation: Interpreting the comparative contributions

of individual units in deep neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 2501–2508, 2020. 56

[67] Meike Nauta, Jörg Schlötterer, Maurice van Keulen, and Christin Seifert. PIP-Net: Patch-Based Intuitive Prototypes for Interpretable Image Classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2744–2753, 2023. 120

[68] Meike Nauta, Ron Van Bree, and Christin Seifert. Neural prototype trees for interpretable fine-grained image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14933–14943, 2021. 120

[69] An-phi Nguyen and María Rodríguez Martínez. On quantitative aspects of model interpretability. *arXiv preprint arXiv:2007.07584*, 2020. 32

[70] Giang Nguyen, Daeyoung Kim, and Anh Nguyen. The effectiveness of feature attribution methods and its correlation with automatic evaluation scores. *Advances in Neural Information Processing Systems*, 34:26422–26436, 2021. 121

[71] Raymond S Nickerson. Confirmation bias: A ubiquitous phenomenon in many guises. *Review of general psychology*, 2(2):175–220, 1998. 9

[72] Weili Nie, Yang Zhang, and Ankit Patel. A theoretical explanation for perplexing behaviors of backpropagation-based visualizations. In *International Conference on Machine Learning*, pages 3809–3818. PMLR, 2018. 27

[73] Eirini Ntoutsi, Pavlos Fafalios, Ujwal Gadiraju, Vasileios Iosifidis, Wolfgang Nejdl, Maria-Esther Vidal, Salvatore Ruggieri, Franco Turini, Symeon Papadopoulos, Emmanouil Krasanakis, et al. Bias in data-driven artificial intelligence systems—An introductory survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 10(3):e1356, 2020. 81

[74] Jose Oramas, Kaili Wang, and Tinne Tuytelaars. Visual Explanation by Interpretation: Improving Visual Feedback Capabilities of Deep Neural networks. In *International Conference on Learning Representations*, 2018. 32, 35

[75] Sebastian Palacio, Adriano Lucieri, Mohsin Munir, Sheraz Ahmed, Jörn Hees, and Andreas Dengel. XAI handbook: Towards a Unified Framework for Explainable AI. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3766–3775, 2021. 3

[76] Ferran Parés, Anna Arias-Duart, Dario Garcia-Gasulla, Gema Campo-Francés, Nina Viladrich, Eduard Ayguadé, and Jesús Labarta. The MAMe dataset: On the relevance of High Resolution and Variable Shape image properties. *Applied Intelligence*, pages 1–22, 2022. 39, 56

[77] Vitali Petsiuk, Abir Das, and Kate Saenko. RISE: Randomized Input Sampling for Explanation of Black-box Models. *arXiv preprint arXiv:1806.07421*, 2018. 24, 27

[78] Ariadna Quattoni and Antonio Torralba. Recognizing indoor scenes. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 413–420. IEEE, 2009. 56

[79] Sukrut Rao, Moritz Böhle, and Bernt Schiele. Towards Better Understanding Attribution Methods. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10223–10232, 2022. 23

[80] Sashank J Reddi, Satyen Kale, and Sanjiv Kumar. On the Convergence of Adam and Beyond. In *International Conference on Learning Representations*, 2018. 56, 88

[81] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "Why Should I Trust You?" Explaining the Predictions of Any Classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016. 23, 27, 29, 56, 76

[82] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-Resolution Image Synthesis With Latent Diffusion Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, June 2022. 99

[83] Yao Rong, Tobias Leemann, Vadim Borisov, Gjergji Kasneci, and Enkelejda Kasneci. A Consistent and Efficient Evaluation Strategy for Attribution Methods. In *International Conference on Machine Learning*, pages 18770–18795. PMLR, 2022. 34, 35

[84] Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215, 2019. 3

[85] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015. 22, 50, 56, 81, 101

[86] Wojciech Samek, Alexander Binder, Grégoire Montavon, Sebastian Lapuschkin, and Klaus-Robert Müller. Evaluating the visualization of what a deep neural network has learned. *IEEE transactions on neural networks and learning systems*, 28(11):2660–2673, 2016. 33, 35

[87] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019. 76

[88] Yoshihide Sawada and Keigo Nakamura. Concept bottleneck model with additional unsupervised concepts. *IEEE Access*, 10:41758–41765, 2022. 120

[89] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*, 128(2):336–359, Oct 2019. 26, 27, 29, 50, 55

[90] Shreya Shankar, Yoni Halpern, Eric Breck, James Atwood, Jimbo Wilson, and D Sculley. No Classification without Representation: Assessing Geodiversity Issues in Open Data Sets for the Developing World. *arXiv preprint arXiv:1711.08536*, 2017. 81

[91] Lloyd S Shapley et al. A value for n-person games. 1953. 24

[92] Hua Shen and Ting-Hao Huang. How Useful are the Machine-Generated Interpretations to General Users? A Human Evaluation on Guessing the Incorrectly Predicted Labels. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, volume 8, pages 168–172, 2020. 121

[93] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *International conference on machine learning*, pages 3145–3153. PMLR, 2017. 26, 27, 60, 76

[94] Avanti Shrikumar, Peyton Greenside, Anna Shcherbina, and Anshul Kundaje. Not just a black box: Learning important features through propagating activation differences. *CoRR*, abs/1605.01713, 2016. 24, 27

[95] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016. 1

[96] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. *CoRR*, abs/1312.6034, 2013. XII, 24, 25, 27, 75

