Master's thesis

Master's Programme in Information Technology, 120 credits

Towards gradient faithfulness and beyond

Computer Science and Engineering, 30 credits

Halmstad, 2023-08-07
Vincenzo Buono, Isak Åkesson

HALMSTAD UNIVERSITY

HALMSTAD UNIVERSITY

MASTER THESIS

Master in Computer Science and Engineering

Towards gradient faithfulness and beyond

Vincenzo Buono, Isak Åkesson

SUPERVISORS:
Mahmoud Rahat
Peyman Mashhadi

EXAMINER:
Eren Erdal Aksoy

Computer Science and Engineering
Halmstad University, July 08, 2023

ABSTRACT

The riveting interplay of industrialization, informalization, and exponential technological growth of recent years has shifted the attention from *classical machine learning techniques* to more *sophisticated deep learning* approaches; yet its intrinsic *black-box* nature has been impeding its widespread adoption in transparency-critical operations. In this rapidly evolving landscape, where the symbiotic relationship between research and practical applications has never been more interwoven, the contribution of this paper is twofold: advancing gradient faithfulness of *CAM* methods and exploring new frontiers beyond it.

In the first part, we theorize three novel *gradient-based CAM* formulations, aimed at replacing and superseding traditional *Grad-CAM*-based methods by tackling and addressing the intricately and persistent *vanishing* and *saturating* gradient problems. As a consequence, our work introduces novel enhancements to *Grad-CAM* that reshape the conventional gradient computation by incorporating a customized and adapted technique inspired by the well-established and provably *Expected Gradients' difference-from-reference* approach. Our proposed techniques– Expected Grad-CAM, Expected Grad-CAM++ and Guided Expected Grad-CAM– as they operate directly on the gradient computation, rather than the recombination of the weighing factors, are designed as a direct and seamless replacement for *Grad-CAM* and any posterior work built upon it.

In the second part, we build on our prior proposition and devise a novel *CAM* method that produces both *high-resolution* and *class-discriminative* explanation without fusing other methods, while addressing the issues of both *gradient* and *CAM* methods altogether. Our last and most advanced proposition, Hyper Expected Grad-CAM, challenges the current state and formulation of *visual explanation* and *faithfulness* and produces a new type of *hybrid* saliencies that satisfy the notion of *natural encoding* and *perceived resolution*. By rethinking *faithfulness* and *resolution* is possible to generate saliencies which are more *detailed, localized*, and *less noisy*, but most importantly that are composed of only concepts that are encoded by the *layerwise* models' understanding.

Both contributions have been quantitatively and qualitatively compared and assessed in a 5 to 10 times larger evaluation study on the ILSVRC2012 dataset against nine of the most recent and performing CAM techniques across six metrics. Expected Grad-CAM outperformed not only the original formulation but also more advanced methods, resulting in the *second-best* explainer with an *Ins-Del* score of 0.56. Hyper Expected Grad-CAM provided remarkable result across each quantitative metrics, yielding a 0.15 increase in *insertion* when compared to the *highest scoring explainer PolyCAM*, totaling to an *Ins-Del* score of 0.72.

ACKNOWLEDGEMENTS

Special thanks to our supervisors Mahmoud Rahat and Peyman Mashhadi who guided us through this journey. Also many thanks to our examiner Eren Erdal Aksoy and course coordinator Slawomir Nowaczyk for their helpful feedback.

– V.B, I.A

A loro che ci sono Sempre. Da V.

– V.B

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

I4.0 Industry 4.0

I5.0 Industry 5.0

XAI Explainable AI

PdM Predictive Maintenance

IoT Internet of Things

SoA State-of-the-art

AI Artificial Intelligence

DL Deep Learning

DNN Deep Neural Network

ReLU Rectified Linear Activation Function

RUL Remaining Useful Life

HI Health Index

RNN Recurrent Neural Network

DeconvNet Deconvolution network

ConvNet Convolution network

CNN Convolutional Neural Network

CAM Class Activation Map

GAP Global Average Pooling

GMP Global Max Pooling

Grad-CAM Gradient-weighted Class Activation Mapping

Grad-CAM++ Gradient-weighted Class Activation Mapping

IG Integrated Gradients

EG Expected Gradients

LSTM Long short-term memory

C-MAPSS Commercial Modular Aero-Propulsion System Simulation

ILSVRC2012 ImageNet Large Scale Visual Recognition Challenge

Expected Grad-CAM Expected Gradient-weight Class Activation Mapping

Expected Grad-CAM Expected Gradient-weight Class Activation Mapping

Expected Grad-CAM++ Expected Gradient-weight Class Activation Mapping++

Guided Expected Grad-CAM Guided Expected Gradient-weight Class Activation Mapping

Hyper Expected Grad-CAM Hyper Expected Grad-CAM

1 INTRODUCTION

The convergence of the rapid advancements in industrialization and informalization, that characterized the past years, have catalyzed an unmatched leap forward in the manufacturing technologies. *Industry* 4.0 (I4.0) [47], based on *cyber-physical* systems and industrial *Internet of Things* (IoT), combined software, sensors, and intelligent controls units to improve industrial processes [53]. With the objective of increasing efficiency through automation, I4.0 propelled a new era of exponential growth of sensors' use [29], which led to an unprecedented production of *industrial big data* [67]. As a result, this enabled the automation of *Predictive Maintenance* (PdM) by analyzing a massive amount of process and related data [53]. Maintenance optimization is a crucial aspect for industrial companies and is considered of utmost importance, as it can reduce costs by up to 60%[13], by effectively detecting and addressing machine failures. *Predictive Maintenance* (PdM) is the most effective of these techniques, as it "maximizes the working life of components by taking advantage of their unexploited lifetime potential" [53]. The exponential rise of computation power, in conjunction with the intrinsic unparallel ability of automatically *extract*, and *create latent features*, has shifted the attention from classical machine learning techniques to more sophisticated *deep learning* approaches [53]. These models are capable of achieving remarkable results and are currently posed as *State-of-the-art* (SoA) solutions [53]. However, due to their *black-box* nature, they lack the transparency for direct interpretation, which represents a key impediment [4] to more widespread adoption, especially in scenarios where their predictions have the potential to significantly impact critical and costly operations. At the cusp of the *fifth industrial revolution* (I5.0), it has become increasingly evident that the model's *interpretability* is just as important as its performances. This is especially relevant, as the successful integration and deployment of these models, within real-world applications, highly rely on their reliability and trustworthiness. In this sense, *Explainable AI* (XAI) is the backbone of the *Industry* 4.0 (I4.0) and *Industry* 5.0 (I5.0), endowing Artificial Intelligence (AI) solutions with the ability to "narrate" the path taken to arrive at a solution, rather than simply providing a solution.

1.1 PROBLEM FORMULATION

Within the domain of PdM, *deep learning-based* (DL) solutions, analogous to classical machine learning and traditional statistical approaches, are deployed to perform one of the following tasks: (I) *anomaly detection*, *diagnosis*, and, *prognosis* [53]. The former aims to detect the machinery's current condition status and identify possible anomalies, while *diagnosis* discriminate the anomalies in order to differentiate healthy against faulty working conditions. *Prognosis*, on the other hand, involves monitoring, tracking, and predicting the machinery's degradation given its current working condition to, ultimately, estimate its *point of failure*. This task is either accomplished as a *classification* problem, by identifying a predefined set of *failure types*, or as a *regression* problem by directly predicting the *Remaining Useful Life* (RUL), typically measured in cycles or time until failure or sometimes by computing the *anomaly deviation* score or *Health Index* (HI).

In this rapidly evolving landscape, where modern industrialization techniques highly relies on technological advancements and, the symbiotic relationship between research and practical applications has never been more interlaced, we investigate the soundness of XAI methods, while discussing the current notion of *faithfulness* for the most robust family of explanations within *prognostics*: CAM[60].

The increasingly growing adoption of more modern and larger Deep Neural Network (DNN) within the industry, often by lending SoA models from contingent fields [55], has resulted in a continuously increasing number of parameters. As inference costs become progressively more expensive, the need and urge for more scalable and parallelizable XAI techniques is only expected to increase. In spite of that, in the past years have been published many new CAM-based techniques[15, 15, 30, 66, 12, 43, 18, 28, 22, 42], almost in their entirety *non-gradient* based, as *gradients* have been deemed unreliable due to their associated issues [50]. Few methods[50, 42] have been implemented to address these challenges, which augmented the original Grad-CAM formulation with provably *difference-from-reference* techniques[63, 58]. However, due to their poor performance, such augmentations have been considered inadequate within CAM, and no further work has been carried out in this direction.

1.2 RESEARCH QUESTIONS

The study discusses and addresses the following research questions:

1. **RQ1**: To what extent does the original formulation of Grad-CAM suffers from saturating and vanishing gradients and can a *gradient-based* CAM method be formulated that does not suffer from such limitations?

2. **RQ2**: Is it possible to create a pure *gradient-based* CAM technique which offers *high-resolution* and *class-discriminative* explanations without combining any other method?

1.3 CONTRIBUTION AND NOVELTY

The propositions and contributions of this work are subdivided into two segments: *(I) Towards Gradient Faithfulness* and *(II) Beyond Faithfulness*.

In the first part we theorize three novel *gradient-based* CAM formulations, aimed at replacing traditional Grad-CAM-based methods, and every formulation built upon, which addresses the vanishing and saturating gradient problems. This results in novel augmentations of Grad-CAM which alter the original gradient computation with a modified and adapted technique, derived from the proven *difference-from-reference* approach *Expected Gradients* (EG)[20], that involves a *path attribution method* of which baseline is sampled from a distribution. Because our method operates on gradient computation, rather than on the recombination and usage of the partial derivatives as weighting factors of the resulting CAM, our proposed techniques, namely Expected Grad-CAM, Expected Grad-CAM++ and Guided Expected Grad-CAM, are intended as full *inplace* replacements of Grad-CAM and any posterior work built upon it. Consequently, our work is of extremely relevance as it allows to rewrite any existing technique based Grad-CAM in terms of our method Expected Grad-CAM, providing immediate, out-of-the-box increase in quantitative performance by providing a *gradient-safe backbone*.

In the second part, instead of rewriting an existing CAM method in terms of our *gradient-safe* proposition, to produce a new variation which contest current SoA methods, we utilized our method Expected Grad-CAM to devise a completely new approach. First, we challenge the current state and formulation of *visual explanations* within the XAI field as a whole. Then we formalize a set of properties and constraints that an *informative* and *human-interpretable* saliency should respect to provide meaningful information. Ultimately, we devise a pure CAM technique, based on our prior work Expected Grad-CAM, which yield both *high-resolution* and *class discriminative* explanations without fusing other methods while addressing the issues of both *gradient* and CAM methods altogether. Our last proposition, namely Hyper Expected Grad-CAM, generates a new type of *hybrid* saliencies which follows our notion of *faithfulness* and *natural encoding* (*More in Section 4.7.4*) by leveraging two novel ideas that go in the opposite direction of prior works:

- Resolution is not just pixels. *Frequency Decomposition is all you need.*

- Saliencies are not informative nor faithful: they do not follow the *natural encoding*.

Whereas current CAM methods [28, 18, 15] aim to improve the original formulation by increasing the spatial resolution of the coarse heatmap, that is, the number of pixels, our approach operates on *frequencies*, grounded on the *natural* notion of *perceived resolution* and *atomic details* (*More in Section 4.8.5*). By rethinking the notion of *faithfulness* and *resolution* it is possible to create a new type of saliencies that are conditioned by the progressive build-up of the model's understanding at any given layer, which representation is encoded only by the *high-level* concepts which the original network understands and has learned during training and not by the arbitrary set of *high-level* features and construct present in the original image. Currently, to obtain both *fine-grained* and *class-discrimnative* explanations "*guided*"-based methods are used, which by combining multiple methods produce a relevancy masking of the input. These methods produce *unfaithful*, according to our notion, and deceitful saliencies, as the composition of the map, in the individuality of each detail, does not encode the model's understanding. In contrast, Hyper Expected Grad-CAM produces new type of *hybrid* saliencies with an *unprecedented* level of detail and clarity, where each saliency is composed of the individual make up of the uncompressed and progressively reconstructed model's understanding, conditioned and gradient-weighted. In other words, the saliency follows the *natural encoding*, where each *atomic detail* is the encoded representation of the model's understanding up to a given layer. Finally, to produce more exhaustive results and quantify the impact and extent of the gradient issues, we conducted a 5 to 10 times larger evaluation study, when compared to prior works, on the ImageNet Large Scale Visual Recognition Challenge (ILSVRC2012) dataset on the most *well-established* quantitative metrics within XAI. The evaluation covered nine of the most recent and performing CAM methods (*More in Chapter 5*) compared across six metrics. The *insertion* and *deletion* metrics have been computed with over 1000 iterations, rounding up to 5 times larger than in previous studies, allowing for more accurate and localized evaluations, totaling an astonishing 293 hours of evaluation time on an *A100* cluster (*More in Chapter 5*).

1.4 CONSTRAINTS AND LIMITATIONS

The most significant limitation of this paper is represented by the employment, or lack of, definitive quantitative evaluation metrics. This issue is not unique to this paper, but rather longitudinal to all papers that propose visualization approaches within XAI. All prior works in the field [71, 57, 74, 9, 52, 50, 58, 42] offer some level of *qualitative* assessment by inspecting the discriminative regions and judging the explanation's performance upon which method better highlights

the object class. This class of evaluations, however, is inherently unreliable, unrepeatable, and intrinsically biased by the observer's assumptions on how the sensitivity maps should look like *w.r.t.* to each class. This often compounds with additional biases and assumptions towards the model's inner workings and which region or feature the model should focus when making a prediction. To further reinforce this notion, prior works have been produced showing the unreliability [32], infidelity [69], and shortcoming of qualitative assessment [38] for human-interpretability of explanations [41]. On the contrary, in recent years, many quantitative strategies have been developed with the most common being based on the notion of *ablation* [27]. These techniques are grounded on the simple notion that if a feature has great importance towards a prediction, then removing such input should greatly affect the model's performances [27]; this class of *ablation* approaches presents a crucial limitation: progressively removing features forms inputs that do not belong from the same distribution of the train set, violating one key assumption in machine learning [27]. In addition Hooker et al. argue that without retraining the model, due to this induced covariate shift, it is unclear whether the degradation in performance is a consequence of the distribution shift or because the ablated features were actually important [27]. Alternative metrics in more recent years have been proposed [68, 68] to try to address the aforementioned limitations, but all seem to come with their own drawbacks, with no definitive solution.

1.5 RESULTS EVALUATION

As elucidated in the preceding section, currently there are no definitive and established quantitative metrics, therefore the findings will be presented and evaluated using methodologies and strategies equivalent and compatible to previous works. In this respect, qualitative visualization will be presented alongside quantitative metrics. In connection with prior contributions [52, 12, 58], we adopted the following metrics: *(I) average drop in confidence*, *(II) average drop in activation score*, *(III) percentage increase in confidence*, *(IV) percentage increase in activation score* and *(V) increase ratio*. Each metric is broadly described in Chapter 5, in conjuction with their implementation.

2

LITERATURE REVIEW

This chapter contains a comprehensive survey of existing methods within the scientific field and their relationship to our propositions.

2.1 EXPLAINABLE AI

Notwithstanding the recent surge of interest in *Explainable AI* (XAI) techniques, attributed to the accelerating shift towards a more algorithmic society [4], the notion of *interpretability* is not novel, van Lent et al. discussed the need, and devised an implementation, of providing explanations for complex military simulation training systems [64]. Doshi-Velez and Kim, in light of the emerging interest and proliferation of publications in the field, provided a formal formulation of *interpretability* and its applications [14], suggested a taxonomy for *rigorous evaluations*, aimed at measuring the *effectiveness* of interpretable machine learning systems, and, exposed open questions towards the development of more reliable XAI methods [14]. The constant increase in *neural networks* performances, experienced in recent years, and its renew inverse relationship with *explainability*, prompted more authors to discuss and highlight the importance of interpretable models and the lack thereof, within different fields [49]. Gilpin et al. examined current literature in an effort to dissect and identify open challenges, debating, and evaluating the current state of XAI, and the availability of explanatory methods, as inadequate, especially for DNN [24]. Adadi and Berrada [4] iterated upon the paradigm of complex *neural network* as *blackboxes*, expressing the fundamental idea, at the time, of the *trade-off* between model's performances, often in terms of accuracy, and interpretability, as two orthogonal properties. Mohseni et al. [40] present a survey and a framework of XAI designs, guidelines, and evaluation methods across multiple disciplines. Barredo Arrieta et al. offered a more recent and comprehensive taxonomy of XAI methods, with a focus on *sub-symbolism* or DNN [6], while discussing and advocating for *responsible AI* and fairness in *machine learning*. Other works have been proposed [26, 11, 65, 36, 46] that discussed the *trade-off* between performance and explainability, while providing a survey of methods and challenges till recent days. Du et al. evaluated relevant XAI methods and offered a detailed analysis and definition of *globally* and *locally* interpretable models and explanations, *post-hoc* explanations, and differentiated *model-agnostic* against *model-specific* explanations.

2.2 PREDICTIVE MAINTENANCE

Prognosis and maintenance optimizations are not novel concepts, as the industry has always been seeking new techniques to maximize high utilization of mechanization, and developed technique to keep track of the health of the machinery [17]. Selcuk [51] offered a comprehensive overview of different *maintenance policies*, highlighting the importance and differences between *corrective maintenance*, *preventive maintenance* and *Predictive Maintenance* (PdM), while providing current implementations and latest trends across industries. Lee et al. [33] draws the attention to the ongoing evolution of modern products, where with the rapid rise in computational power and data availability, render maintenance no longer an aftermarket service but a core essential functionality of the product development. Simultaneously, early works [39] remarks the importance of PdM in *continuous condition monitoring* of machinery with computer-based techniques [39]. With the advent of I4.0, Bousdekis et al. [7] discusses the available techniques in the context of *smart manufacturing*, emphasizing on the benefits of the new technological advances. Similarly, Poor et al. [44], Zonta et al. [76] and Achouch et al. [3] explore the opportunities within the PdM field connected with the advancements brought by I4.0. In 2019 Silvestrin et al. [55] provided a comparative study of SoA *machine learning models* employed within the PdM such as *DT*, *KNN*, *TCN* and *LSTM*, showing how classical ML models outperform more complex networks when data is scarce. Conversely, more recent study by Serradilla et al. [53] provided a more elaborate analysis and broader comparison of more modern SoA *deep learning* architectures categorized by industrial applications, their results are compared. Towards the RUL estimation multiple methods have been proposed including *DCNN* [5], *CNN+ FFNN* [35], *LSTM* [73], *LSTM* with attentional interface [31]. More recently, more advanced *hybrid approaches*, as combination of the previous methods, have been proposed with current SoA performances, including *CNN-LSTM* [1], *CNN-BiLSTM* [70] and *multibranch CNN-BiLSTM* [8].

2.3 EXPLANATIONS

Gradient-based XAI methods operate on differentiable models and, using the network's gradients, obtained in one or multiple forward and backward passes, capture the relationship between the input and the output, allowing to identify the relevancy of each feature and their influence towards a given prediction. One of the most significant paper, that established the foundation for all subsequent publications, was proposed by Zeiler and Fergus [71] that introduced a novel visualization technique based on the *Deconvolution network* (DeconvNet), that allowed to inspect convolutional feature maps. Notwithstanding,

the groundbreaking success of this work, this technique was limited to Convolutional Neural Network (CNN) models. In 2013 Simonyan et al. [57] addressed the limitation of DeconvNet and extracted a generalized approach based on the *backpropagation* of the gradients i. e., computing the gradients of the target output with respect to the input, allowing to visualize a neuron in arbitrary layer. Simonyan et al. also discussed the concept of *saliency* or *sensitivity* maps as a visualization techniques to outline important regions or patches of an image used to make predictions. By combining the previous techniques, Springenberg et al. [61] proposed a technique that combine the advantages of both methods, *guided backpropagation*. In addition, it formulated an *all convolutional network* [61], in which the pooling layer were replace by *convolutional layers*, and ultimately, showing the effectiveness of *guided backpropagation* without the use of the *switches*, employed in the DeconvNet. As an attempt to tackle the issue related to the vanishing gradients which lead to incorrect visualization, Zhou et al. propose *Class Activation Map* (CAM) [75] which involved the use of a *Global Average Pooling* (GAP) layer to produce *discriminative* image regions. CAM required, most often, to alter the networks architecture as most model did not have a GAP layer at the end, but rather a dense layer, which required fine tuning and small retraining to apply this technique. These limitations were addressed in Grad-CAM [52], where Selvaraju et al. provided a generalized version of CAM which did not require to alter the model's architecture. Similar to previous work [74], the map could be generate with a single forward pass and a more efficient backward pass which would stop at the last convolution layer, or the one under analysis. Selvaraju et al. also devised an approach that combined Grad-CAM and *backpropagation* to produce *class discriminative high-resolutions* maps i. e., *guided* Grad-CAM [74]. In 2017, building on previous work [52], Chattopadhay et al. presented *Grad-CAM++* [9, 10]. In Chattopadhay et al. work, this concept is described as a *generalization* of Grad-CAM[52], which provides better visualization using solely the positive partial derivatives of the last convolutional layer. In this direction, Omeiza et al. proposed an enhanced version of *Grad-CAM++* i. e., *Smooth Grad-CAM++*[42] based on Smilkov et al. work, *SmoothGrad* [58]. In Recent publication [34], Lerma and Lucas questions Omeiza et al. approach and work, point out problems in their proposed method arguing that *Grad-CAM++* is equivalent to the original work [52] with positive gradients. All the proposed *gradient-based* techniques produce incorrect visualization whenever the gradients are 0 due to the discountinuities in the space, therefore Sundararajan et al. propose an *axiomatic* attribution approach i. e., IG [63] which provide feature importance with respect to a *baseline*. Smilkov et al. enhanced IG by adding noise to similar images and then taking the average of the resulting sensitivity maps, presenting *SmoothGrad* [58]. Recently, Erion et al. introduced a new method that aim to outper-

form existing attribution methods, i. e., *Expected Gradients* [20] where batch training procedure is used to approximate the expected gradients by regularizing using the entire training set as a reference. Less popular *gradient-based* methods have also been proposed as a recombination of the latter, such as *FullGrad* [62], *Score-CAM* [66] or *Integrated Grad-CAM* [50].

3

FOUNDATIONS

The following section contains a comprehensive and detailed overview of the mathematical and theoretical foundations that underpin our proposed approach, encompassing the fundamental concepts that characterize established visualization methods within the XAI field. In each section is presented a relevant component with respect to our proposition, establishing a robust theoretical basis. In addition, the sub-sections are organized sequentially in a manner that corresponds to the order in which they build upon one another, reflecting the conceptual progression of their respective propositions, and in accordance with the degree of the contribution they provide to our approach.

3.1 DECONVNET

The following subsection discusses the theoretical notions and implications of *Deconvolution network* (DeconvNet)[71] focusing solely on its visualization novelty (Section 3.1.1) and the ablation study (Section 3.2) within the context of our proposition, therefore, focusing on the aspects that directly influenced the implementation details and the derivation of the intuition discussed in more detail in Chapter 4.

3.1.1 *Visualization*

DeconvNet proposes an approach that allows to interpret the features activity in intermediate layers [71], by mapping these *activities* back to the original input space. This, concretely, shows which input pattern caused a specific given activation in the feature maps [71]; this mapping is performed using a *Deconvolution network* (DeconvNet), hence the name, allowing to map the *filtering* and *pooling* operations back to the input, effectively reversing the convolution process. This showed to be of particular interest as the *feature maps* of the last convolution layer, retains the higher level *features* and patterns extracted, which are often indicated of utmost interest and importance, in terms of *interpretability*, towards the prediction. These maps, however, contain increasingly more compressed latent feature and pattern detectors in relation to the relative depth of the network due to the application of the *pooling operations* during the forward pass. In spite of that, due to the nature of the *convolution operation*, which as direct effect, progressively increase the *receptive field* of subsequent layers, provides the maps, and therefore its object detector, to relatively align, dimension-

ality aside, with the input. Moreover, because on the backpropagation step, the *unpooled* approximation is constructed using the switches (locations) computed during the forward pass, and these are aligned with the input image, consequently the reconstruction obtained will also align, representing a sub-region (in compressed space) of the original input and weighted in accordance to its contribution to the feature map activation [71]. In this context, the DeconvNet is used to reverse the *convolution process*, by providing an approximation of it, and *upscale* the subsampled representation such that they align with the original input. In this sense, the DeconvNet, to provide visualization means, is coupled to each of the respective *Convolution network* (ConvNet) layers as depicted in *Figure 1*. The *convolution process* can be reversed by applying its original transformations in reverse orderer, namely (I) *unpool*, (II) *rectify* and (III) *filtering* [71]. The *pooling operation*, cannot be inversed directly as its an *non-invertible* operation [71], therefore its computed an approximation of it, by recording the locations, in terms of indexes, of the *maxima* within each region during the forward step and storing them in a set called *switches* [71]; during the *unpooling operation* these values are copied and placed in the correct location, according to the switches, to provide an approximation of the reconstruction such that the activation can be plotted, and consequently align, in the original image input space. The *unpooled maps* are then *rectified* to ensure that the feature reconstructions are also positive and retains the same non-linearity characteristic of the forward pass. Ultimately, the *rectified unpooled maps* are filtered using a *transpose* version of the filters involved in the ConvNet section; this is particularly important, as conversely to previous works [72], the DeconvNet is not used in any *learning capacity* [71], but solely to visualize, thence it re-utilizes the same convolution filters of the ConvNet side (no training involved), and, its inversion is performed by symmetrically flipping each filter across the horizontal and vertical axis. Remarkably, because the original work [71] utilized a model trained for classification (i. e.*AlexNet*) for the ConvNet side, the obtained reconstructions, on the DeconvNet side, are also discriminative.

3.2 SALIENCY MAPS

Saliency Maps belong to a type of XAI techniques that allows, given an input, to identify the most influencial features with respect to a particular prediction. Despite *saliency maps* are independent from the previous approach, and therefore can be generated using a multitude of techniques, this section is centered around the work [57], which proposes a *gradient-based* generalization of the DeconvNet, untying the previous work from *ReLU-based* CNN. The logits of a model f, for a given class c (S_c, where $S_c \rightarrow I$ is a highly non-linear function), given an input image I, can be approximated by first-order Taylor

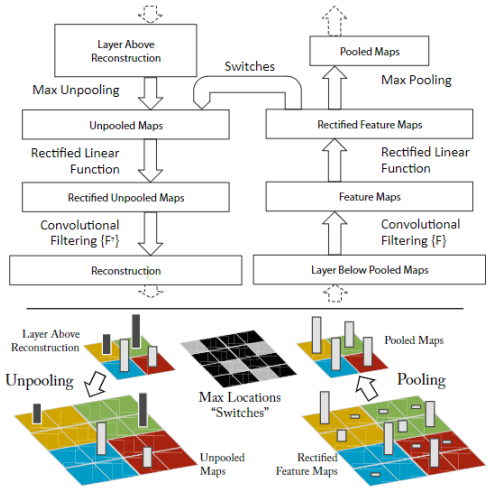


Figure 1: *Deconvolution network* (DeconvNet) - *Convolution network* (ConvNet) coupling. Adapted from [71]

expansion [57], given by Equation 10, where the w is the derivative the importance score *w.r.t* the input image at point I_0 (Equation 11). The magnitude of the elements w, given a class c directly represent the importance of that input when predicting class c. The result is a $3D$ tensor, given a $2D$ image with k channels, ($w_c \in \mathbb{R}^{H \times W \times K}$), to be able to extract a single class saliency the maximum magnitude of w_c along all the channels is taken (Equation 12). Building from the work proposed in the previous section, the connection between *gradient-based* visualization and DeconvNet is that the reconstruction of an input layer X_n is equivalent to the gradient activity f when computed with respect to the input X_n [57]. More strictly, by retracing the steps in the previous section, layer by layer, we can observe that X_{n+1} is obtained:

1. for **convolution layer**:

$$X_{n+1} = X_n * K_n \rightarrow \text{ convolution} \tag{1}$$

$$\frac{\partial f}{\partial X_n} = \frac{\partial f}{\partial X_{n+1}} * \widehat{K}_n, \quad \text{where } \widehat{K}_n = F^T \tag{2}$$

$$\Rightarrow R_n = R_{n+1} \star \widehat{K_n} \tag{3}$$

2. for **activation layer**:

$$X_{n+1} = \max(X_n, 0) \rightarrow \text{ReLU} \tag{4}$$

$$\frac{\partial f}{\partial X_n} = \frac{\partial f}{\partial X_{n+1}} \mathbb{1}(X_n > 0) \tag{5}$$

$$\Rightarrow R_n = R_{n+1} \mathbb{1}(R_{n+1} > 0) \tag{6}$$

$$R_n \neq \frac{\partial f}{\partial X_n} \tag{7}$$

3. for **max-pool layer**:

$$X_{n+1}(p) = \max_{q \in \Omega(p)} X_n(q) \rightarrow \text{ maxpooling} \tag{8}$$

$$\frac{\partial f}{\partial X_n(s)} = \frac{\partial f}{\partial X_{n+1}(p)} \mathbb{1}\left(s = \arg\max_{q \in \Omega(p)} X_n(q)\right) \tag{9}$$

By inspecting Equation 1 in light of Equation 2, is clear that the gradient of f, the activity output, w.r.t to the n^{th} layer input X_n, by applying the chain rule, is provided by the derivative of the input of the next layer, convolved by the inverse of the original filter K_n (Equation 2; the inverted filter $\widehat{K}_n$, represents the transposed filter F_T in the DeconvNet (Figure 1), which is a symmetrically flipped version of K_n, in both horizontal and vertical axis. Therefore, the convolution with the inverted filter K_n (Equation 3) represents the same operation carried out in the DeconvNet when reconstructing the n^{th} layer R_n.

Similarly, during the computation of the derivative of the *maxpooling operation*, where the maximum value of the feature maps within the neighborhood $\Omega(p)$ of the input is computed (Equation 8), the locations of the *max* pooled values are stored in s, similar to the computation carried out in the DeconvNet when saving the *switches*.

The only difference is present in the computation of the sub-gradient for the activation function ReLU, where on the DeconvNet the *indicator* function (Equation 13) is computed on the reconstructed output R_{n+1}, while in the gradient approach is computed directly on the layer input x_n. For this reasons, the reconstruction R_n used in a DeconvNet is equivalent to the computation of the derivative $\partial f / \partial X_n$ using back-propagation [57], making the *gradient-based* visualization a generalization of the work [71] discussed in the previous section. This provides a more general method to compute the contributions to any layer, i. e.to compute its saliency map.

$$S_c(I) \approx w_c^T I + b_c \qquad \forall I \in N(I_0), \qquad \textit{where} \quad \begin{matrix} I_0 := input \\ c := class \end{matrix} \tag{10}$$

$$w_c = \left.\frac{\partial S_c(I)}{\partial I}\right|_{I=I_0} \tag{11}$$

$$M_{ij} = \max_k \left| w_{ijk}^c \right| \qquad \textit{where } I_0 \in \mathbb{R}^{H \times W \times K}, \quad w_c \in \mathbb{R}^{H \times W \times K} \tag{12}$$

$$\mathbb{1}_A(x) = \begin{cases} 1 & \text{if } x \in A \\ 0 & \text{if } x \notin A \end{cases} \tag{13}$$

3.2.1 *Notation Remark*

The mathematical notation proposed in the previous section has been adapted to fit the original work from the respective authors and common notation, therefore whereas the *convolution* operator ($*$) is present, it is used in place of the *cross-correlation* operator, hence the associativity property in the aforementioned cases does not apply. Equation 14 and Equation 15 shows a discretized implementation of *cross-correlation* and *convolution* respectively, on a $2D$ input for a kernel K.

$$f_{i,j} = K \otimes G = \sum_{u=-k}^{k} \sum_{v=-k}^{k} K_{u,v} G_{i+u,j+v} \tag{14}$$

$$f_{i,j} = K * G = \sum_{u=-k}^{k} \sum_{v=-k}^{k} K_{u,v} G_{i-u,j-v} \tag{15}$$

3.2.2 *Class Saliency*

Notably the authors Simonyan et al. proposed a technique to visualize what the model is dreaming when thinking of a specific class i. e.to visualize the models learned for a specific class. This is accomplished by numerically generating an image using prior work [19] that maximizes (Equation 16) the output logits with respect to a specific class [57]. This is accomplished by utilizing gradient ascent to iteratively modify an input image to maximize the logits output [72]. The best result have been found using prior scores (before *softmax*) to produce a smooth image using L_2 regularization and an excert from the original paper has been shown in Figure 2.

$$\arg\max_{I} S_c(I) - \lambda \|I\|_2^2, \qquad \textit{where} \quad \begin{matrix} I := input \\ c := class \end{matrix} \tag{16}$$

3.3 GUIDED BACKPROPAGATION

Guided backpropagation [61] builds upon the existing techniques discussed in the previous sections, namely DeconvNet[71] and *saliency maps*[57] and provides more effective and sharper visualizations by incorporating additional guidance signals from higher layers. Springenberg et al. formulated a CNN in which all the *max-pooling* operations were replaced by a *convolutional* layer. Moreover, given some feature maps $f \in \mathbb{R}^{H \times W \times N}$, the subsampling with pooling size k, as a replacement of the original *pooling operation*, is obtained taking the *p-norm* of the sum of the absolute values of the feature map elements

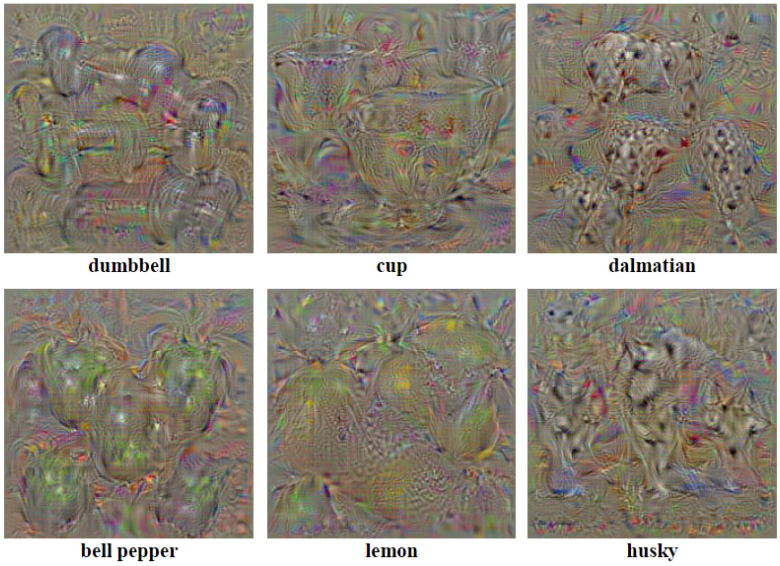


Figure 2: Numerically computed images, illustrating the class appearance models, learnt by ConvNet. Adapted from [57]

within the local neighborhood of size k (Equation 17). When p is chosen carefully i. e.using the *infinity norm* ($p \to \infty$), then the *pooling* becomes *maxpooling*. The neighborhood with stride r is computed according to Equation 18, where the indices i, j represents the spatial coordinate, and h, w defines the window size in the horizontal and vertical direction respectively. Addition the *convolution operation* is defined in Equation 19 where f_g is the same mapping as before, while $\theta_{h,w,u,o}$ are the weights, or more specifically the filter's parameters which are constructed as a $4D$ tensor with a width, height, input and output channel representing the variables h, w, u, o respectively. The activation function σ is often chosen as Rectified Linear Activation Function (ReLU). Notably, the *pooling* operation can interpreted as a *feature-wise* or *depth-wise* convolution in which the traditional activation is replaced by the *p-norm* [61]. Furthermore, when the number of input channel u is equal to the output channel o, then the $\theta_{h,w,u,o} = 1$; more interestingly in any other case the $\theta = 0$ turning the tensor from $4 - dimensional$ to $3 - dimensional$ as it is sliced *depth-wise*. By inspecting the Figure 3 is clear that the *pooling* layer can be replace with similar model's performances with a *convolutional* layer with similar stride. Note that in the *All-CNN-C* the *convolution layer*, when performing subsampling, it replaces the *pooling* and the *convolution* layer below.

$$s_{i,j,u}(f) = \left(\sum_{h=-\lfloor k/2 \rfloor}^{\lfloor k/2 \rfloor} \sum_{w=-\lfloor k/2 \rfloor}^{\lfloor k/2 \rfloor} \left| f_{g(h,w,i,j,u)} \right|^p \right)^{1/p} \quad where \quad f \in \mathbb{R}^{H \times W \times N} \tag{17}$$

Model		
Strided-CNN-C	ConvPool-CNN-C	All-CNN-C
Input 32 × 32 RGB image		
3 × 3 conv. 96 ReLU 3 × 3 conv. 96 ReLU with stride $r = 2$	3 × 3 conv. 96 ReLU 3 × 3 conv. 96 ReLU 3 × 3 conv. 96 ReLU	3 × 3 conv. 96 ReLU 3 × 3 conv. 96 ReLU
	3 × 3 max-pooling stride 2	3 × 3 conv. 96 ReLU with stride $r = 2$
3 × 3 conv. 192 ReLU 3 × 3 conv. 192 ReLU with stride $r = 2$	3 × 3 conv. 192 ReLU 3 × 3 conv. 192 ReLU 3 × 3 conv. 192 ReLU	3 × 3 conv. 192 ReLU 3 × 3 conv. 192 ReLU
	3 × 3 max-pooling stride 2	3 × 3 conv. 192 ReLU with stride $r = 2$
⋮		

Figure 3: Performance comparison of traditional ConvPool-CNN and All-CNN. Adapted from [61]

$$g(h, w, i, j, u) = (r \cdot i + h, r \cdot j + w, u) \tag{18}$$

$$c_{i,j,o}(f) = \sigma\left(\sum_{h=-\lfloor k/2 \rfloor}^{\lfloor k/2 \rfloor} \sum_{w=-\lfloor k/2 \rfloor}^{\lfloor k/2 \rfloor} \sum_{u=1}^{N} \theta_{h,w,u,o} \cdot f_{g(h,w,i,j,u)}\right) \tag{19}$$

3.3.1 *Conditioning*

As previously discussed in Section 3.1 , the DeconvNet[71] produces a reconstruction R_n, where, starting from a high-level feature map, reverses the typical forward flow of a ConvNet from neurons activations to the input image, showing the image patch, in the original input space, which most strongly activated the neuron [61]. Since the *max-pooling* layers are *non-invertible*, Zeiler and Fergus propose a technique that involves the use of *switches* to store the positions of the maxima during the forward pass with respect to each *pooling* region. The usage of the *switches*, however, since they are computed during the forward pass, creates visualizations that are *conditioned* on the input image, not allowing to visualize the learned features correctly [61]. This is particularly relevant for higher layers, where the DeconvNet fails to produce sharp [61] visualizations. This is due to the fact that higher layers learn more invariant representations, and those do not have a single image that can maximally activate those neurons; lower-level features, in contrast, learn more generic features and therefore have less capacity to invariance [61]. On the other hand, as discussed in Section 3.2, neuron's activation can also be computed, by generalizing the DeconvNet approach, therefore computing the gradient of activity with respect to the input image. This approach is inheretly *conditioned* on the image by design, first by the activation functions and then through the *switches* [61]. In this context, to produce a reconstruction that is *conditioned* on the input but

that does not rely on the *pooling* layers, and therefore *switches*, Springenberg et al. propose techniques that combine the two approaches, DeconvNet and *backpropagation*, namely *guided backpropagation* [61]. In this sense, re-iterating what discussed in Section 3.2, DeconvNet can be generalized to a backward pass (Equation 2, Equation 9), except when propagating through a non-linear activation (Equation 7), as its gradient is only computed based on the top gradient signal (Equation 22) [61]. The vanilla backpropagation, on the other hand, only relies on the *bottom signal* from the input (Equation 21). *Guided backpropagation* (Equation 23) combines these two methods [61, 71, 57], i. e., instead of zeroing negative values flowing from the top signal (DeconvNet) or the bottom (*backpropagation*), all values for which at least one of these is negative are masked out, effectively preventing the backward flow of negative gradients, which otherwise would decrease the overall activation of the neurons under analysis present in higher layers[61].

$$f_i^{l+1} = \text{relu}\left(f_i^l\right) = \max\left(f_i^l, 0\right) \tag{20}$$

$$R_i^l = \left(f_i^l > 0\right) \cdot R_i^{l+1} \qquad \textit{where} \qquad R_i^{l+1} = \frac{\partial f^{\text{out}}}{\partial f_i^{l+1}} \tag{21}$$

$$R_i^l = \left(R_i^{l+1} > 0\right) \cdot R_i^{l+1} \tag{22}$$

$$R_i^l = \left(f_1^l > 0\right) \cdot \left(R_i^{l+1} > 0\right) \cdot R_i^{l+1} \tag{23}$$

3.4 CAM: CLASS ACTIVATION MAP

Class Activation Map (CAM), similarly to the methods discussed previously, produces a *discriminative* visualization of the regions of interest with respect to a specific category (Figure 5) [74] . The method relies on the usage of a *Global Average Pooling* (GAP) layer, that is placed subsequently to the *convolutional* layer to visualize, that is to retain *spatial* and *localization* information. Since the methods operate on the last *convolutional* layer it is generally faster as it does not require to backpropagate all the way to the input, as previous works [57, 61]. More formally, given a set of feature maps which are placed in the last *convolutional* layer, where $f_k(x, y)$ is the activation of neuron k at spatial index x, y, let the GAP defined as Equation 24, then the logits of the model, therefore before being fed to *softmax* (Equation 27), are a linear combination of the weights for class c given neuron k and GAP (Equation 25). In this case the weights w_k^c, directly indicates the importance of F_k for class c [61]. Provided that GAP produces linearity, by plugging Equation 24 in Equation 25, is possible to redistribute F_k and obtain the CAM for class c, in other words, S_c represents the importance of the activations, as linear combinations, at a specific spatial locations x, y for a specific class c. Notably, Zhou et al. outlined the usage of the GAP layer not just as *structural regularizer*, as pointed

out by previous work [37], but showed, in addition, the advantages of GAP to detect the extent of an object rather than solely focusing on the identification of one discriminate part, as previous obtained by using *Global Max Pooling* (GMP) [75]. It is noteworthy to mention, as it represents a core limitation of the approach, and building block of later approaches, that the highly reliability of the aforementioned approach on the GAP layer, the following method only works whereas a GAP layer is present in the first place, and therefore requires additional restructuring in those model's architecture that do not provide it, necessitating supplementary training. Moreover, since the *feature maps* at the last *convolutional layer*, as previously discussed in Section 3.1, contain compressed high-level representation, implying that they are in a lower resolution compared to input, due to the subsequent subsampling applied by the prior *pooling layers*. This therefore requires, the application of an *upsampling* operation, applied either at the *feature maps* level or at the CAM level, to upresolve the maps to the original input space. For *2-dimensional* images, a common interpolation used is the *bilinear interpolation* i. e., applying *linear interpolation* iteratively in each direction. Figure 4 shows an example of CAM, displaying artifacts (jaggered edges) due to the *upsampling* operation.

$$F_k = \frac{1}{N} \sum_{x,y} f_k(x,y) \tag{24}$$

$$S_c = \sum_k w_k^c F_k \tag{25}$$

$$\begin{aligned} S_c &= \sum_k w_k^c \sum_{x,y} f_k(x,y) \\ &= \sum_{x,y} \sum_k w_k^c f_k(x,y) \\ \Rightarrow M_c(x,y) &= \sum_k w_k^c f_k(x,y) \end{aligned} \tag{26}$$

$$\text{softmax}(\boldsymbol{X}) = \frac{1}{\sum_{i=1}^{n} e^{x_i}} \cdot \begin{bmatrix} e^{x_1} \\ e^{x_2} \\ \vdots \\ e^{x_n} \end{bmatrix} \tag{27}$$

3.5 GRADCAM

To solve the drawback discussed in the previous section, Selvaraju et al. devised a method, namely *Gradient-weighted Class Activation*

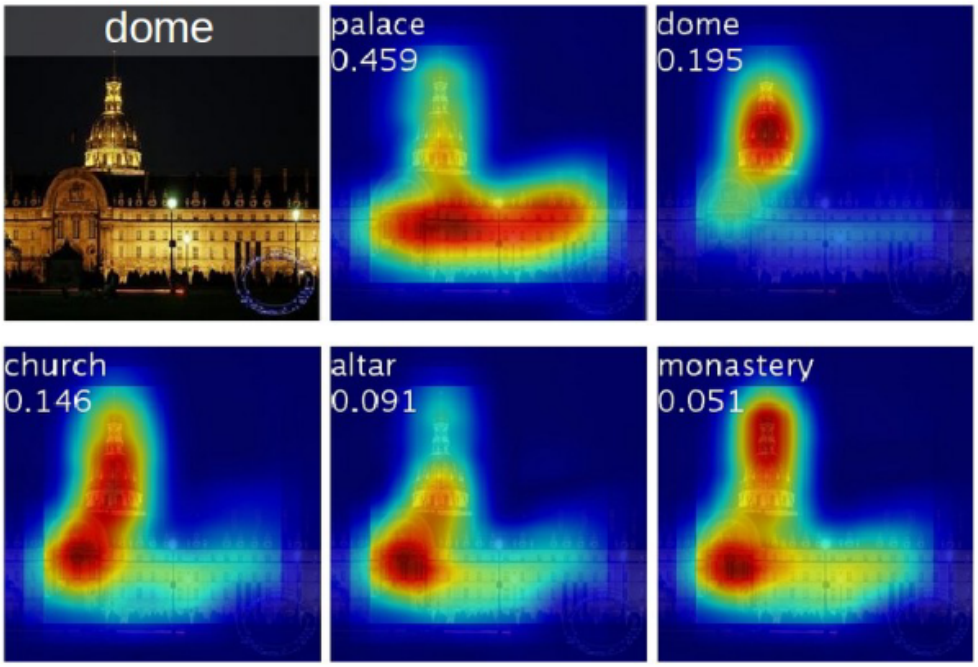


Figure 4: Examples of the CAMs generated from the top 5 predicted categories for the given image with ground-truth as dome. Adapted from [75]