[97] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional networks for large-scale image recognition. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. 40, 50, 56

[98] Leon Sixt, Maximilian Granz, and Tim Landgraf. When explanations lie: Why many modified bp attributions fail. In *International Conference on Machine Learning*, pages 9046–9057. PMLR, 2020. 33, 35

[99] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise, 2017. 25, 27, 56

[100] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. 73, 76

[101] Jost Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. 12 2014. XII, 25, 27

[102] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, pages 3319–3328. PMLR, 2017. 25, 27, 32, 35, 56, 75

[103] Harini Suresh and John Guttag. A Framework for Understanding Sources of Harm throughout the Machine Learning Life Cycle. In *Equity and access in algorithms, mechanisms, and optimization*, pages 1–9. 2021. 81, 82

[104] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016. 52

[105] Tom Vermeire, Dieter Brughmans, Sofie Goethals, Raphael Mazzine Barbossa de Oliveira, and David Martens. Explainable image classification with

evidence counterfactual. *Pattern Analysis and Applications*, 25(2):315–335, 2022. 21, 22

[106] Sandra Wachter, Brent Mittelstadt, and Luciano Floridi. Why a right to explanation of automated decision-making does not exist in the general data protection regulation. *International Data Privacy Law*, 7(2):76–99, 2017. 5

[107] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The caltech-ucsd birds-200-2011 dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011. 22

[108] Haofan Wang, Zifan Wang, Mengnan Du, Fan Yang, Zijian Zhang, Sirui Ding, Piotr Mardziel, and Xia Hu. Score-CAM: Score-weighted visual explanations for convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 24–25, 2020. 26, 27

[109] Yue Yang, Artemis Panagopoulou, Shenghao Zhou, Daniel Jin, Chris Callison-Burch, and Mark Yatskar. Language in a bottle: Language model guided concept bottlenecks for interpretable image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19187–19197, 2023. 120

[110] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 818–833, Cham, 2014. Springer International Publishing. XII, 24, 25, 27

[111] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014. 23, 27

[112] Matthew D. Zeiler, Graham Taylor, and Rob Fergus. Adaptive deconvolutional networks for mid and high level feature learning. volume 2011, pages 2018–2025, 11 2011. 24

[113] Jianming Zhang, Sarah Adel Bargal, Zhe Lin, Jonathan Brandt, Xiaohui Shen, and Stan Sclaroff. Top-down neural attention by excitation backprop. *International Journal of Computer Vision*, 126(10):1084–1102, 2018. 31, 35

[114] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016. 26

[115] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million Image Database for Scene Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. 56, 60

WEBPAGE REFERENCES

[116] 2022 ICLR Keynote. https://youtu.be/Ub45cGEcTB0. Accessed: April 2023. 5

[117] Cleveland Museum: Open access. https://www.clevelandart.org/open-access. Accessed: April 2020. 39

[118] LACMA Launches New Collections Online Website. https://www.lacma.org/press/lacma-launches-new-collections-online-website. Accessed: April 2020. 39

[119] Met Museum: Image and Data Resources. https://www.metmuseum.org/about-the-met/policies-and-documents/image-resources. Accessed: April 2020. 39

[120] Proposal for a regulation of the European Parliament and of the Council laying down harmonised rules on Artificial Intelligence (Artificial Intelligence Act) and amending certain Union Legislative Acts. https://www.europarl.europa.eu/meetdocs/2014_2019/plmrep/COMMITTEES/CJ40/DV/2023/05-11/ConsolidatedCA_IMCOLIBE_AI_ACT_EN.pdf, 2021. Accessed: May 2023. 5

ACRONYMS

ACE Automatic Concept-based Explanation

ACE Automatic Concept-based Explanation

AI Artificial Intelligence

CAM Class Activation Mapping

CAV Concept Activation Vector

CBM Concept Bottleneck Models

CEM Contrastive Explanations Method

CNNs Convolutional Neural Networks

CNN Convolutional Neural Networks

CVE Counterfactual Visual Explanations

CV Computer Vision

DAG Directed Acyclic Graph

DeepLIFT Deep Learning Important FeaTures

DL Deep Learning

EU European Union

FS fixed shape

GAN Generative Adversarial Network

GAP Global Average Pooling

GB Guided Backpropagation

GDPR General Data Protection Regulation

GPT-3 Generative Pre-trained Transformer 3

GradCAM Gradient-weighted Class Activation Mapping

HLEG-AI High-Level Expert Group on Artificial Intelligence

HR high resolution

IG Integrated Gradients

IMDB Internet Movie Database

IoU Intersection Over Union

LACMA Los Angeles County Museum of Art

LGTBI Lesbian, Gay, Bisexual, Transgender, and Intersex

LIME Local Interpretable Model-agnostic Explanations

LLM Large Language Model

LRP Layer-Wise Relevance Propagation

LR low resolution

MAMe Museum Art Medium

ML Machine Learning

NLP Natural Language Processing

NNs Neural Networks

PIECE PlausIble Exceptionality-based Contrastive Explanations

ReLU Rectified Linear Unit

RISE Randomized Input Sampling for Explanation

ROAD RemOve And Debias

ROAR RemOve And Retrain

SEDC-T Search for EviDence Counterfactual with Target counterfactual class

SEDC Search for Explanations for Document Classification

SHAP Shapley Additive Explanations

SST-2 Standford Sentiment

TCAV Testing with Concept Activation Vector

UB Universitat de Barcelona

USA United States of America

VS variable shape

XAI eXplainable Artificial Intelligence