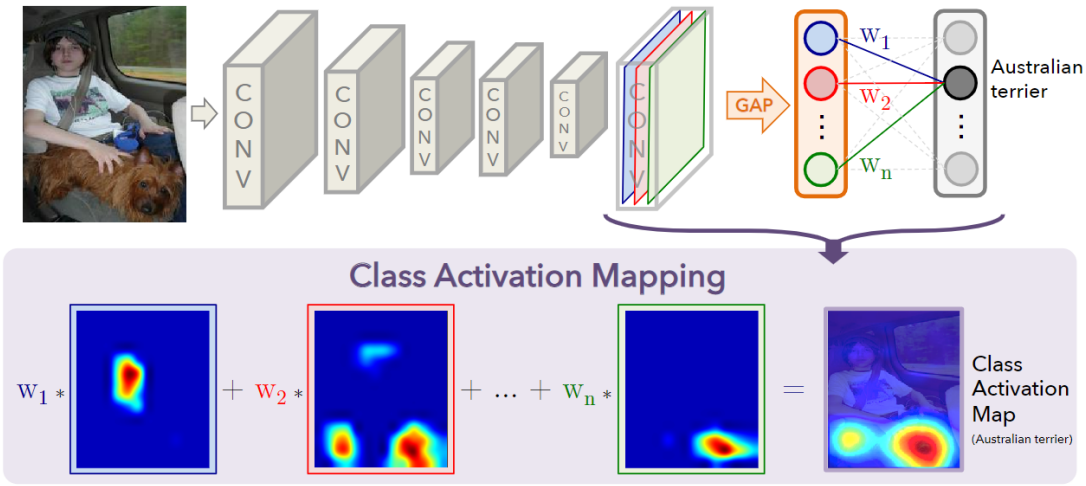


Figure 5: *Class Activation Map* (CAM) Architecture overview. Adapted from [75]

Mapping (Grad-CAM) (Figure 6), that produces *highly discriminate* coarse heat map in its normal variant, or *high-resolution* and *highly discriminate* fine-grained visualization, when combined with *guided backpropagation*, namely *Guided* Grad-CAM [52]. In either implementations, this approach eliminates the need of a gap layer subsequent the *convolutional* layer to be analyzed [52], permitting this approach to be applied to a wide variety of CNN-based models eliminating effectively the need for extra training or fine tuning derivated to the model's architecture change necessary in previous work [74]. Formally, the *class discriminative* coarse localization map $L^c_{\text{Grad-CAM}}$, is obtained computing the weighted combination of the *globally averaged pooled* feature activation maps followed by a ReLU function[52]. Specifically, first the importance weight k for class c i. e., w^c_k are computed by back-propagating the logits y^c with respect to the *feature map* A^k, and then passed through a GAP to produce the aforementioned scalers w^c_k (Equation 28). A linear combination of the importance weights w^c_k and the feature map A^k is computed and rectified (Equation 29). Notably, the last rectification is applied in order to visualize only the positive contribution, i. e., the feature that positively influence the prediction for a given class c or increase the probabilities for that prediction [52]. Conversely, negative gradients can be thought as the contributions, in the classification task, that increase the log probabilities of other classes that are not c, which in this case are clearly not desirable as the visualization would highlight regions that contribute to different classes [52]. It is also noteworthy that the logits y^c don't have to be a class score produced by an image classification CNN, but can be any differentiable activation [52]. Finally, recalling the results discussed in the previous section, it is clear that Grad-CAM is a generalization of CAM, whereas CAM is a specialized case of Grad-CAM composed of an architecture that contains a GAP layer prior the output layer [52].

$$w^c_k = \frac{1}{Z}\sum_i\sum_j \frac{\partial y^c}{\partial A^k_{ij}} \tag{28}$$

$$L^c_{\text{Grad-CAM}} = \text{ReLU}\left(\sum_k w^c_k A^k\right) \quad \textit{where} \quad L^c_{\text{Grad-CAM}} \in \mathbb{R}^{u\times v} \tag{29}$$

3.6 COUNTERFACTUAL ANALYSIS

The techniques that have been discussed in the previous sections, which rely on the computation by backpropagation of gradients, suffer from a fundamental issue that can result in erroneous visualizations and improper feature attributions, that occurs when the gradients are 0. Specifically, due to the *non-linearity* introduced by ReLU activations and *max-pooling* layers, the gradients of the target *w.r.t.* the input i. e., $\frac{\partial y}{\partial x_i}$ might not exist, be 0 or vanish or become undefined

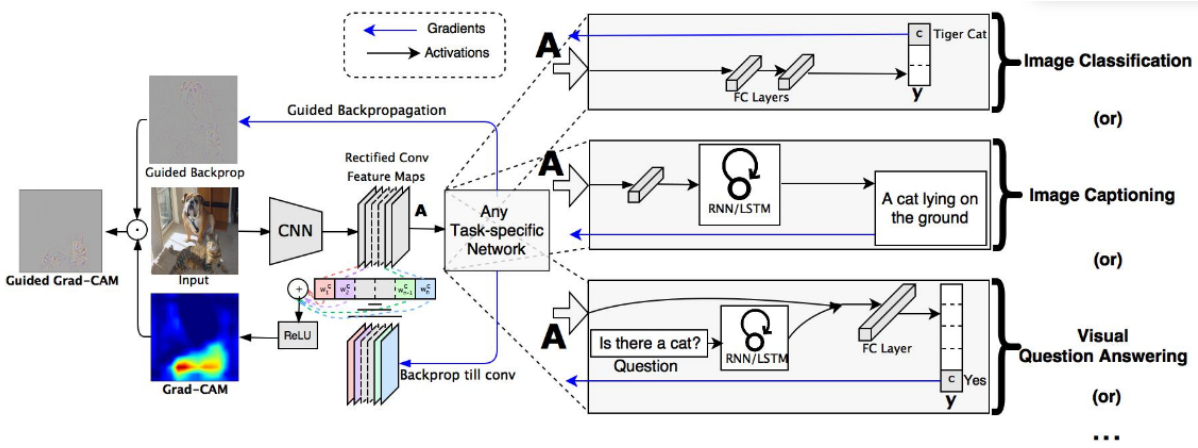


Figure 6: *Gradient-weighted Class Activation Mapping* (Grad-CAM) Architecture overview. Adapted from [52]

as the function space contains discontinuities. That is, let $C_{x,c}$ be the contribution of feature x to predict class c, generally it is a desirable property towards proper visualizations that the contribution $C_{x,c}$ is nonzero even if its derivative is 0 (Equation 31). These motivations drove the implementation of counterfactual methods and *difference-from-reference methods* such as [54] that imply the involvement of a reference during the feature importance computation, often referred as a baseline [63]. Given a DNN $F : \mathbb{R}^n \to [0, 1]$, and $x = (x_1, \ldots, x_n) \in \mathbb{R}^n$ being a set of inputs, then the attributions of those features should be a function of the model F *w.r.t.* to the input x relative to a baseline $x' \in \mathbf{R}^n$ i. e., $A_F(x, x') = (a_1, \ldots, a_n) \in \mathbb{R}^n$ [63]. Concretely, we are seeking a counterfactual explanation, that is, to assign an importance score to a feature while implicitly accounting for the case where that feature is absent, and, use this as reference in order to compare the predictions; this outlines the fundamental framework and rationale of the technique and methods discussed in the next sections such as *Integrated Gradients* (IG) [63].

$$C_{x_i,c} \neq 0 \ \rightarrow \ \text{if} \quad \frac{\partial y^c}{\partial x_i} = 0 \tag{30}$$

$$\text{Given } C_{x_i,c} \neq 0 \ \rightarrow \hat{C}_{x_i,c} \neq 0 \text{ if } \quad \frac{\partial y^c}{\partial x_i} = 0 \tag{31}$$

4

METHODS

In the subsequent sections, a rigorous and formal definition of the proposed methods, accompanied by the associated mathematical notation, is proposed. Accordingly, three novel formulations, aimed to replace traditional Grad-CAM-based methods and every other variation built on top, are introduced, namely Expected Grad-CAM, Expected Grad-CAM++, and, Guided Expected Grad-CAM. Ultimately, Expected Grad-CAM is utilized to build a novel *gradient-based* methodology that produces a new *hybrid* type of saliencies focused on a more human-centered knowledge extraction and representation of the DNN's inner workings by rethinking the current composition of visual explanations, formally Hyper Expected Grad-CAM. Ahead the sections are presented and subdivided into two main portions: *(I) Towards Gradient Faithfulness* and *(II) Beyond Faithfulness*, where in the first segment a *gradient-issue* free Grad-CAM replacement, and its associated variations, are introduced, while, in the second part, such tools are used to create a novel approach that generates a new type of saliencies.

4.1 TOWARDS GRADIENT FAITHFULNESS

4.2 INTEGRATING GRADIENTS WITH PRIORS

Recalling from the mathematical basis presented in Section 3.6, given a DNN $f : \mathbb{R}^n \to [0,1]$, and $x = (x_1, \ldots, x_n) \in \mathbb{R}^n$ being a set of inputs and x' being the baseline, the original formulation of *integrated gradients* along the i^{th} dimension is given by Equation 32 [63], i. e., ϕ_i^{IG} represents the importance score of feature i computed by accumulating the gradients from the result of the interpolation between the baseline x' and the input x. Even in the simplest formulation, the *counterfactual* notion of utilizing a *difference-from-reference* approach, represents a highly effective strategy to tackle not just the *(I) vanishing* of the gradients, often due to discountinuities within the space i. e., when the gradients *w.r.t.* input are zero $\frac{\partial y}{\partial x_i}$, but also to address the opposite problem, *(II) saturation*. In the first case, as discussed in Section 3.1 and subsequently in Section 3.2, *gradient-based* methods such as [57, 57, 61] because of the nonlinearity introduced by nonlinear activations such as ReLU, respond to stimuli, that is, backpropagate only across the path that are at least activated at the input [63]. On the latter issue i. e., *saturation* of gradients, the opposing problem is reversed: gradients at the input might exhibit small gradi-

ents despite they have highly predictive power, ultimately resulting in a distorted low importance score. Integrating over an interpolation, meaning a path, between input and baseline we can control local gradients [63]. In practice the integral in Equation 32 can be efficiently approximated using discrete summation (*Riemman* approximation), by summing gradients at sufficiently small intervals along the path from x' to the input x [63] for m steps (Equation 33). For a more efficient computation, an early stop strategy mechanic can be implemented by exploiting the *completeness* axiom (Equation 34) [63] by iteratively checking for convergence i. e., if the magnitude is smaller than an error ϵ (Equation 35).

$$\phi_i^{IG}\left(f, x, x'; \alpha\right) = \left(x_i - x_i'\right) \times \int_{\alpha=0}^{1} \frac{\partial f\left(x' + \alpha\left(x - x'\right)\right)}{\partial x_i} d\alpha \tag{32}$$

$$\phi_i^{IG}\left(f, x, x'\right) = \left(x_i - x_i'\right) \times \sum_{k=1}^{m} \frac{\partial f\left(x' + \frac{k}{m} \times \left(x - x'\right)\right)}{\partial x_i} \times \frac{1}{m} \tag{33}$$

$$\sum_i \phi_i^{IG}\left(f, x, x'\right) = f(x) - f\left(x'\right) \tag{34}$$

$$\left| \sum_i \phi_i^{IG}\left(f, x, x'\right) - \left(f(x) - f\left(x'\right)\right) \right| < \epsilon \rightarrow \text{stop} \tag{35}$$

4.2.1 *Distribution sampling*

More generally, instead of using a *constant baseline* e. g., vector of 0*s* [63], it is possible to conceptualize a *super* baseline such that it is composed by the average of multiple baselines, each sampled from a specific distribution $\mathbb{D}$, i. e., we compute an additional integration across all the $x' \in \mathbb{D}$ with weights corresponding to the probability density function of $\mathbb{D}$ for a continuous random variable (Equation 36). By substituting Equation 32 into Equation 36, it can be observed that now we are computing two integrals, and moved the *hyperparameter* choice from the *baseline determination* to a *distribution determination* (Equation 37). Equation 38 shows the implementation of the discretized version, implemented similarly as discussed in the previous section, where the baseline x' and the interpolation factor α are sampled each from their respective distribution, where α^j is the j^{th} draw.

$$\phi_i(f, x; \mathbb{D}) = \int_{x'} (\phi_i^{IG}\left(f, x, x'\right) \times p_D\left(x'\right) dx') \tag{36}$$

$$\phi_i(f, x; \mathbb{D}) = \int_{x'} (\left(\left(x_i - x_i'\right) \times \int_{\alpha=0}^{1} \frac{\partial f\left(x' + \alpha\left(x - x'\right)\right)}{\partial x_i} d\alpha \right) \times p_D\left(x'\right) dx') \tag{37}$$

$$\hat{\phi}_i(f,x;\mathbb{D}) = \frac{1}{k}\sum_{j=1}^{k}\left(x_i - x_i'^j\right) \times \frac{\partial f\left(x'^j + \alpha^j\left(x - x'^j\right)\right)}{\partial x_i} \tag{38}$$

4.2.2 *Gaussian distribution sampling*

When carefully selecting a distribution i. e., the normal *gaussian distribution* such that sampling $\epsilon_\sigma \sim \mathcal{N}\left(0, \sigma^2 X\right)$ and plugging into Equation 38, we obtain Equation 39. By close inspection it is clear that it is similar to the proposition *SmoothGrad* [58] differing only for *(I) the j^{th} sampled term is multiplied by the input* and *(II) the interpolation occurs only at the beginning of the interpolated path.*

$$\hat{\phi}_i\left(f,x;\mathcal{N}\left(x,\sigma^2 X\right)\right) = \frac{1}{k}\sum_{j=1}^{k}\epsilon_\sigma^j \times \frac{\partial f\left(x + \left(1-\alpha^j\right)\epsilon_\sigma^j\right)}{\partial x_i} \tag{39}$$

$$p_D(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{40}$$

$$\phi_i^{SG}\left(f,x;N\left(\overline{0},\sigma^2 I\right)\right) = \frac{1}{k}\sum_{j=1}^{k}\left(x + \epsilon_\sigma^j\right) \times \frac{\partial f\left(x+\epsilon_\sigma^j\right)}{\partial x_i} \tag{41}$$

4.2.3 *Expected gradients*

Building from Section 4.2.1, is possible to collapse the multiple integrations from Equation 37, using the method proposed by Erion et al., by considering both integrals as expectations, that is $\mathbb{E}_{x'\sim\mathbb{D},\alpha\sim U(0,1)}$ i. e., *Expected gradients* [20] can be formally written as Equation 43. Notably, Equation 42 is *expected gradients* written in terms of *integrated gradients* when sampling from a distribution, hence is the same equation provided in Equation 36.

$$\begin{aligned}
\phi_i^{EG}(f,x;\mathbb{D}) &= \int_{x'} \phi_i^{IG}\left(x,x'\right) p_D\left(x'\right) dx' && (42)\\
&= \int_{x'}\left(\left(x_i - x_i'\right) \times \int_{\alpha=0}^{1} \frac{\partial f\left(x' + \alpha\left(x-x'\right)\right)}{\partial x_i} d\alpha\right) p_D\left(x'\right) dx', \\
&= \mathop{\mathbb{E}}_{x'\sim\mathbb{D},\alpha\sim U(0,1)}\left[\left(x_i - x_i'\right) \times \frac{\partial f\left(x' + \alpha \times \left(x-x'\right)\right)}{\partial x_i}\right] && (43)
\end{aligned}$$

4.3 Expected Gradient-weight Class Activation Mapping

4.3.1 *path integrated gradients*

To enhance and augment the capabilities of Grad-CAM, through the incorporation of EG, it is necessary to delve into the fundamental notion

behind IG i. e., *path integrated gradients* [63]. As discussed in Section 4.2, IG operates accumulating the gradients along the *straight line* between a defined baseline, namely x' and the input x [63]. Notwithstanding, this is concretely only one of many *nonlinear* possible paths [63] that can be drawn to interpolate two points. Therefore, given a *smooth*, that is continuous and differentiable, *path function* $\gamma = (\gamma_1, \ldots, \gamma_n) : [0,1] \to \mathrm{R}^n$ that *monotonically* connects the baseline x' with the input x, i. e., $\gamma(0) = x'$ and $\gamma(1) = x$, *path integrated gradients* [63] is defined in Equation 44. Concluding that IG is a specific type of *path methods* that *integrates gradients* [63] such that the interpolation function γ is a straight line (Equation 45).

$$\phi_i^{\gamma}(f,x) = \int_{\alpha=0}^{1} \frac{\partial F(\gamma(\alpha))}{\partial \gamma_i(\alpha)} \frac{\partial \gamma_i(\alpha)}{\partial \alpha} d\alpha \quad \textit{where} \quad \begin{matrix} \gamma(0) := x' \\ \gamma(1) := x \end{matrix} \tag{44}$$

$$\gamma^{IG}(\alpha) = x' + \alpha \times \left(x - x'\right) \quad \textit{where} \quad \alpha \in [0,1] \tag{45}$$

4.3.2 *Local Integrated Gradients*

Drawing from the insights obtained in the earlier section, we can reformulate the Grad-CAM approach, discussed in Section 3.5, by altering Equation 29 gradient computation, with the *path method* IG, using the generalized *path integrated method* i. e., we reformulate Equation 29 in terms of Equation 44, obtaining the averaged *integrated local sensitivity maps* (Equation 47); where $y^c_{\gamma(\alpha)}$ represent the output towards class c using the *interpolation* function γ at point $alpha$, and $A^{k,l}_{i,j}$ is the rectified feature map k at layer l and at spatial coordinates i, j. Additionally, in Equation 48 is presented an equivalent version, with the same notation as the generalized *path integrated gradient* (Equation 44). Expanding the recent obtained formulation of Grad-CAM (Equation 47), plugging Equation 46 we obtain the expanded integral in Equation 49. Ultimately, L^c_{IGC} represents the Grad-CAM version implemented using *Integrated Gradients* (IG).

$$\frac{\partial \gamma(\alpha)}{\partial \alpha} = \frac{\partial}{\partial \alpha}\left[x' + \alpha\left(x - x'\right)\right] = x - x' \tag{46}$$

$$L^c_{\mathrm{IGC}} = \int_{\alpha=0}^{1} \mathrm{ReLU}\left(\frac{1}{Z}\sum_k^N \sum_i \sum_j \frac{\partial y^c_{\gamma(\alpha)}}{\partial A^{k,l}_{i,j}} \frac{\partial \gamma(\alpha)}{\partial \alpha} d\alpha\right) \tag{47}$$

$$\equiv \int_{\alpha=0}^{1} \mathrm{ReLU}\left(\frac{1}{Z}\sum_k^N \sum_i \sum_j \frac{\partial F^c(\gamma(\alpha))}{\partial A^{k,l}_{i,j}} \frac{\partial \gamma(\alpha)}{\partial \alpha} d\alpha\right) \tag{48}$$

$$L^c_{\text{IGC}} = \int_{\alpha=0}^{1} \text{ReLU}\left(\frac{1}{Z}\sum_k^N \sum_i \sum_j (x_{i,j} - x'_{i,j}) \frac{\partial F^c(x' + \alpha\,(x - x'))}{\partial A^{k,l}_{i,j}} d\alpha\right) \tag{49}$$

4.3.3 *Local Expected Gradients*

Continuing the work proposed in Section 4.3.2, is possible to rewrite the *integrated gradient* version of Grad-CAM (Equation 47), by applying the same technique expressed in Section 4.2.1 i. e., given a distribution $\mathbb{D}$ and a baseline x', we integrate over all baselines $x' \in \mathbb{D}$ weighted by their *probability density function*. Similarly to Equation 36, rewriting Equation 47 in terms of distribution $\mathbb{D}$ we obtain the double integral in Equation 50. Analogously, as done in Equation 43, since the sampling occurs over a distribution, it is possible to remove the double integral by conceiving it as an expectation over the given distribution (Equation 51), i. e., the baseline is sampled from the distribution $x' \sim \mathbb{D}$ and α from an uniform distribution $\alpha \sim U(0,1)$. Correspondingly as before, equivalent formula using *generalized path integrated gradient* notation is provided in Equation 52, while the expanded version is presented in Equation 53

$$L^c_{\text{EGC}} = \int_{x'} \int_{\alpha=0}^{1} \text{ReLU}\left(\frac{1}{Z}\sum_k^N \sum_i \sum_j \left(\frac{\partial y^c_{\gamma(\alpha)}}{\partial A^{k,l}_{i,j}} \frac{\partial \gamma(\alpha)}{\partial \alpha} d\alpha\right) p_D(x') dx'\right) \tag{50}$$

$$= \text{ReLU}\left(\frac{1}{Z}\sum_k^N \sum_i \sum_j \mathop{\mathbb{E}}_{x' \sim \mathbf{D}, \alpha \sim U(0,1)} \frac{\partial y^c_{\gamma(\alpha)}}{\partial A^{k,l}_{i,j}} \frac{\partial \gamma(\alpha)}{\partial \alpha}\right) \tag{51}$$

$$\equiv \text{ReLU}\left(\frac{1}{Z}\sum_k^N \sum_i \sum_j \mathop{\mathbb{E}}_{x' \sim \mathbf{D}, \alpha \sim U(0,1)} \frac{\partial F^c(\gamma(\alpha))}{\partial A^{k,l}_{i,j}} \frac{\partial \gamma(\alpha)}{\partial \alpha}\right) \tag{52}$$

$$L^c_{\text{EGC}} = \text{ReLU}\left(\frac{1}{Z}\sum_k^N \sum_i \sum_j \mathop{\mathbb{E}}_{x' \sim \mathbf{D}, \alpha \sim U(0,1)} \left[(x_{i,j} - x'_{i,j}) \frac{\partial F^c\,(x' + \alpha(x - x'))}{\partial A^{k,l}_{i,j}}\right]\right) \tag{53}$$

4.3.4 *Expected Grad-CAM: averaged expected gradients from data distribution*

Ultimately, drawing from all the antecedent work, proposed in the preceding sections, the final step towards our proposition *Expected Gradient-weight Class Activation Mapping* (Expected Grad-CAM) is to provide a distribution to sample from. Erion et al. in [20] concludes

that the training data distribution outperformed similar methods and yield the *highest color invariance* [20]. Formally, our proposition computes the local importance scores α_c^k as presented in Equation 54, while the *discriminative downsampled coarse heatmap* is computed as Equation 55. Equation 56 presents the discretized way of computing it in practice, where M is a *hyperparameter* defining the number of samples to draw from each respective distribution, i. e., the data distribution $\mathbb{D}$, and s the s^{th} drawn sample. Noteworthy, in accordance to the style of previous work [52, 74] the upsampling has been omitted from the following formulas as it relies on the dimensionality of the input. Moreover, in Equation 56 for clarity has been omitted the dot product with the rectified feature maps A^k, which is however present in the final equation. Figure 9 depicts an illustration of an interpolated time series when the baseline is sampled from the training data.

$$\alpha_k^c = \frac{1}{Z}\sum_i\sum_j \mathop{\mathbb{E}}_{x'\sim\mathbf{D},\alpha\sim U(0,1)} \frac{\partial y^c_{\gamma(\alpha)}}{\partial A^{k,l}_{i,j}} \frac{\partial \gamma(\alpha)}{\partial \alpha} \tag{54}$$

$$L^c_{\mathrm{E\,G\,C}} = \mathrm{ReLU}\left(\sum_k^N \alpha_k^c A^k\right) \quad \textit{where} \quad l = L-1 \tag{55}$$

$$\begin{aligned} L^c_{\mathrm{E\,G\,C}}(M) &= \mathrm{ReLU}\left(\frac{1}{Z}\sum_k^N\sum_i\sum_j \mathop{\mathbb{E}}_{x'\sim\mathbf{D},\alpha\sim U(0,1)} \frac{\partial y^c_{\gamma(\alpha)}}{\partial A^{k,l}_{i,j}} \frac{\partial \gamma(\alpha)}{\partial \alpha}\right) \\ &= \mathrm{ReLU}\left(\frac{1}{Z}\sum_k^N\sum_i\sum_j \mathop{\mathbb{E}}_{x'\sim\mathbf{D},\alpha\sim U(0,1)} \left[(x_{i,j}-x'_{i,j})\frac{\partial F^c\left(x'+\alpha(x-x')\right)}{\partial A^{k,l}_{i,j}}\right]\right) \\ &= \mathrm{ReLU}\left(\frac{1}{Z}\sum_k^N\sum_i\sum_j \frac{1}{M}\sum_{s=1}^M (x_{i,j}-x'^s_{i,j})\frac{\partial F^c\left(x'^s+\alpha^s(x-x'^s)\right)}{\partial A^{k,l}_{i,j}}\right) \end{aligned} \tag{56}$$

4.3.5 *DFP: Double Forward Pass*

A fundamental component of IG, but more generally of *difference-from-reference* approaches, is composed by the *difference-from-baseline* term i. e., the $(x-x')$ present in Equation 42 and subsequently in the final formulation of Expected Grad-CAM (Equation 56). This term, as discussed in Section 4.3.1, is the $\frac{\partial \gamma_i(\alpha)}{\partial \alpha}$, when γ is a linear path. This implies that the implementation is *trivial* when the *computation occurs at the input*, i. e., as presented in the original formulation, since all the terms involved have the same dimensionality. When computing the partial derivative *w.r.t.* intermediate layers, due to the network architecture and *convolution/pooling arrangements*, the tensor shape largely

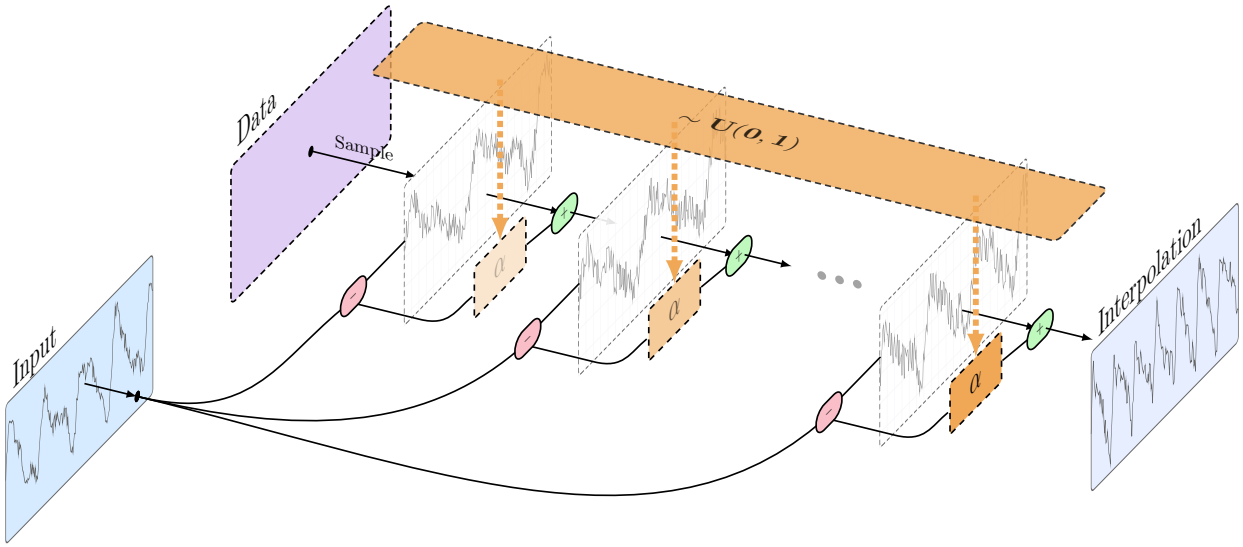


Figure 7: Expected Grad-CAM baseline/input interpolation overview. The *interpolation* is the result of $x'^s + \alpha^s \left(x - x'^s\right)$

differs from the original input shape x; this renders the product of the *difference-from-baseline* and the derivative not readily executable. This posed an inherent challenge which spans beyond the problem of dimensionality and involves understanding the original intent of this term and translating it for any arbitrary intermediary layer. Prior methods either utilized a different variation of IG, which does not involve it [42] or they circumvented it altogether [50]. More formally, in the context of Expected Grad-CAM, the product under analysis is shown in Equation 57. In view of the fact that we are interested, as in the original problem formulation, in the $\frac{\partial \gamma(\alpha)}{\partial \alpha}$ but encoded at layer l, then the quantity $\frac{\partial \gamma(\alpha)}{\partial \alpha}$ can be computed by a single *partial forward and backward pass* upon which the gradients and activations components of the encoded difference are extracted and multiplied with the second term components. Moreover, we are looking for the transformed version of the difference between the input and the currently sampled baseline, at the step α, where the transform function f is the model itself, which transform the interpolation from the input space, to any arbitrary layer l (Figure 8). This encompasses that such operation is computed for each step alongside the path γ at each point α for as many times as there are samples to draw (M in Equation 56). This is essential to obtain faithful explanations (*More in Section 4.3.3*), and it requires to be computed at each step of the interpolated path γ as the baseline x' is differently drawn at each step α.

$$\cdots \frac{\partial \gamma(\alpha)}{\partial \alpha} \frac{\partial y^{\varepsilon}_{\gamma(\alpha)}}{\partial A^{k,l}_{i,j}} = \left(x_{i,j} - x'_{i,j}\right) \frac{\partial y^{\varepsilon}_{\gamma(\alpha)}}{\partial A^{k,l}_{i,j}} \tag{57}$$

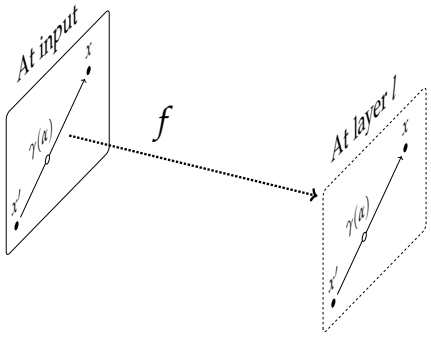


Figure 8: DFP: difference-from-baseline transformation

4.4 Expected Gradient-weight Class Activation Mapping++

Although recent studies [34] have shed some light on the actual dubious improvement yielded by Grad-CAM++, since some prior methodologies have relied upon, and, a core aim of this work is to provide a *gradient-safe* CAM replacement, Expected Grad-CAM is further augmentated with the proposition presented in the original paper [9, 10]. Due to the unclear math behind Expected Grad-CAM++ (Equation 58), has been adopted a similar solution utilized by prior method such as [42] and implemented in the popular *TorchCAM* [21] library rather than the one presented in the original work. Thence, the partial derivative *w.r.t* to the target feature map i. e., $\frac{\partial y^c_{\gamma(\alpha)}}{\partial A^{k,l}_{i,j}}$, present in the original formulation of Expected Grad-CAM can be rewritten as Equation 59 representing the gradient formulation of Expected Grad-CAM++. In conclusion, by plugging Equation 59 into Equation 60 is obtained the final formulation of Expected Grad-CAM++ (Equation 60).

$$\alpha^{kc}_{ij} = \frac{\frac{\partial^2 Y^c}{\left(\partial A^k_{ij}\right)^2}}{2\frac{\partial^2 Y^c}{\left(\partial A^k_{ij}\right)^2} + \sum_{ab} A^k_{ab}\left\{\frac{\partial^3 Y^c}{\left(\partial A^k_{ij}\right)^3}\right\}} \tag{58}$$

$$\frac{\partial y^c_{\gamma(\alpha)}}{\partial A^{k,l}_{i,j}} = \frac{\left(\frac{\partial y^c}{\partial A^k_{i,j}}\right)^2}{2\left(\frac{\partial y^c}{\partial A^k_{ij}}\right)^2 + \sum_{i,j} A^k_{i,j}\left(\frac{\partial y^c}{\partial A^k_{ij}}\right)^3} \tag{59}$$

$$L^c_{\mathrm{E\,G\,C}}(M) = \mathrm{ReLU}\left(\frac{1}{Z}\sum_k^N \sum_i \sum_j \mathop{\mathbb{E}}_{x'\sim\mathbf{D},\alpha\sim U(0,1)} \frac{\left(\frac{\partial y^c}{\partial A^k_{i,j}}\right)^2}{2\left(\frac{\partial y^c}{\partial A^k_{ij}}\right)^2 + \sum_{i,j} A^k_{i,j}\left(\frac{\partial y^c}{\partial A^k_{ij}}\right)^3} \frac{\partial \gamma(\alpha)}{\partial \alpha}\right) \tag{60}$$

4.5 Guided Expected Gradient-weight Class Activation Mapping

As previously discussed in Section 3.4 and subsequently in Section 3.5, Grad-CAM-based approaches, but more generally CAM maps are inherently *class-discriminative* [52], but due to the spatially coarse nature of the heatmap, they lack the ability to highlight fine-grained details [52] which by contrast are present in *pixel-space* gradient methods [52]. A solution was first introduced in the original Grad-CAM paper, which involved the fusion of a CAM generated map and typically a gradient-based method *w.r.t.* to the input; this initiated a wider class of approaches which are, namely, *guided*. In the case of *Guided* Grad-CAM, as the name implies, *Guided Backpropagation* was used, as, at the time of writing of the paper, was the least noisy method after DeconvNet. Albeit, these methods are often referred as *high-resolution* (*More in Section 3.5*), they are just a fusion of two separate methods, whereas the original CAM, acts as a mask for the second method, which is typically produced by some backpropagation, when using *gradient-based* methods, w.r.t. the input. Hereof, our proposition Guided Expected Grad-CAM represents the fusion of the coarse heatmap produced by Expected Grad-CAM with EG (Equation 61). This aims at producing explanation *w.r.t.* the input which is also alleviated by the gradient issues discussed in Section 3.6 and produces overall less noisy maps. Due to the architecture of Expected Grad-CAM (Section 4.3), the ϕ_i^{EG} w.r.t. to the input can be computed efficiently as *autograd* already partially tracked and computed the gradients up to the target feature map; therefore, a smaller tree traversal is necessary compared to computing ϕ_i(s) separately. This is further improved by the implementation of a caching strategy that allows to store and share baseline's tensors in-between Expected Grad-CAM executions.

$$H^c_{\text{EGC}}(M) = L^c_{\text{EGC}}(M) \cdot \phi_i^{EG}(f, x; \mathbb{D}) \tag{61}$$

4.6 explainer: optimizations and practical remarks

Although our proposition provides a wide and extensive theoretical coverage, significant emphasis and effort has been put on practical considerations; core aim of this work has been to develop not only the theoretical aspects but also solutions that were readily deployable in *real-life* scenarios and in modern *multi-node* cloud environments. Following are presented a set of practical remarks, technical solutions, and optimizations.

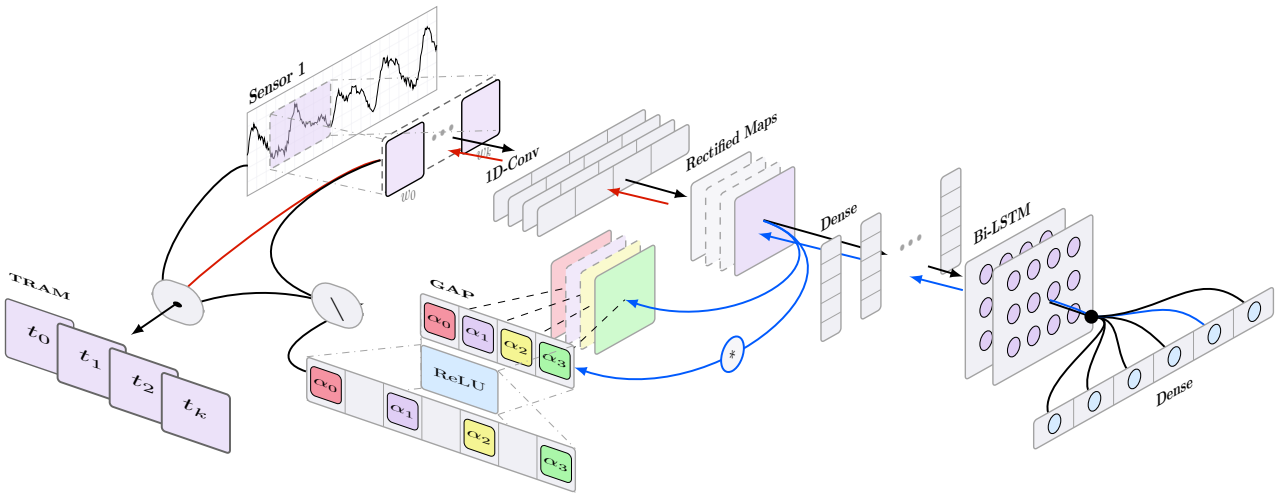


Figure 9: Proposed method *Expected Gradient-weight Class Activation Mapping* (Expected Grad-CAM) overview. Excerpt of a multi-headed CNN-BiLSTM (stacked) - slice of individual head

4.6.1 *Convergence and Quality: A formal definition*

Expected Grad-CAM requires the sampling and generation of multiple *interpolated* tensors, where the number of interpolations correspond to the steps along the path resulted from the selected *interpolator function* γ (Equation 45), i. e., the discrete count of *samples* M to draw from a given distribution $\mathbb{D}$ (Equation 56). Similarly to IG and consequently EG, Expected Grad-CAM requires a sufficient number of draws M in order to obtain *satisfactory results*; more concretely, we define a coarse heatmap to be *satisfactory* when the sum of cumulative gradients stops diverging or settles within a threshold ϵ i. e., $|\sum_{i=1}^{n} \nabla f^c(x_i)| < \epsilon$. This yields a *single-valued* metric that enables the measurement of an even more crucial parameter: *convergence*. Although this value can be employed to automatically set the only significant *hyperparameter* of Expected Grad-CAM, namely the number of samples to draw M, its value should not be confused or interpreted as a quality measure. Counterintuitively, a lower *convergence difference* may not result in a higher quality heatmap, seemingly a higher*convergence difference* does not necessarily lead to a less *satisfactory* heatmap. Howbeit, in practice it is often the case that a better *convergence-difference* will yield in a higher scoring map (according to reference metrics). Previously, the term "*convergence-difference*" has been used to refer to the quantity obtained from Section 4.6.1 as in this field *convergence-delta* is often used in place for the *completeness-axiom* (Equation 34). As discussed in Section 4.2, the *completeness-axiom* can be equally exploited to measure *convergence*. All the points raised for the *convergence-difference* are also valid for the *convergence-delta*, with exception, that the latter is more computationally intensive, however, it implicitly enforces a

very desirable property: *completeness*. Ultimately, both provide a weak condition to model a satisfactory heatmap, as a draw-sampling stopping strategy with different degrees of computational overhead and involvement, while providing an effective *proxy value* to assess convergence.

- **convergence-difference**: $|\sum_{i=1}^{n} \nabla f^c(x_i)| < \epsilon$
- **convergence-delta**: $|\sum_{i=1}^{n} \phi_i\left(f, x, x'\right) - \left(f(x) - f\left(x'\right)\right)| < \epsilon$

4.6.2 *Convergence Drop Rate and Mini-Batching*

As discussed in the previous section, Expected Grad-CAM requires the computation of multiple subsequent draws and interpolations which may result in a costly operation as multiple baselines have to be sampled and moved to the target device, when a dedicated *GPU* or different *node* or *accelerator* is used; this can rapidly become an issue in resource-limited environments or where *VRAM* usage is bounded. Furthermore, this can represent a bottleneck that can jointly limit the explainer *time performances* and *convergence drop rate*. In this regard, an extremely simple and widely adopted approach within AI can be adopted: *Mini-Batches*. This apparently simple but extremely powerful technique does not just solve all the aforementioned issues, but resolves an intrinsic nontrivial limitation of Expected Grad-CAM, which highly limits the explainer's performances. This phenomenon involves that after a certain threshold, as the number of draws increases, the *convergence drop rate* starts to decline hindering the *speed* and *time-to-convergence*. This is not necessarily an issue, as it is a common behavior for many *difference-from-reference* approaches, but nevertheless increases the explainer's time complexity, requiring higher computation times to achieve *satisfactory results*. Figure 10 depicts an example of the phenomenon. As previously introduced, the solution is to use a *Mini-batch*-like technique. Given $\Gamma = (\gamma_1, \ldots, \gamma_n)$ being the superset of all *intepolator functions* such that $\gamma_i(0) = x'_i$, where x' is the i^{th} baseline sample tensor, then for K *mini-batches* we would select K distinct *interpolator functions* i. e., $(\gamma_1, \ldots, \gamma_k)$. This implies that we would have K paths, each originating at a different point in the vector space, that is, from a different baseline sample, and all converging at the same point x i. e., the input tensor. Alternatively, this can be viewed as a single "*super*" path where the starting point is the *combination* of all the *interpolator functions* at $\alpha = 0$, i. e., $\gamma_i(0)$. Consequently, the result will be K coarse heatmaps i. e., $L^c_{\text{EGC}}(M)_1, \ldots, L^c_{\text{EGC}}(M)_k$. This, however, when compared to the traditional approach, requires an additional step which involves averaging the heatmaps and subsequently normalizing them. Notably, it is crucial to note that in this

context the usage of the term *mini-batches* differs from its conventional meaning within the DNN field; the number of *mini-batches*, in this case, are not related to the number of tensors moved to a target device, nor have any performance-related implication. Ultimately, this technique provides a "*superconvergence*-like" effect (*More in Section 6.1*) alleviating or completely solving *(I) the issue caused by pixel-wise saturation of over-sampled interpolations, (II) poor sampled baselines with similar pixel-wise variance, (III) insufficient number of drawn samples, and, (IV) impossibility to achieve satisfactory maps due to memory constraints*.

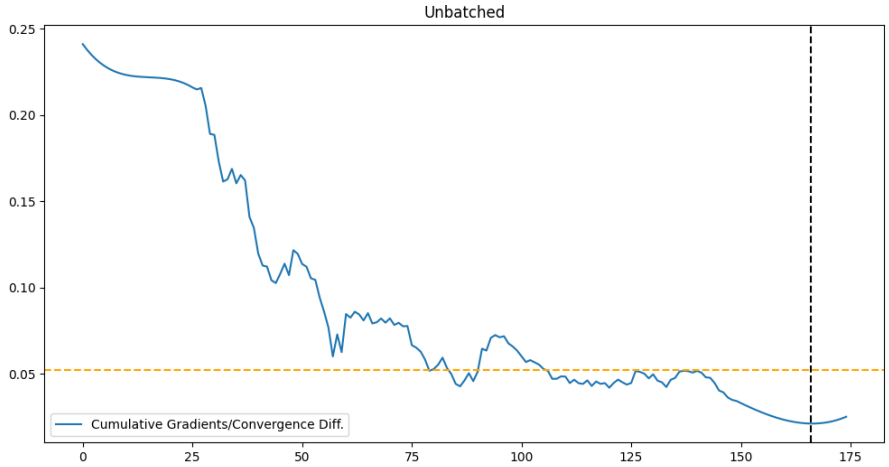


Figure 10: Cumulative gradients/Convergence difference of Expected Grad-CAM - Unbatched

4.6.3 *Baseline Sampler: Towards Reproducibility*

Toward higher reproducibility and a more deterministic behavior of the explainer, especially for research purposes, even at extremely low sample counts, the baseline sampler is *user-settable* allowing to provide a *SequentialSampler* with *user-defined* sample's indices. Figure 11 illustrates the difference between sampling from a linear and uniform space.

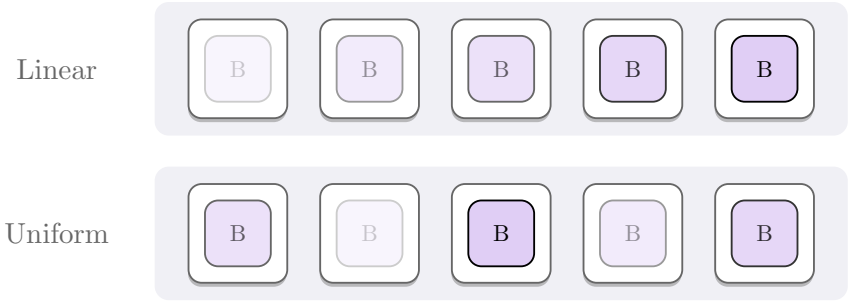


Figure 11: Baseline sampling from a uniform vs. linear space

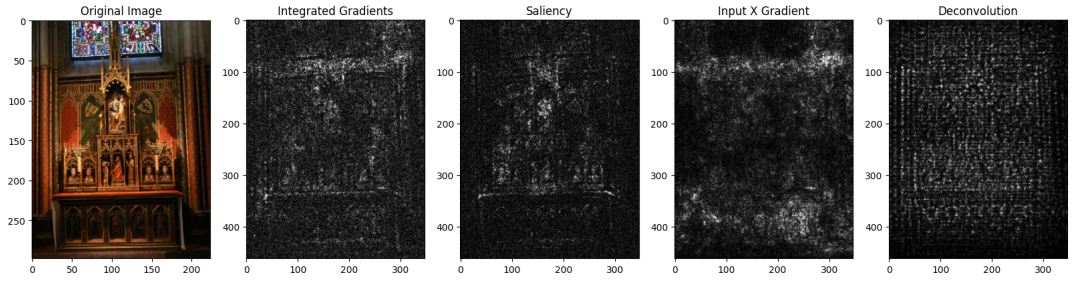


Figure 12: Pixel-wise saliency comparison.

4.7 BEYOND FAITHFULNESS

In the upcoming sections, we challenge the current state and formulation of *visual explanations* within the XAI field while investigating and rethinking faithfulness as a whole. We then formalize a set of properties that an *informative* saliency should respect to provide meaningful information. Ultimately, we devise an approach that leverages two novel ideas that go in the opposite direction of prior works:

1. **Resolution is not just pixels. *Frequency Decomposition* is all you need.**

2. **Saliencies are not informative nor faithful: they do not follow the *natural encoding*.**

4.7.1 *Rethinking Visual Explanations*

Current *visual explanation methods* within the field of XAI, as introduces in Chapter 2, and discussed more broadly in Chapter 3, are either *gradient-based* such as [57, 71, 61] or in the form of *Gradient x Input* [63, 20, 54], *pertubation-based* [71, 43] or CAM-based [74, 52, 42, 66]. *Pixel-space* gradient visualization are often noisy (Figure 12), but do provide insights with regard to finer details. On the other hand, CAM-based approaches provide *class discriminative* image region highlights, but fail to capture fine-grained details (Figure 13). Seemingly, this implies that the latter approach points out the relevant portions of the image that are of importance with respect to the target class, but it does not identify significant features within it i. e., the reason behind the prediction. This motivated the development of *Guided* methods, which not only produce *class discriminative* maps but also capture the *fine-grained* details within the image. Nonetheless, these middle-ground solutions naturally inherited not only the strength of both methods, but also their weaknesses. In the following subsections, we investigate the flaws and shortcomings of both strategies.

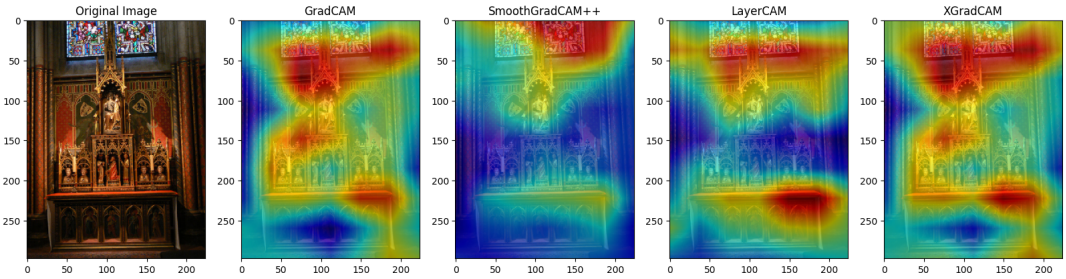


Figure 13: CAM saliency comparison.

4.7.2 *Pixel-wise Saliency Maps are not Informative*

Continuing from the previous section, for *Guided* approaches, the *fine-grained* composition of the saliency is constructed from *pixel-wise* saliency methods, typically gradient-based. Such visualization constructs details as sparse circular activations, where typically the alpha channels, is used to determinate the *feature-wise* pixel importance. An example is depicted in Figure 14 using Integrated Gradients (IG). Despite the noise, such saliencies are neither *expressive*, nor *informative* toward a human-centered understanding as they are not created using visual cues and tools that the human visual system uses to identify objects nor provide meaningful insights on the underlying DNN's mechanisms. The human visual system understands objects through shapes and patterns, which in turn are made up of lines, curves, and edges which are hardly encoded by such circular sparse dots present in current saliencies. Moreover, current visual explanations do not provide meaningful, expressive, nor relevant insights towards the underlying inner understanding of the input that the network has, that is, the actual encoding that the model has created, as a compressed representation towards the various layers; in other words, if we sample a set of pixels $p_1, \ldots, p_n \in N$, where N is a neighboring area, from such saliencies (e. g., Figure 14) by inspecting the pixel p_k at position (i, j) we can gauge its *feature importance* encoded as the alpha channel, but we cannot understand precisely whether the model is evaluating the *shape*, that is, a set of lines that make up the contour, or the pattern of the object. More strictly, we cannot determine what type of feature detectors in the neighborhood N are activated. For example, in Figure 14, in the *Close Up 2*, we cannot establish whether the model is focusing on the shape of the body of the snake, on the shape of the dorsal scales, or on the pattern intended as the texture, of the scales itself. Ultimately, these types of saliencies are meant to be "*high-resolution*", that is, to provide and highlight the finer details of the image, however, as observable in Figure 14 (*Close Up 1*), despite the the map itself has the same resolution of the input image i. e., the same number of pixels, it is clear that it is not able to distinguish the not so small parts of the image, such as the pupil or the brille, both

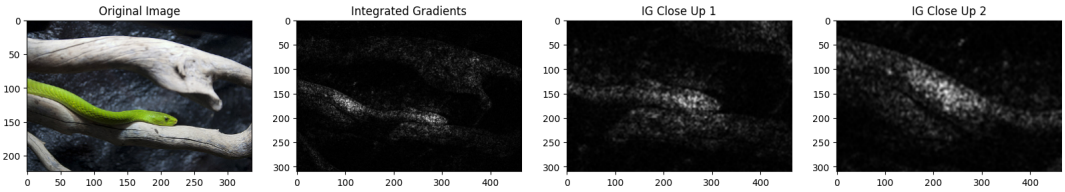


Figure 14: *Integrated Gradients* (IG)[63] saliency comparison with detail close ups

of which, due to the concentrated number of activations in the area, most likely are very significant for the model.

4.7.3 *CAM Saliency Maps are not Informative*

CAM methods, in contrast to the approaches described in the previous sections, offer visual explanations with respect to the intermediary layers of the network, typically within the last convolutional blocks. These layers typically contain the highest-level and most abstract features, which are often the most relevant toward any given class prediction. The spatial resolution of the resulting heatmap, generated by intermediate layers, is significantly reduced compared to the original input, due to the architecture and topology of convolutional-based DNN, predominantly influenced by the *pooling* operations. For instance, given a *VGG-16*, a $214x214$ input image will result in a $7x7$ when explaining the last layer in the feature block, which is extremely compressed compared to the input. Figure 15 shows the coarse heatmap generated using Grad-CAM as well as its up-sampled superimposed version. Notably, the first input (Figure 15) of size $224x298$ generates a heatmap of size $7x9$ while the second sample of size $303x224$ produces an heatmap of size $9x7$. CAM explanations, similarly to the *gradient*-based methods described earlier, also suffer from resolution's issue; however, in this case, they are caused by the low pixel count of the map itself i. e., the spatial resolution. The issue becomes particularly evident and pronounced when the explainer attempts to highlight the contour or edges of objects (Figure 15) which results in the highlighting of incorrect and wrongfully explained elements, especially in images with multiple objects. This undesirable effect will compound in *Guided* approaches, as the upscaled heatmap is used as a mask and incorrectly hides relevant portions of the images.

4.7.4 *Rethinking faithfulness*

Faithfulness describes the extent to which an explanation accurately reflects the underlying DNN's mechanics and dynamics towards a given prediction. This implicitly entails that a *faithful* explanation should

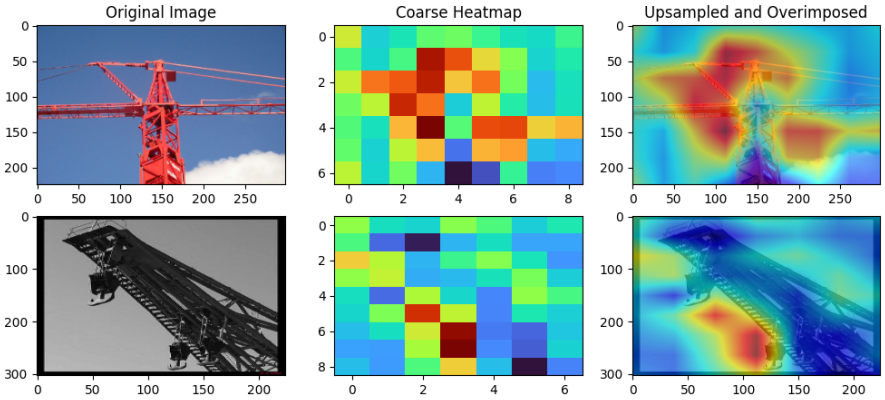


Figure 15: Side-by-side comparison of correctly classified "*crane*" of the coarse heatmap (middle column) and upsampled overimposed heatmap (rightmost column)

capture the most *important* and *relevant* features that *contribute* to the model's prediction. *Importance* and *feature relevance* are not trivial and are encoded differently depending on the XAI method adopted. In previous sections have been investigated the weaknesses and saliency properties of some *gradient*-based methods, in this regard in Figure 12, it is visible how, for *input-gradient* the notion of *importance* is encoded as the individual *pixel sensitivity* which, from the model standpoint, is its sensitivity with respect to perturbation of the input. This is one of the earliest type of explanations i. e., where the saliency S is defined by the gradient of the model w.r.t. the input $S = \nabla_x f(x)$. More advanced method, which detects contributions within the *flat regions* of the model's function approximator i. e., they do not suffer from gradient saturation, such as IG or Expected Gradients (EG), provides a different notion of *importance* which enforces that the sum of the *contributions* adds up to the model's output. This observation is of significant importance as such attributions provide only a weak condition and dependence, but do not define a concrete notion of saliency [62].

As *faithfulness* describes the scale and degree of which an explanation adheres to the model's inner workings, then the notion of importance encodes the conditions at which it occurs. Neither of which provides a strong requirement, nor notion, of how such quantities should be encoded or represented. Consequently, we argue that a *faithful* saliency should reflect the model's behavior with *human-interpretable* encoding i. e., with elements that are part of our visual extraction system such as *edge, lines, angles* and *textures*. In addition, it should accurately represent the progressive build-up understanding, and compressed representation, that the model has, of the features within the input image. This implies that, for any given neighborhood, should be possible to understand what types of feature detectors, of the model, have been activated. This will yield the saliency extreme expressiveness and interpretability, as for any given spatial

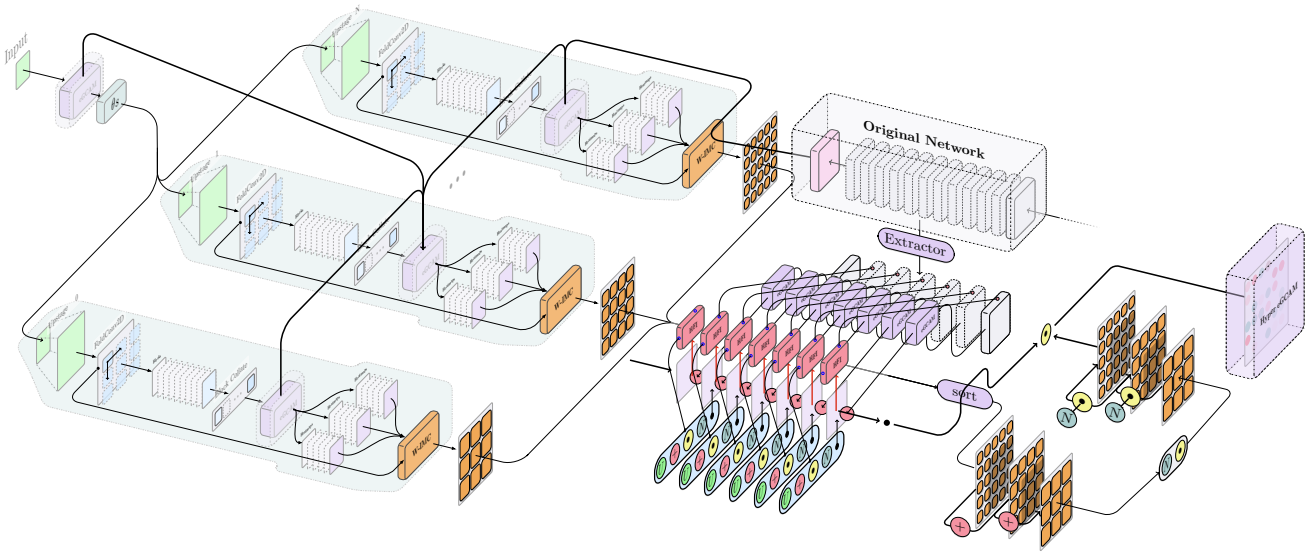


Figure 16: Hyper Expected Grad-CAM – Complete Overview of all interoperating components and stages.

location in the map, it would be possible to **understand and differentiate the real intent and focus of the model** i. e., identify whatever the model is looking at the edge of the object, its contour (shape), the angle or the texture. This means that despite two saliency **might have the same local faithfullness**, according to quantitative metrics (*More in Section 5.1*), **one may possess higher informative power** than the other. More formally, *faithfullness* is not a strong condition for saliency *expressivity* nor *informative power*. Conversely, this also implicates that a map can be *faithful* but not *informative*. We argue that such encoding is the *natural way* of encoding visual explanations which both satisfies the model *faithfulness* and *human interpretability*, ultimately, providing a saliency map which concretely fulfills its original goal: to explain to a human audience.

4.8 Hyper Expected Grad-CAM

Building on the intuitions and properties described in the prior sections, we propose a novel method, namely Hyper Expected Grad-CAM, which fulfills the desirable properties discussed in Section 4.7.4. This generates a new, *more informative*, *hybrid* type saliency which jointly satisfies *faithfulness*, according to our notion of *faithfulness* (More in Section 4.7.4), and *human-interpretability*. We refer to such *encoding* as the *natural encoding*, as it follows both the human visual system as well as the convolution-based DNN's feature extraction mechanisms. Following are presented some relevant parts of Hyper Expected Grad-CAM. Figure 16 depicts an overview of the Hyper Expected Grad-CAM explainer, showing how all components interoperate.

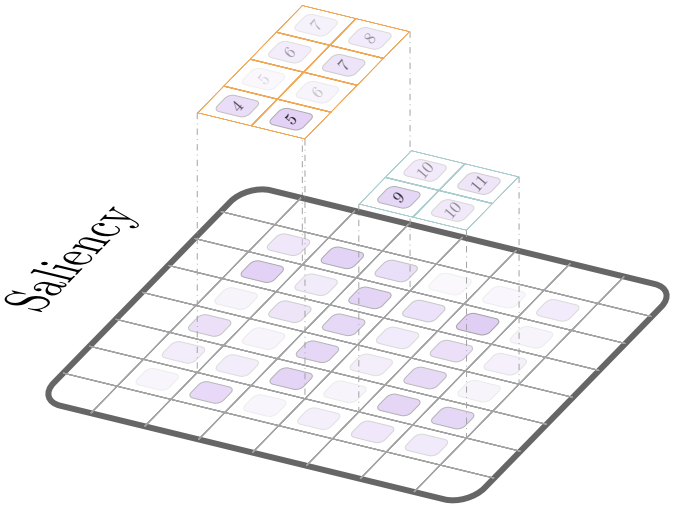


Figure 17: Local vs. Global Completeness Illustration

4.8.1 *Constraints*

According to the new notion of *faithfullness* and *human-interpretability* discussed in Section 4.7.4, we can explicitly extract the following set of constraints that our novel method follows and obeys to:

Constraint 1. The input dimensionality, in its entirety, has to match the original input at all stages, both in terms of spatial resolution i. e., number of pixels or *PPI* as well as retaining the original aspect ratio. More formally this entails that the *inflation* and *deflation* rate remains paired with the sample to explain. This property ensures *faithfulness*.

Constraint 2. The saliency has to follow the *natural encoding* (as formalized in Section 4.7.4). This ensures both model's *faithfulness* and *human interpretability*.

Constraint 3. The input, and consequently all intermediary maps, at any given point in the pipeline, must preserve the original *global completeness* w.r.t. to the input. This ensures that the relative hierarchical correlation between *neighborhood* is preserved. Concretely, implies that given set of pixels $p_1, \ldots, p_n \in N$, where N is a neighboring area, if we sample a pixel $p_k \in N$, where $P_{adj} \subset N$ is the subset of adjacent pixel of p_k, then $\forall p_a \in P_{adj}$ must preserve their relative order i. e., spatial positions. This must hold for any arbitrary size of N i. e., stretching from containing 1 pixel to all pixels in the image.

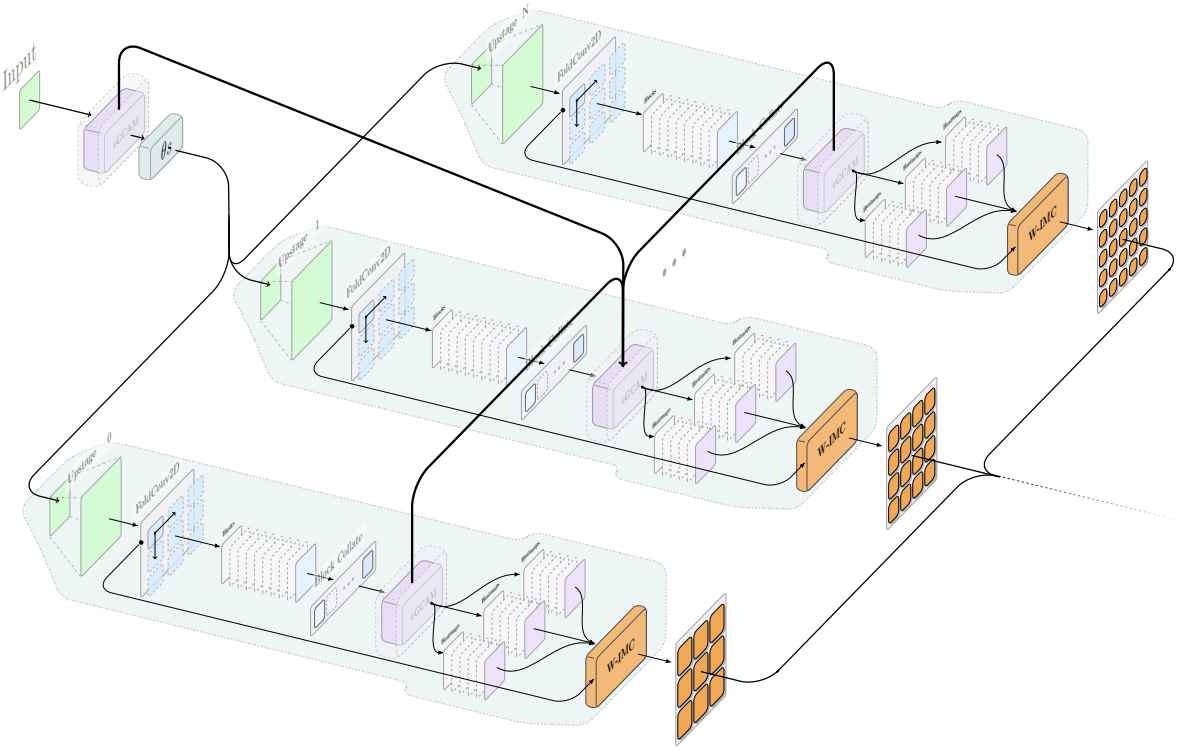


Figure 18: Hyper Expected Grad-CAM - Multistage Feature Dependency Extraction Illustration Extract.

As a side remark, is important to point out that because of the *constraints* described above, *Input x Gradient* or any *gradient*-based method w.r.t. to the input cannot be used as it violates the *natural encoding* and notion of *faithfulness* (**Constraint 2.**). This applies also to any utilization of the input for any intermediary operation; such methods do not encode, as shown in Section 4.7.2, the model's understanding and compressed representation in terms of visual feature extractors. Prior works within XAI also operate by upscaling the input, changing its aspect ratio, or any other input-related manipulation. All these operations are not permitted as they violate the first constraint (**Constraint 1.**). This includes any transformation applied to the input, such as *CenterZoom* or even squaring the image to a fixed typical arbitrary size (e. g., 224*x*224). Furthermore, this is not allowed because it implicitly alters the *deflation rate* between the input layer and the target layer.

4.8.2 *Parallelizable Multiple Multi-stage Pipelines*

The production of *Hyper Expected Gradient-weighted CAM*s involves *multiple multi-stage pipelines* each serving a distinct purpose and ful-

filling a specific role. With the intent of creating a concrete explainer, deployable in a real-life scenario, each stage of the pipeline has been designed to maximize *parallelization and scalability*. The whole process can be subdivided into four main segments: *(I) Stage parameter's computation, (II) Feature Dependency Extraction, (III) Frequency Decomposition and (IV) CAMs Accumulator*.

4.8.3 *Stage Parameter's Computation*

The main process initiates with the computation of a baseline CAM using our proposition Expected Grad-CAM (*More in Section 4.3*) w.r.t. , the *target layer L* (Figure 18). This operation involves the calculation of the number of stages as well as all the parameters within each stage (θs) (More in Section 4.8.4). This allows to compute the transformation function $f^L(x)$, which maps from an input x, all the transformations applied to it, till the layer L i. e., all the convolution, pooling, and flattening operations that directly influence the tensor dimensionality. The number of stages within each *feature dependency extractor* (*More in Section 4.8.4*) is calculated by estimating the size of the object, or objects, to explain; this is achieved by adding the highly activated attributions and computing the area. The number of stages in the next step is then the *max* number N boxes of given IoU(*Intersection Over Union*) that fit within the pre-computed area. As discussed more broadly in the next section (Section 4.8.4), this allows to extract, and subsequently, explain features of different scales within the input. Hence, such computation does not require to be exact i. e., to accurately estimate the size of the object, but provides a starting point for the further refinement. In fact, empirically has been found that for images of roughly 50176 pixels, three stages are sufficient to capture most dependencies with finer objects (when analyzing the ILSVRC2012 [48] dataset). Figure 18 shows an overview of the first two parts of the pipeline.

4.8.4 *Explaining Feature Dependencies*

As discussed in Section 4.7.4, *local sensitivity* and *global completeness*, despite being desirable properties on their own, cannot be simultaneously satisfied [62]; this implies that if we sample two distinct neighborhoods N_1 and N_2 from a saliency S, such that $\sum_i \phi_i > \sum_j \phi_j$, $\forall \phi_i \in N_1$ and $\forall \phi_j \in N_2$, where ϕ_i and ϕ_j are the i^{th} and j^{th} individual feature attribution within N_1 and N_2 respectively, then the attribution importance of the neighborhood N_1 is not necessary more *relevant* than N_2 i. e., the neighborhood importance cannot be derived from the sum of its individual components (Figure 17). This phenomenon is extremely apparent and relevant when quantifying the explanations with quantitative metrics such as *insertion and deletion* (*More in*

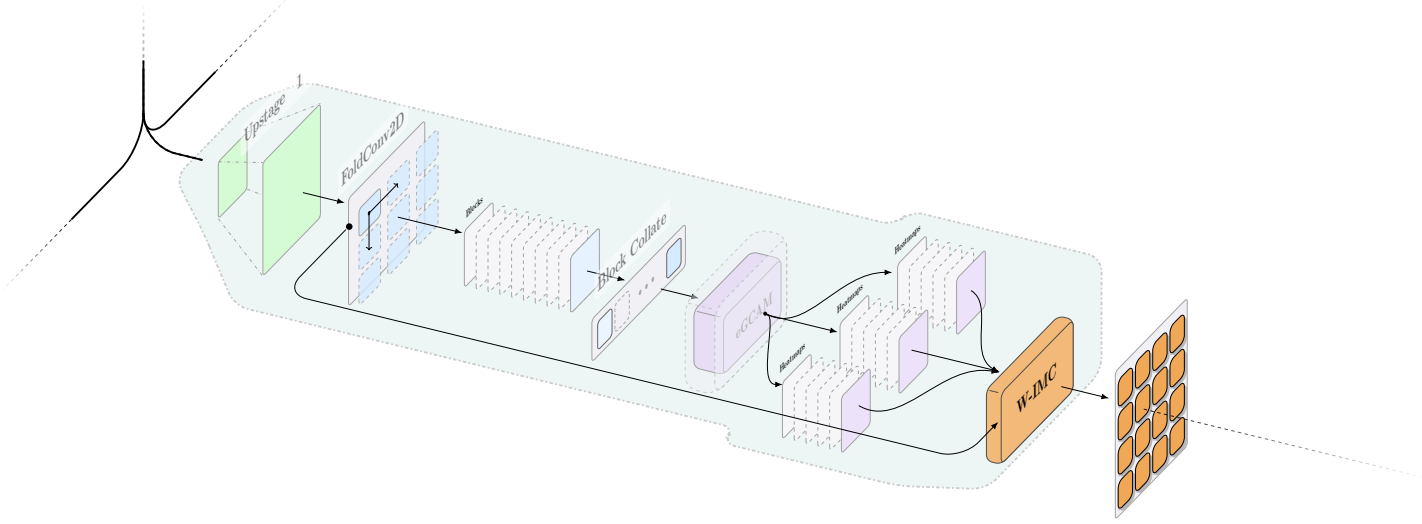


Figure 19: Hyper Expected Grad-CAM - Individual Feature Dependency Extraction Stage Illustration Extract.

Section 5.1). Intuitively, by removing a set of unimportant features, you would expect the *drop in confidence* of the model to decrease in relation with the importance of each feature, that is, to produce an effect that is equal to the sum of effect of the individual attribution. However, in practice, removing a set of features produces a compounding unexpected behavior which is not linearly related to the individual pixel-wise attributions i. e., the effects of such operation (deletion) is not the the results of all the individual causes. In an attempt to mitigate such conditions, we propose a hybrid approach which aims to preserve *global completeness* (**Constraint 3.**) w.r.t. to local neighborhood's. Each stage of this segment of the pipeline (Figure 19) is responsible for generating explanations of neighborhoods of different scales. The first stage extracts neighborhoods, and consequently produces *local explanations*, of dimensions equal to the the size (K_0) of the most activated area pre-computed in the previous step. The extraction is performed through a *Convolution* operation with constant fixed kernel size of size K_0; In practice, this is implemented using a *Folding* operator (Figure 19). Notably, at every kernel stride step the spatial location of each feature map are stored. Each extracted feature map or *block* represents a *local neighborhood* of size K_i of the original input. Each block is then collated into a set of blocks to increase the *throughput* of the method. Every individual block is then sent to Expected Grad-CAM which produces a set of saliencies equal to the number of blocks extracted. Every individual saliency, at this stage, represents a *local explanation* of the neighbor of size K_0 w.r.t. to the original input, when investigated individually. Ultimately, the maps are re-

assembled by the *W-IMC* module using the spatial locations saved during the *FoldConv2D* operations i. e., each *local explanation* is position in the original spatial coordinate in order to create a *supermap* which retain relative *global completeness* (Figure 19). Though, recalling from Section 4.8.1, due to **Constraint 1.**, all such operations are not allowed; the initial *FoldConv2D* operation extract blocks according to a kernel of size K, that is, each resulting feature map will have size K, which $\forall K < X$ where X is the original input to explain, will produce explanations which do not match the original input spatial resolution, violating **Constraint 1**. This is extremely important as such operations alters the original *deflate rate* resulting in *unfaithful* results. Such constraint is fundamental as it ensures that at each intermediary step, the original model to explain operates, at each layer, with the same spatial resolutions maps as in the original sample. Similarly, prior work [28] artifically upsampled the input so that the intermediary maps would have had higher resolution. We argue such operations breaks the notion of *faithfulness*, as despite the model produces more fitting maps, during the explanation process the networks, at each intermediary layer, works with a number of pixels (higher) than the one it had access during the original inference. Therefore it could focus on different details, potentially not distinguishable in the original case due to the lower resolution map, but discernible in the upsampled version. To solve this issue, in our approach, we invert the formulation of the problem: rather than having a input of size X and a *FoldConv2D* with kernel size K, which extract blocks of size $K, \forall K < X$, we use a kernel of size X and we upstage the input such that each stride step has a size K, effectively preserving original block relation. This involves computing the inverse of the *deflate region*, the *inflate region*, that is from going from a feature map of layer l up to the input. Given the learned mapping $f^L(x)$ (Section 4.8.3), for any arbitrary set of layer x and l where l has a lower spatial dimension of x i. e., is a deeper layer, the *inflation region* can be computed according to equation Equation 62. Therefore to preserve the mapping represented by the transformation $f^L(x)$, each neighborhood must have a size that is a multiple of the *inflated region* and each extraction a stride equal to the same quantity. This ensure faithfulness, as, simply put, allows each neighborhood to be equally and distinctly represented by a single coarse attribution in the compressed explanation. More concretely, such mapping, linearly encodes the portion of the original input which is ultimately mapped to a single pixel in the feature map of the target layer L. The input, before being processed by the *FoldConv2D* is then upstaged by a factor equal to a multiple of the *inflate region* w.r.t. to the original size of the kernel. Such set of operations (stage) is performed N times computation (Section 4.8.3), where at each stage the kernel size i. e., the size of the local explanation derived

from a bigger neighborhood, is halved to produces local explanation w.r.t. to features of progressively smaller scale Figure 18.

$$IR_x^l = \left(\frac{x_w}{l_w}, \frac{x_h}{l_h} \right) \tag{62}$$

4.8.5 *Resolution is not just pixels: Frequency is all you need*

Resolution, in the context of images, is often referred as a quantitative metric which quantifies the number of pixels per unit of space (*Pixel per Inch*). Thus, increasing the spatial resolution of an image indicates increasing the number of pixels of such image.

In an attempt to partially address the issues described in Section 4.7.2 and subsequently in Section 4.7.3, prior methods implemented various approaches, all pivoted on the notion stated above: to produce "higher resolution" saliency, formally CAM, the spatial resolution (as number of pixels) of such explanation would need to be increased as well. In one way or another, such methods, involved altering the original *inflation* and *deflation* rates by performing some upsampling, which violates our notion of *faithfulness* (More in Section 4.7.4) and our set of constraints (More in Section 4.8.1).

In contrast with prior works, we devised a method which operates on *frequencies*, grounded on the *natural* notion of *perceived resolution* i. e., the degree at which you can discern and resolve individual features such as *edges, lines and fine structures*, which both satisfies our notion of *faithfulness* and *human-interpretability* (More in Section 4.7). In this context, following our definition of *natural encoding* (More in Section 4.7.4), **there exist only one resolution**, the one at which the model operates at any given layer. More explicitly, if the *resolution* is defined by the degree at which you can discern the individual make up of an image, then for any given layer within a DNN, its resolution is the make up of each *atomic detail* which is derived by the collective composition of its prior filters within the same area, governed by the *inflation* and *deflation* factors of each layer. In this sense, we refer to "*atomic detail*" as the *feature detectors* of the network available up until the *target layer*. A *faithful* saliency w.r.t. to the model, hence one that follows the *natural encoding* (Section 4.7.4), is a representation, by progressive build ups, of the model's understanding and encoding of the make ups of the image, intended as *atomic details*, known up to the layer of interest. This is significant as deeper layers contains higher-level and more sophisticated object and semantic detectors, which may not be available at the desired target layer. Moreover, our notion of *faithfulness* enforces, that a *faithful* saliency is composed by only the *atomic details* that the model's understand *up to* each given layer.

In practice this is performed by extracting a set of *intermediary layers* up to the *target layer*. Each *intermediary layer* is then fed into Ex-

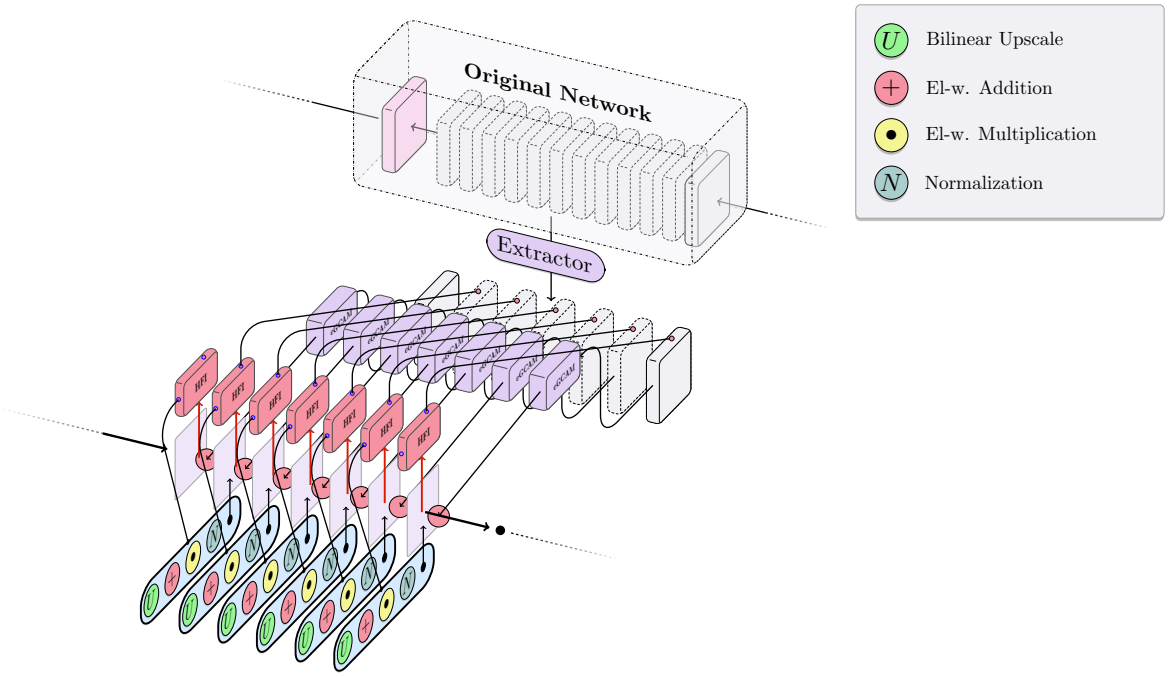


Figure 20: Hyper Expected Grad-CAM - Frequency Decomposition Illustration.

pected Grad-CAM, which produces a *intermediary saliency*. Every *intermediary map* is then pass through into a *high-pass* filter, where the high-frequency components are stored and re-injected into the next extraction step. This is performed to extract and preserve "*atomic details*" produces by *edge* and *curve* detectors of early layers of the network. Such signals needs to be preserved due to the natural compression nature of the *refinement* which suppresses such signals the closer you get to the *target layer*. In this regard, different *high-pass* filters can be employed, from *Laplacian* to *a band-pass FFT*. The important part is that the extraction coefficient has to match the interlayer *inflate rate* i. e., the coefficients have to be paired with the *compression* rate at which each "*representation*" is subjected when passing from one layer to another. Higher compression should translate in a *higher frequency isolation*. Notably, the *intermediary saliency* are generated from the first extracted *intermediary layer* up to the *target layer*, but the *frequency isolation* is performed on the map in the opposite order. Intuitively, during such refinement, you start with the most coarse map and you progressively adds the more abstract "*atomic details*" up to the *target layer*. Ultimately, the *intermediary layers* are selected based on whatever they influence the *interlayer inflate rate* i. e., if they produce a change in the tensor dimensions; such layers or blocks are likely to contain newer *atomic details* or unseen feature detectors. In practice, such layer are chosen according to a $1+1$ policy i. e., the marked layer, as previously discussed, plus the subsequent *ReLU* if present. Figure 20 shows the overview of this segment of the pipeline.

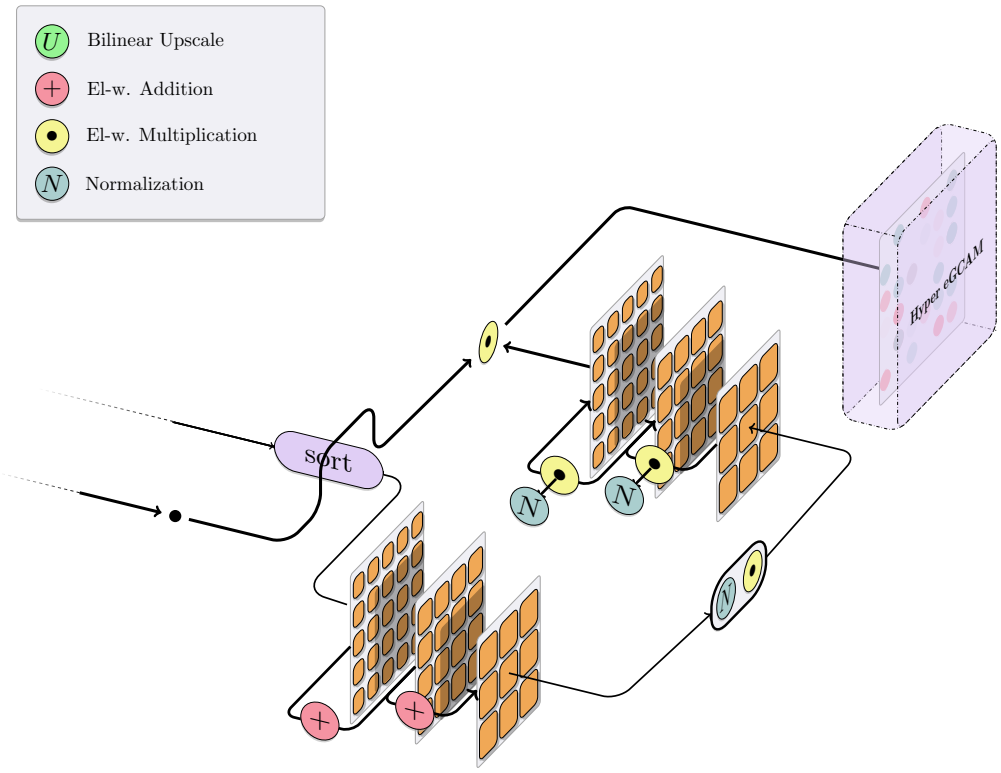


Figure 21: Hyper Expected Grad-CAM - CAMs Accumulation Illustration Extract.

4.8.6 *CAMs Accumulation and Fusion*

As discussed in Section 4.8.2, each stage of the pipeline is fully *vectorized* and parallelizable (*More in Section 6.1*) i. e., it can be executed in parallel with a *DataParallel* on a different GPU or node. This involves that the results of each intermediary stage of the pipeline has to be added and fused to ultimately produce a single saliency. Consequently, such steps are performed at the end, allowing each segment to be executed independently and, if desired, in parallel. All the *supermaps*, one for each stage, obtained from the second step of the pipeline (Section 4.8.4) are sorted by the extraction kernel size, then firstly element-wise added, from *finer-to-coarser* and subsequently normalized; the resulting *supermap*, which contains, a set of point of interests, is normalized and multiplied by the product of each, *coarse-to-finer* map that is, in reverse order. Ultimately, the *Hyper Gradient-weight* CAM is generating by multiplying the *supermap* with the *intermediary map* obtained by the *frequency decomposition* process (Section 4.8.5). Such convoluted process is carried out, because despite the maps obtained from the *frequency decomposition* process are class discriminative i. e., the *atomic details* present are only relevant towards a given class, it is desirable to weight such maps by the *local explanations*.

5

EXPERIMENTS

In this section, a through and detail explanation, alongside the implementation, of the empirical evaluation strategies employed for the assessment and discussion of the reported findings is presented. Each method, as discussed in Section 1.5, is grounded on established and compatible metrics adopted in prior works [52, 12, 58].

5.1 QUANTITATIVE EVALUATION

All quantitative evaluations have been performed on the *ImageNet Large Scale Visual Recognition Challenge* (ILSVRC2012) public dataset for the *image classification tasks*, and on the *Commercial Modular Aero-Propulsion System Simulation* (C-MAPSS) for *time-series* respectively. Explanations with respect to the images have been generated on a pretrained *VGG16*[56] from the *Pyorch model Zoo* backbone and, in accordance with previous work, across randomly selected samples due to the complexity of the validation metrics involved; for this study have been selected 5000 samples within the ILSVRC2012 validation set, which is typically 2.5 to 5 times higher than previous papers. Each sample was normalized with the mean and standard deviation computed on the ILSVRC2012 dataset. The PdM evaluation have been produced on *CNN-LSTM* and *CNN-BiLSTM* on the C-MAPSS *test set* and involved only *IIC*, *AD* and *ADD* metrics, excluding *Insertion* and *Deletion metric* due to the type of data. For comparison the following *CAM* methods have analyzed: *Grad-CAM*[52], *Grad-CAM++*[9], *Smooth Grad-CAM++* [42], *XGrad-CAM*[22], *LayerCAM*[30], *Score-CAM*[66], *HighRes-CAM*[15], *Ablation-CAM*[12], *Poly-CAM*[18]. All methods have been sourced from the popular *TorchCAM* [21] library besides *HiRes-CAM*[15] and *Ablation-CAM*[12] which have been imported from the PyTorch CAM explorer with the most stars on GitHub[23]. Ultimately, all evaluations have been performed on a single GPU (A100-SXM4) setup with the only exception with the performance benchmarks, where a multi-GPU rig is used for comparison. Full details of the test rig have been proposed in Table 1.

5.1.1 *Average confidence*

Granted the notion that if a feature has great importance towards a prediction, then masking such input should greatly affect the model performance [27], we compute a quantitative metric of the average *drop*% [52]. Hence, given a model $f : \mathbb{R}^n \to [0, 1]$, and $x = (x_1, \ldots, x_n) \in$

Table 1: Testing Rig

	Hardware					
	CPU	**Cores**	**RAM**	**GPU**	**VRAM**	**CUDA**
Single	Xeon Gold 5317	12	90GB	A100-SXM4	80GB	v12.0
Multi-GPU	Xeon Gold 5317	12	180GB	**2x**A100-SXM4	40GB	v12.0

$\mathbb{R}^n$ being a set of inputs, then the attributions of those features and $T(pn) : \mathbb{R}^n$ a function with ablates the pn% most important feature, then the $drop$% can be computed as shown in Equation 63. Similarly, the $increase$% can be computed by comparing the ablated output with the original (Equation 64).

$$Drop\% = \frac{1}{N}\sum_{i=1}^{N}\frac{\text{ReLU}(y_i^c - o_i^c)}{y_i^c} \quad where \quad \begin{aligned} y_i^c &:= f(x_i) \\ o_i^c &:= f(x_i \circledcirc T(x_i)) \end{aligned} \tag{63}$$

$$Increase\% = \frac{1}{N}\sum_{i=1}^{N}\text{sign}(o_i^c - y_i^c) \quad where \quad \begin{aligned} y_i^c &:= f(x_i) \\ o_i^c &:= f(x_i \circledcirc T(x_i)) \end{aligned} \tag{64}$$

5.1.2 *Increase ratio*

Reiterating the notion of *ablation*, an accurate map should exhibit consistent results when provided with masked input [12], that is by measuring the ratio at which the model's output increases when only the score map is provided across a set of inputs (Equation 65).

$$increase\ ratio = \frac{1}{N}\sum_{i=1}^{N}\frac{\mathbb{1}(y_i^c < o_i^c)}{N} \quad where \quad \begin{aligned} y_i^c &:= f(x_i) \\ o_i^c &:= f(x_i \circledcirc T(x_i)) \end{aligned} \tag{65}$$

5.1.3 *Insertion and Deletion*

As widely discussed in Section 1.4 and subsequently in Section 1.5, despite the growing interest within the field of XAI, there is still no clear agreement on the appropriate strategy and metric to evaluate and quantify the explainability of DNN [25]. Among all the metrics, *insertion* and *deletion*[43] are the most utilized and predominant within the *visual explanation* methods. *Deletion* measures the decline in the probability w.r.t. the predicted class, as more pixels are progressively removed [43]. The removal follows the pixel-wise feature attribution provided by the supplied saliency map. The *area under the curve* (AUC) of this metric is meant to be minimized. In contrast, *Insertion* assess

the increase in the softmax probability as more pixels are introduced [43], where a higher AUC indicates a more coherent explanation.

5.2 DATASET

With the end goal of evaluating our proposed approach as whole, providing a more comphrensive evaluation that is not domain-specific, we provide evaluation with respect to *image classification* and PdM, demonstrating the general applicability of the proposed methods. Therefore, this study is conducted, and its evaluation carried out, as introduced in Chapter 5, for the PdM *prognosis* problem, using the *NASA* benchmark dataset *Commercial Modular Aero-Propulsion System Simulation* (C-MAPSS) [2] and for the *image classification* tasks on ILSVRC2012[48] public dataset.

5.2.1 *C-MAPSS*

Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) dataset comprises four individual sets, each partitioned into train and test, i. e., $FD001$, $FD002$,$FD003$ and $FD004$ [2], respectively. Each set has been simulated under different combinations of operational conditions and fault modes [2], where each individual multivariate time series corresponds to a separate engine, which is used to simulate sets of homogeneous fleet. In addition, each engine is presented with distinct initial levels of wear, and each sensor data is masked with noise. At the outset of each series (*cycle*), the engine is operating under normal condition, i. e., not in fault state, then subsequently after an undiscloused number of *cycles*, experiences a system failure [2]. In the training set, each engine is run until failure, while in the test set, the target value i. e., RUL is provided. The data comprises *sensor* 1 to *sensor* 26, the number of the unit, the current cycle, and 3 operational settings, which according to Abhinav Saxena et al. have a great effect on the engine performances [2].

5.2.2 *ILSVRC2012*

The *ImageNet Large Scale Visual Recognition Challenge* (ILSVRC2012)[48] is a comprehensive and widely adopted dataset within the field *computer vision* and represents a well-established set to evaluate and benchmark visual explanation methods. It contains more than 1.2 million images, of which $50,000$ within the validation set, spanning across 1000 categories covering the most diverse types of object. In particular, class instances are represented *in-context*, where they are captured in their typical environment, which often includes many other objects from other classes, providing a more realistic test set.

6 RESULTS

In this chapter are presented the results of our proposed methods (Chapter 4), and their comparison with existing and well-established explainers, across the metrics discussed in Section 5.1. The evaluations have been conducted following the procedures discussed in Chapter 5, and the findings, subdivided into qualitative and quantitative, are shown below.

6.1 QUANTITATIVE EVALUATIONS

6.1.1 *Model Training*

As discussed in Section 1.1, the focus of this paper is *explainability* and *interpretability* techniques, therefore PdM results are provided as side remarks towards the general applicability of the described methods. Individual model's performances, with respect to the test set, are not of significant impact, as the interest of this study is towards the relative improvement of each method. Notwithstanding, in the direction of more clarity and reproducibility, quantitative metrics have been proposed for each model in Table 2. The testing has been executed on the C-MAPSS test set (*More in Section 5.2*), while the training has been performed on the testing rig discussed in Section 5.1 using *Lightning Fabric* in conjunction with *LR Finder* and *Batch Size Finder* and *super-convergence* scheduling and training approach[59].

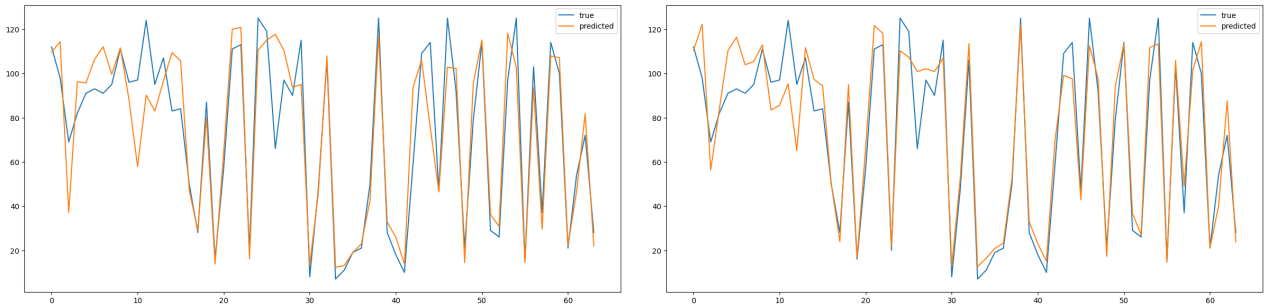
Table 2: Model Training Reults

	CNN-LSTM				CNN-BiLSTM			
	↓ **MSE**[1]	↓ **RMSE**[1]	↓ **MAE**[1]	↓ **MAPE**[1]	↓ **MSE**[1]	↓ **RMSE**[1]	↓ **MAE**[1]	↓ **MAPE**[1]
Test Set	243.19	15.59	11.06	0.19	169.53	13.02	10.04	0.18

Note[1]: Lower is better (↓)

6.1.2 *Faithfulness*

The *faithfulness* of our proposed approaches has been quantitatively evaluated in the way and measures discussed in Section 5.1 and the results tabulated in Table 4. In Figure 23 are plotted all the *insertion* and *deletion* curves for all the 5000 samples with respect to each method. Based on the most relevant and widely adopted metrics, that



(a) CNN-LSTM (b) CNN-BiLSTM

Figure 22: RUL predictions on the test set. Blue line is the ground truth. Orange line plot are the predicted values.

Table 3: C-MAPSS Faithfulness Metrics

	CNN-LSTM		CNN-BiLSTM	
	↑ **Avg. Increase**[1]	↓ **Avg. Drop**[2]	↑ **Avg. Increase**[1]	↓ **Avg. Drop**[2]
Grad-CAM	27.0	31.7	31.1	38.5
Expected Grad-CAM	**35.2**	**34.4**	**34.0**	**40.8**

Note[1]: Higher is better (↑)

Note[2]: Lower is better (↓)

Table 4: Faithfulness Metrics

		VGG16					
	Method	↑ **Ins.**[3]	↓ **Del.**[4]	↑ **Ins-Del**[3]	↑ **Avg. Inc.**[3]	↓ **Avg. Drop**[4]	↑ **ADD**[3]
Gradient	Grad-CAM	0.60	0.09	0.51	0.10	0.27	0.37
	Grad-CAM++	0.58	0.10	0.49	0.08	0.29	0.36
	Smooth Grad-CAM++	0.44	0.17	0.27	0.04	0.37	0.27
	XGrad-CAM	0.62	0.09	0.53	0.11	**0.26**	0.38
	HiRes-CAM	0.57	0.10	0.47	0.07	0.31	0.34
	LayerCAM	0.57	0.10	0.47	0.08	0.27	0.37
	Score-CAM	0.56	0.11	0.46	0.06	0.29	0.34
	Ablation-CAM	0.57	0.10	0.48	0.07	0.31	0.34
	Poly-CAM±	0.68	**0.07**	0.61	0.06	0.57	0.37
Ours	Expected Grad-CAM[1]	0.65	0.09	0.56	0.11	0.30	0.41
	Expected Grad-CAM++[1]	0.64	0.10	0.54	0.11	0.29	0.40
	Hyper Expected Grad-CAM[2]	**0.83**	0.11	**0.72**	**0.14**	0.27	**0.48**

Note[1]: Computed using 50 draws (iterations) per explanation

Note[2]: Computed using 50 draws (iterations) per explanation and bidirectional modulation set to 0.95

Note[3]: Higher is better (↑)

Note[4]: Lower is better (↓)

is, *insertion* and *deletion*, our proposed method Hyper Expected Grad-CAM outperformed every other other prior work, providing a substantial 0.15 improvement compared to *Poly-CAM*[18] which was the second best performing method according to this metric. Our *gradient-safe* replacement i. e., Expected Grad-CAM and its plus-plus variation (Expected Grad-CAM++), exceeds its *gradient*, and non, counterparts, resulting the third and *fourth*-best method, respectively. According to the *deletion* metric, *Poly-CAM*[18] provided the best results, followed by Expected Grad-CAM which provided a marginally lower score (0.02). The *Insertion-Deletion* metric, introduced with *Poly-CAM*[18], Hyper Expected Grad-CAM produced the best result, with a remarkable score of 0.72, followed by *Poly-CAM* with 0.61. When comparing local faithfulness using less robust and least recent metrics, Hyper Expected Grad-CAM confirmed the results surpassing every other method, including Expected Grad-CAM in 2 out of 3 metrics i. e., *average increase in confidence* and *average drop in deletion*. A similar scenario is observed on the C-MAPSS set (Table 3) with Expected Grad-CAM outperforming the original formulation. When comparing Hyper Expected Grad-CAM on the ILSVRC2012 dataset w.r.t. , the last metric *average drop*, it produced the second-best score (0.27), with only 0.1 difference from the *XGrad-CAM*[22] which provided the best score (0.26). Notably, *Smooth Grad-CAM++*[42] yield the worst scores overall, with significant inferior metrics when considering both *insertion* and *deletion*. This is of particular interest, as previously discussed in Section 1.1, *Smooth Grad-CAM++*[42] in conjuction with *Integrated Grad-CAM*[50] have steered prior works in the field away from employing *difference-from-reference* augmentations such as IG and EG in CAM-based methods as deemed to not perform well, which was also confirmed by our results (Table 4). Our proposition, Expected Grad-CAM, on the other hand, which implements a similar *type* but different augmentation (*More in Section 4.3*), showed a different picture (Table 4), proving the effectiveness of *distribution sampling difference-from-reference* methods even within the context of CAM. Moreover, Expected Grad-CAM exceeded our expectations, as previously discussed in Section 4.3, this method is not intended as *another variation* of Grad-CAM nor CAM, but rather an *inplace* replacement which does not suffer, or at least in less measure, from the *gradient* issue of the original method. In this sense, it significantly outperformed the original method Grad-CAM and Grad-CAM++ (Table 4) as well as its direct competitor *Smooth Grad-CAM++*[42] producing remarkably higher scores across every analyzed metric. Furthermore, it placed *second* or *third*-best method in 4 metrics out of 6 even when compared to more complex methods (Table 4) showing the degree and relevance of the *gradient issue* within CAM methods. Hyper Expected Grad-CAM, predictably, yield higher score metrics than our base proposition Expected Grad-CAM in every test besides *deletion*; this is hardly surprising given that it is based on *multiple* multi-stage

phases which involve Expected Grad-CAM. On the other hand, it is extremely remarkable the extent and degree of the increase in metric's score yielded by our proposition, especially when considering the *insertion* parameter, which scored 0.15 higher than previous methods. Despite the *deletion* metric is not as favorable as other metrics, ending just in line with prior approaches, it is expected, as within such metrics there is known to exist a trade-off between *insertion* and *deletion* performances, where often the increase of one results in the decrease of the other [45]; however, it is striking how, in spite of such an increase in the *insertion* metrics, our proposition was able to retain an adequate *deletion* score. Ultimately, Expected Grad-CAM, and its variant Expected Grad-CAM++, provided higher-than-expected results, outperforming not only their original counterparts and original method, but provided better quantitative scores than most, more advanced and modern methods. Hyper Expected Grad-CAM demonstrated impressive capabilities, outclassing prior methods in 5 out of 6 metrics, when considering *average drop* within tolerance, proving high capabilities of identifying relevant portion of the image. This is remarkable if considering the Hyper Expected Grad-CAM was not designed with such metrics in mind. The metrics presented in this section, as pointed out from previous work and widely discussed in section Section 1.5 and subsequently in Section 5.1, have not to be taken as the end-of-all or a definitive answer towards *faithfulness*. As discussed in Section 4.7.4, *faithfulness* strongly relies on the notion of *importance* and *relevancy*, which are both very contentious concepts within the field XAI. Moreover, these metrics don't quantify *faithfulness* according to our notion (*More in Section 4.7.4*), nor evaluate the *natural encoding*, but, more specifically they don't provide any measure of *interpretability*, which arguably is equally important in *visual explanations*.

6.1.3 *Convergence*

In accordance with the implementation of *mini-batching*, discussed in Section 4.6.2, by inspecting the results (Table 5) obtained by computing the quantitative metrics using Expected Grad-CAM with different number of draws is possible to observe a similar level of performances despite the different number of samples. When running Expected Grad-CAM with half the number of samples but using *mini-batching* in $5x5$ configuration, is possible to obtain performances equivalent to running it with twice (**2x**) the number of samples within a single batch (Table 5). Moreover, by inspecting the *cumulative ingredients*, when using mini-batching in configuration $5x6$, is possible to achieve a *converge-difference* equivalent to over 160 traditional samples/iterations (Figure 24). As discussed in Section 4.6.2, *mini-batches*, in this context, does not refer to any the practical detail, nor multiple parallel sample computation, but rather the aggregation and averaging of

multiple *interpolator function's* paths, which lead to higher robustness and faster convergence (*More in Section 4.6.2*).

Table 5: Faithfulness Metrics - Expected Grad-CAM N.Draws Comparison

		VGG16						
	Method	**N. Draws**	**↑ Ins.**[2]	**↓ Del.**[3]	**↑ Ins-Del**[2]	**↑ Avg. Inc.**[2]	**↓ Avg. Drop**[3]	**↑ ADD**[2]
Ours	Expected Grad-CAM[1]	5x5	0.64	0.09	0.55	0.10	0.29	0.39
	Expected Grad-CAM	50x1	**0.65**	**0.09**	**0.56**	**0.11**	**0.30**	**0.41**

Note[1]: Computed using *mini-batching* (More in Section 4.6.2) with 5 mini-batch with 5 draws each for a total of 25 draws (iterations) per explanation

6.1.4 *Efficiency*

The remarkable results obtained by Expected Grad-CAM and Hyper Expected Grad-CAM, with respect to the quantitative metric under anal-

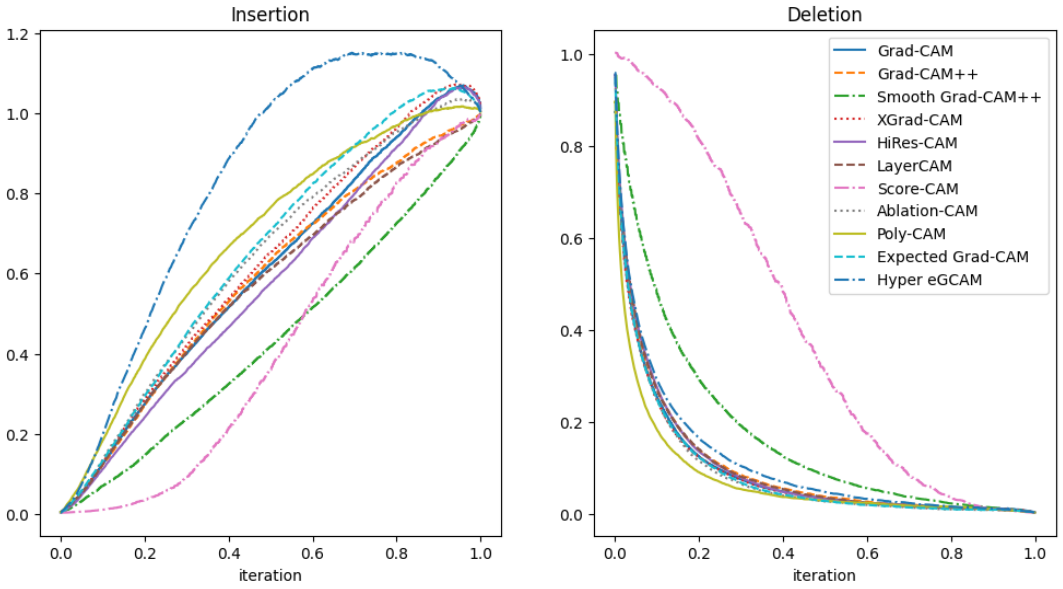


Figure 23: Insertion and Deletion average curves of each method across all the 5000 samples. Mean values across 1003 iterations.

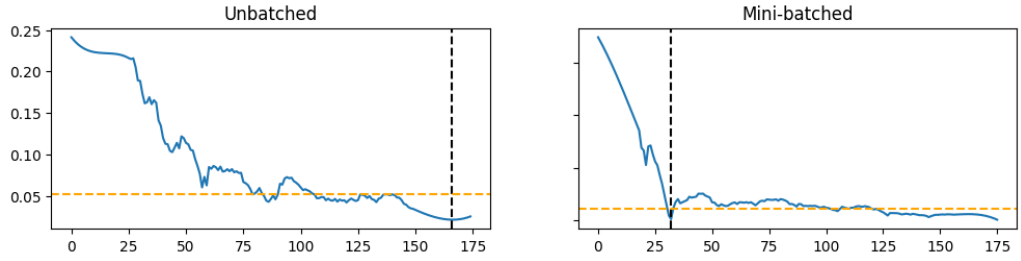


Figure 24: Expected Grad-CAM Convergence different comparison between Unbatched and Batched technique.

ysis, come at the cost of some computational overhead. Because of the intrinsic nature of the method, Expected Grad-CAM, requires multiple draw sampling followed by subsequent interpolations and partial backpropagations w.r.t. the target layer. Our original implementation of Expected Grad-CAM took 3.71s (on average) per explanation when tested across 50 different iterations. When compared to the baseline Grad-CAM, which is the fastest explainer, is *371x* slower. By adopting *dynamic programming* techniques, by precomputing all *alphas* and intermediary interpolator factors, and caching intermediary maps and baselines across batches by using caching, is possible to drastically reduce the execution time of Expected Grad-CAM to 0.26s. Such improvement is quite remarkable and faster than all *non-gradient*-based explainers Table 6. Moreover, our implementation is time-wise comparable with the *Smooth Grad-CAM++* implementation from *TorchCAM*[21], while providing significantly higher scores w.r.t. to quantitative metrics (Section 6.1) showing the quality and effectiveness of the *dynamic programming and caching* strategies employed. Ultimately, Hyper Expected Grad-CAM, as largely discussed in Section 4.8 involves *multiple multi-stage* phases which each requires multiple passes of Expected Grad-CAM; this results in significant computational overhead and higher execution times. By following the original implementation, with no optimizations, the average execution time is of 8.92s, which is quite high. By caching intermediary maps in between stages

Table 6: Efficiency and Time-related performances

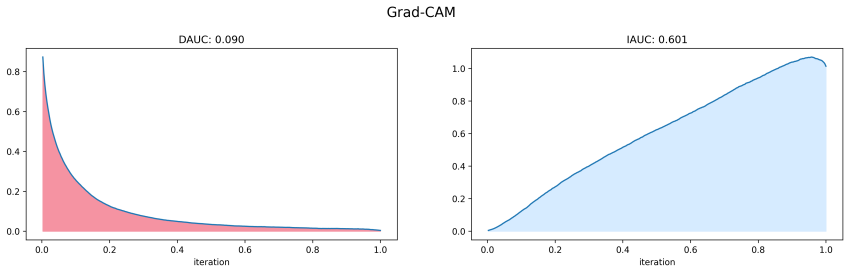
			VGG16	
	Method	**Explainer Params**	**Iterations**	**↓ Avg. Execution Time[1] (s)**
	Grad-CAM	*default*	50	0.010
	Grad-CAM++	*default*	50	0.010
	Smooth Grad-CAM++	*default*	50	0.261
	XGrad-CAM	*default*	50	0.010
	HiRes-CAM	*default*	50	0.010
	Score-CAM	*default*	50	0.274
	Ablation-CAM	*default*	50	0.313
	Poly-CAM±	*default*	50	3.657
Ours	Expected Grad-CAM	*default*	50	3.711
Ours	Expected Grad-CAM	*DP&Caching*	50	0.260
Ours	Hyper Expected Grad-CAM	*default, Single*[2]	50	8.921
Ours	Hyper Expected Grad-CAM	*Caching, Single*[2]	50	1.727
Ours	Hyper Expected Grad-CAM	*Caching, Dual*[3]	50	0.639

Note[1]: Lower is better (↓)

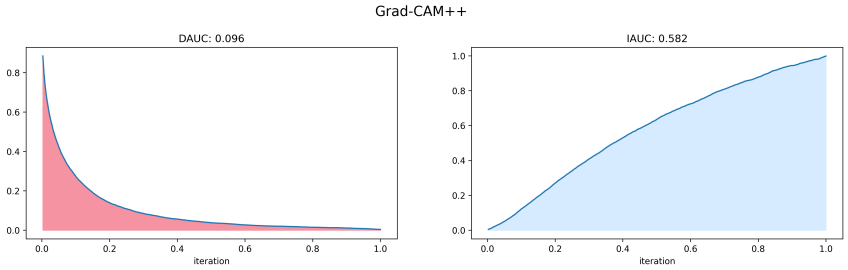
Note[2]: Using single GPU setup (Table 1)

Note[3]: Using dual GPU setup with pipeline parallelization (Table 1)

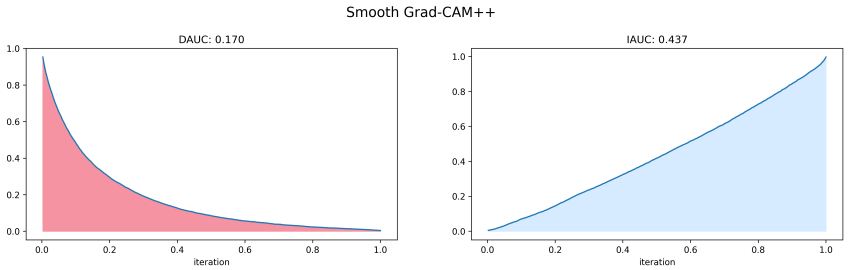
Figure 18 is possible to reduce the computation time per explanation to 1.73s which is acceptable, and faster than almost any *non-gradient based* method. As discussed in Section 4.8.2, Hyper Expected Grad-CAM has been designed to be parallelizable and moved across multiple GPU and nodes; by using the dual GPU configuration (Table 1) is possible to reduce the average execution time to 0.64s.



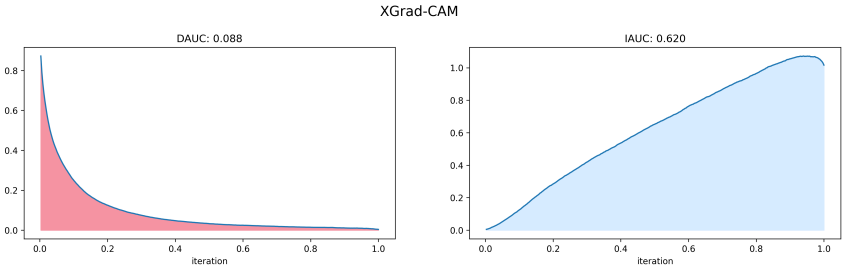
(a) Grad-CAM



(b) Grad-CAM++



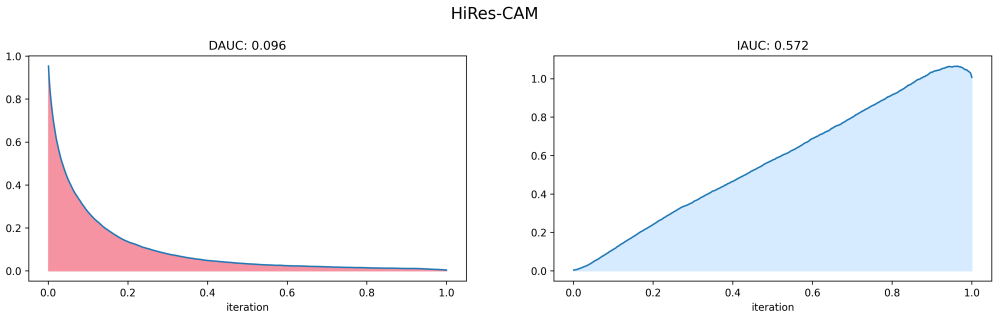
(c) Smooth Grad-CAM++



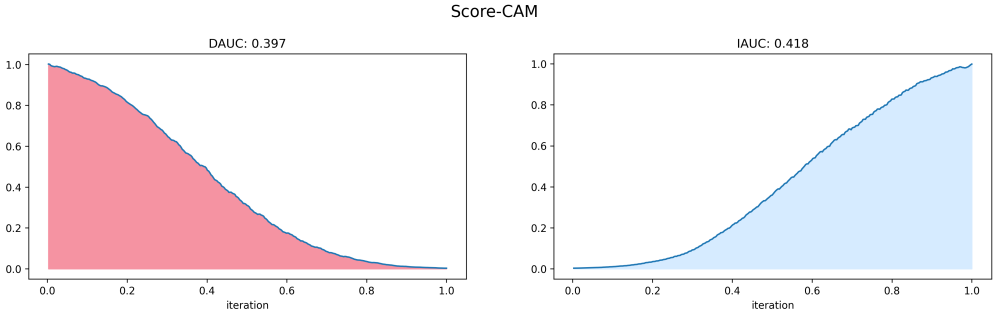
(d) XGrad-CAM

Figure 25: Individual Insertion and Deletion curves of the baseline methods *Grad-CAM*[52], *Grad-CAM++*[10], *Smooth Grad-CAM*[42] and *XGrad-CAM*[22]

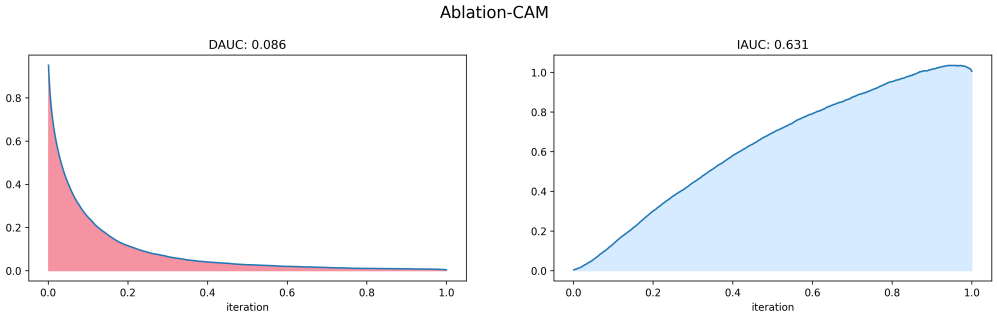
.



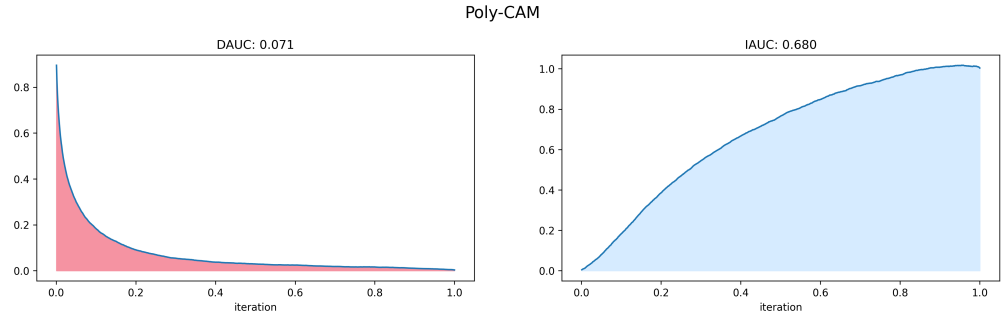
(a) HiRes-CAM



(b) Score-CAM



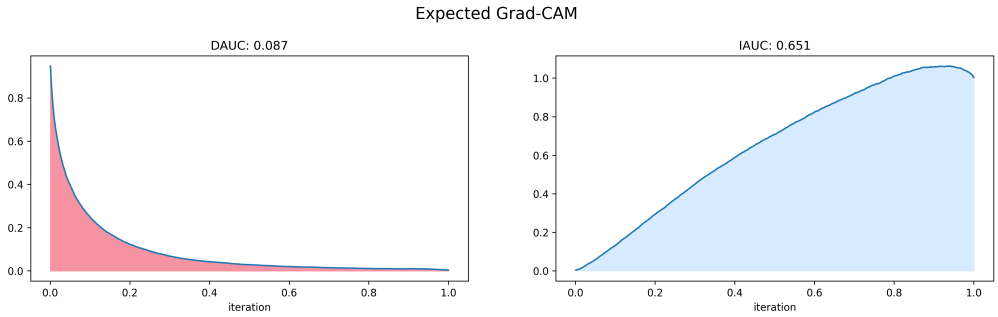
(c) Ablation-CAM



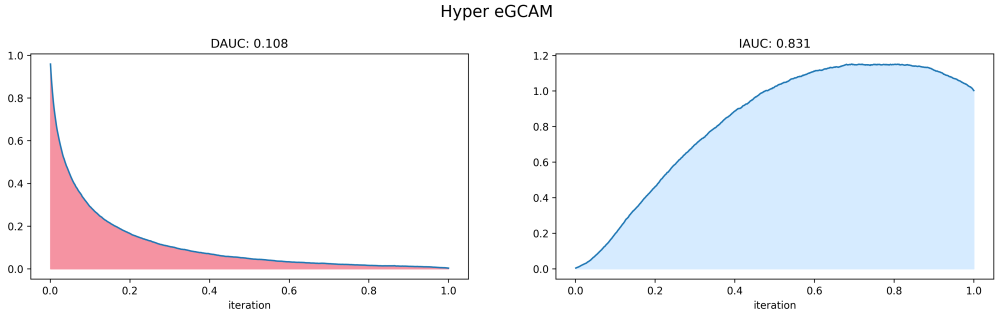
(d) Poly-CAM+

Figure 26: Individual Insertion and Deletion curves of the baseline methods *HiRes-CAM*[15], *Score-CAM*[66], *Ablation-CAM*[12] and *Poly-CAM*±[18]

.



(a) Expected Grad-CAM



(b) Hyper Expected Grad-CAM

Figure 27: Individual Insertion and Deletion curves of our proposed methods Expected Grad-CAM and Hyper Expected Grad-CAM

6.2 QUALITATIVE VISUAL ASSESSMENT

6.2.1 *Expected Grad-CAM is a Gradient-safe Grad-CAM Replacement*

Building on our formulation of Expected Grad-CAM and its variant Expected Grad-CAM++, and following the quantitative results obtained in Section 5.1, it is clear that not only fulfilled its expectations, but exceeded them. As widely discussed in Section 4.3, and subsequently confirmed by the results obtained in Section 6.1.2, Expected Grad-CAM has been developed as an *inplace* full replacement of Grad-CAM, which in contrast with the original method, does not suffer from *gradient-issue*. This implies that Expected Grad-CAM performs comparably to the original proposition, whereas the gradients are neither saturating the output nor vanishing, but perform better in all these scenarios where such conditions are not met i. e., where the original method does not work correctly. However, even when the original method operates correctly, Expected Grad-CAM, in practice, due to the averaging of multiple interpolated paths, given by the *dfp* (*More in Section 4.3.5*), often results in more localized maps (Figure 28) and generally less noisy (Figure 29). The dissimilarity in saliencies occurs when *gradients* issues arise: when gradient saturate, typically results in maps, in the

original method, which do not fully cover the relevant portion of the image (Figure 30), or, they can completely fool the explainer as the *rate of change* diminishes, by classifying unimportant portion of the image (Figure 32). As such effects can occur in any combination and degree, so are the significant improvements yielded by our proposition Expected Grad-CAM, which when such causes compounds Expected Grad-CAM implicitly alleviate the issue of Grad-CAM-based methods, which struggle with multiple instances of the same class within the same image (Figure 31) without using any *plus-plus* strategy. The Expected Grad-CAM++, as presented in Section 4.4, augments our proposition of Expected Grad-CAM with the *plus-plus*[9] as some previous works have used it [42], therefore we decided to propose a *gradient-safe* version of it, namely Expected Grad-CAM++. However, the results are in the best-case scenario comparable to our base proposition. Guided Expected Grad-CAM has also been proposed for completeness and in accordance with the original paper for a full *gradient-safe* replacement. However, as in the original work, and broadly discussed in Section 4.5, Guided Expected Grad-CAM, is solely the fusion of our proposition Expected Grad-CAM with a *gradient-based* method w.r.t. the input to produce a *class-discriminative* and "*high-resolution*", intended as in the original work, map.

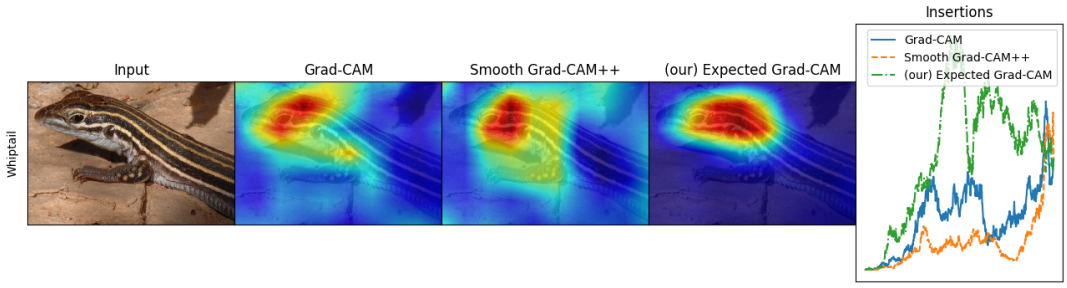


Figure 28: *Grad-CAM*[52], *Smooth Grad-CAM++*[42] and Expected Grad-CAM Comparison with accessory insertion plot

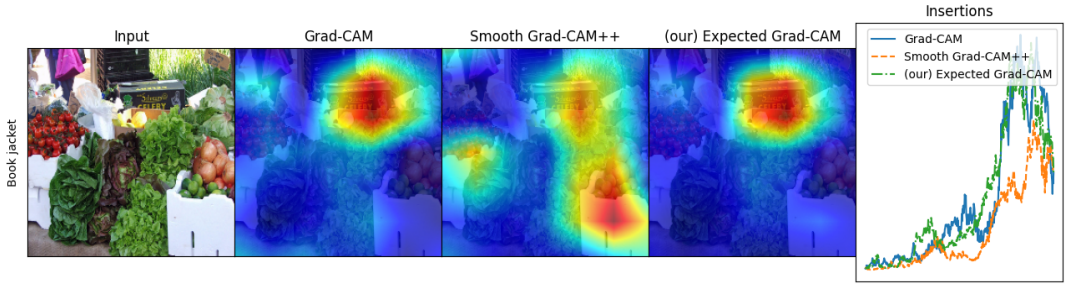


Figure 29: *Grad-CAM*[52], *Smooth Grad-CAM++*[42] and Expected Grad-CAM Comparison with accessory insertion plot

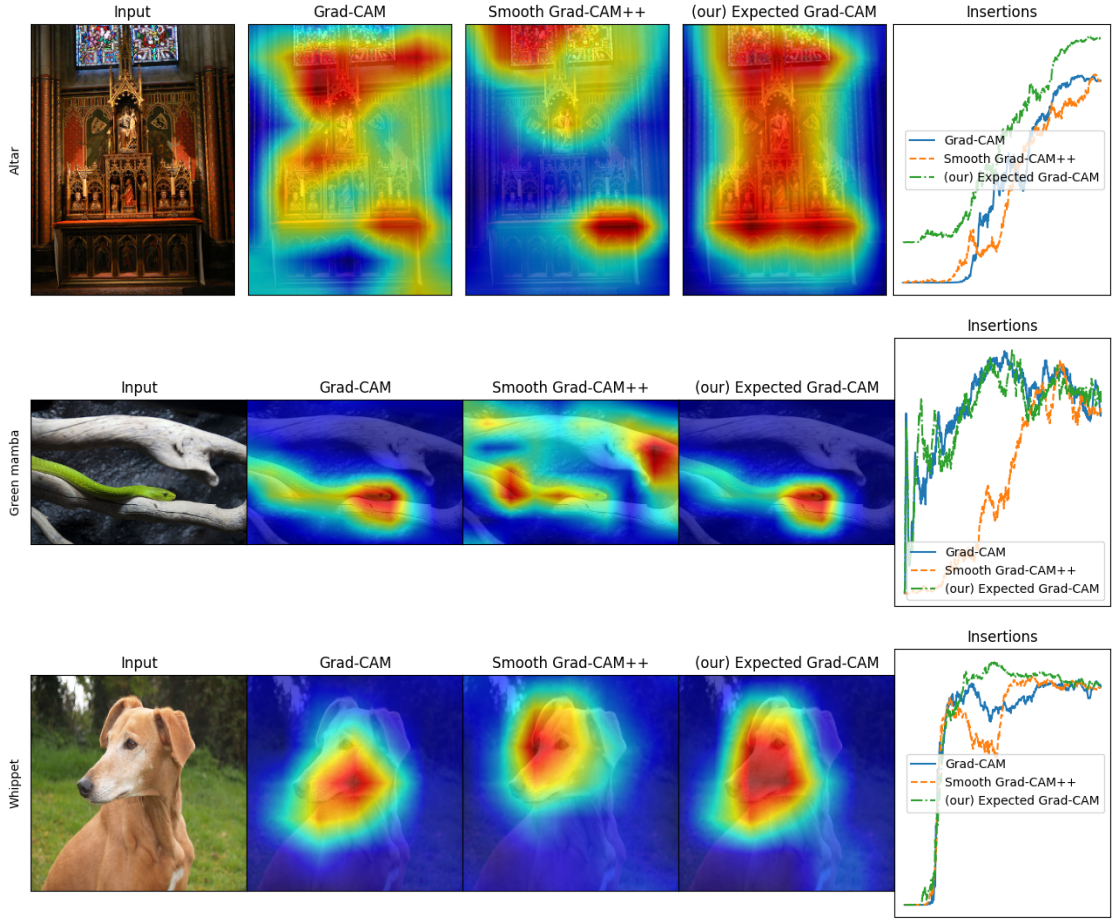


Figure 30: *Grad-CAM*[52], *Smooth Grad-CAM++*[42] and Expected Grad-CAM Comparison with accessory insertion plot

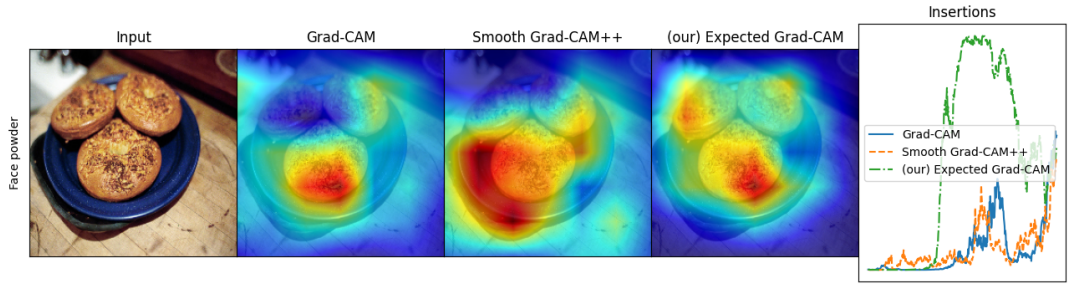


Figure 31: *Grad-CAM*[52], *Smooth Grad-CAM++*[42] and Expected Grad-CAM Comparison with accessory insertion plot

6.2.2 *Hyper Expected Grad-CAM*

In this section are presented qualitative evaluations of the saliencies generated by Hyper Expected Grad-CAM, while discussing some of the notions and properties that our methods satisfy. As previously stated, despite Hyper Expected Grad-CAM yielded remarkable results in quantitative metrics, greatly outperforming current SoA methods within XAI,

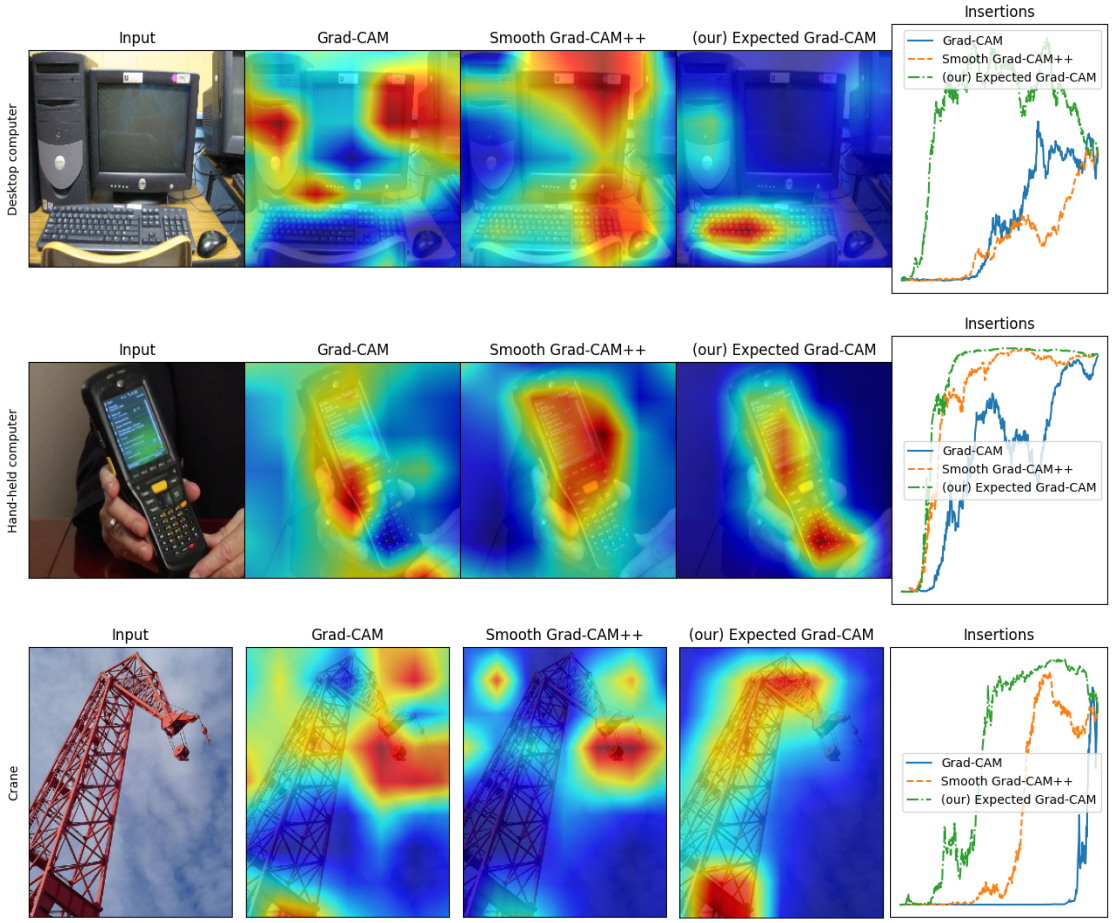


Figure 32: *Grad-CAM*[52], *Smooth Grad-CAM++*[42] and Expected Grad-CAM Comparison with accessory insertion plot

is also extremely important to visually assess the resulting maps to address the limitation of the former metrics. Moreover, is essential to emphasize that in addition to the limitation of such metrics, widely discussed in Section 1.5, current quantitative evaluation, proposed in the previous section, do not cover the extent and cohesive increase of information and interpretability provided by Hyper Expected Grad-CAM. Since these metrics do not provide a quantifying proxy value for *interpretability*, nor evaluate our notion of *faithfulness* or *natural encoding*, they cannot effectively measure the *expressivity* of the saliencies generated by our method. For these, and the reasons previously discussed, visual assessment is a core part of saliency evaluation. In the next sections are proposed a set of saliencies, each compared to the currently highest scoring explainer: *PolyCAM*[18], a non-*gradient* CAM method aimed at producing *high resolution* saliencies.

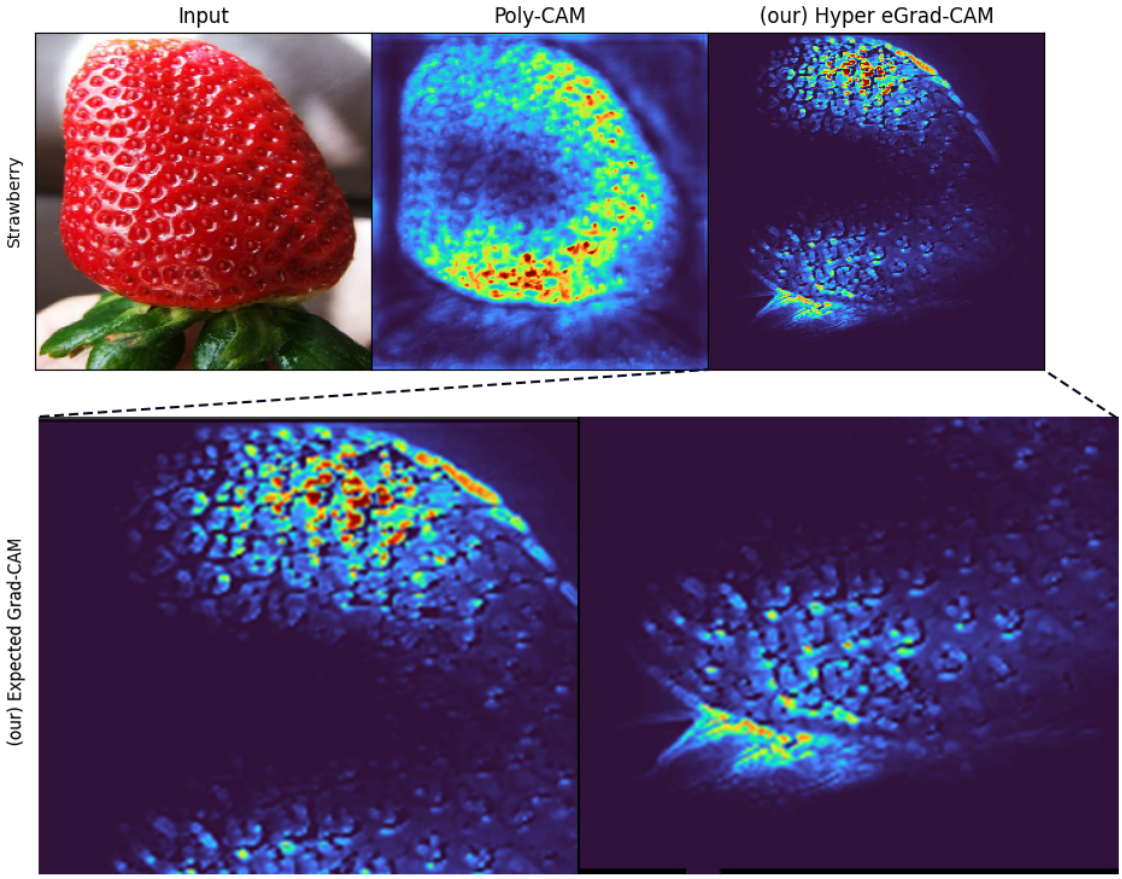


Figure 33: Side-by-side comparison of resolution, clarity, noise and localization difference between *Poly-CAM*±[18], and Hyper Expected Grad-CAM. Second row provides a zoom on the *atomic details* encoding.

6.2.2.1 *Resolution as atomic details*

Hyper Expected Grad-CAM produces a hybrid type of saliencies which are conditioned by the progressive build-up of the model's understanding of the input up to selected target layer. This, as a result of multiple multi-stage CAM-based refinements, creates maps that are both *class-discriminative* and capture *fine-details* without involving or fusing external methods, otherwise used in *guided*-like approaches. Therefore, our approach is not the product of a combination of a "*high-resolution*" method such as IG or EG, masked by a *class-discriminative* map, as discussed in Section 3.5, but it is a natural *class-discriminative* reconstruction of the model's understanding, where each reconstruction is itself *class-discriminative* and gradient weighted. Satisfying our notion of *faithfulness* (*More in Section 4.7.4*) and *resolution* (*More in Section 4.8.5*) it does not produce *high-resolution* maps as prior works aimed at; by formalizing resolution as the degree at which you can discern the individual make-up of an image, then, for any given

layer to satisfy model's *faithfulness*, according to our notion, there exists only one resolution, the one at which the model operates at any given layer. This implies that each saliency is conditioned by the progressive build-up of the model's understanding, that is, the map has only one resolution as it is composed of the individual atomic details at each spatial location. This reformulation of the problem statement **produces saliencies with unprecedented levels of "*resolution*" and *faithfulness*** (Figure 35) as we are not concerned anymore about the arbitrary number of pixels within the saliency, but instead, to extract all the conditioned *atomic details* (Figure 33) that the model has encoded up to any given layer. This implicitly produces *(I) more expressive* and *(II) more faithful saliencies*. The former is the result of our approach satisfying the *natural enconding*, that is, the generated saliencies are encoded using the basic building blocks that make up the *human visual system* i. e., *edges, lines, angles* and *textures*. Prior methods, and with respect to the input, such as *Gradient x Input*, provides saliencies maps where feature importances are encoded as a set of sparse circular activation which, if any, fail to represent or distinguish specific model's intentions i. e., if the model is looking at the edge of an instance, its shape (contour) or the texture of a given surface. Hence, such maps provide a **mono-dimensional encoding of the feature importance**, where the **attribution relevancy** is encoded by a single-valued quantity typically expressed as the intensity of the pixel. Hyper Expected Grad-CAM, on the other hand, provides **multi-dimensional** explanations where not only the importance is encoded, but also the characteristic of it, which **reflects the true intention, and contextual encoding**, of the model up to any given layer. Specifically, it is guaranteed *our notion of faithfulness*, which ensures that the **resulting map is composed only of *atomic details* that the model has learned and encoded** up to the target layer, which concretely, are extracted during the *frequency decomposition* step (*More in Section 4.8.5*). Ultimately, is necessary to explicitly state one **key consequence** of our *notion of resolution*; prior works aimed at increase the **spatial resolution**, as intended as the number of pixels, of the CAM to produce more accurate, localized and less noisy maps. Intuitively, if you are attempting to more accurately visualize a circle, given any discrete number of points around a radius r, increasing the point's count, given a discrete space set, directly defines the *shape roundness*. In a similar fashion, previous approaches, which work on this directions, are deemed to break faithfulness. In this sense, *guided*-like approaches, according to our *notion of faithfulness* and *constraints*, break *faithfulness* as they represent feature with a level of resolution i. e., the roundness in the example above which are not the one employed by the model when producing an inference, as the model is originally working with a much lower dimension space and is therefore unable to discern with such level of details the features. This in turns **breaks interpretabil-**

ity, because as human, wrongfully assume that the model has been utilized a certain feature, while, at that layer, **such abstraction might not even been encoded**. Such distinction is not possible within current methods, and certainly not using *guided*-like methods, which, inheretly, due to element-wise fusion with the input, **produce maps which encodes features of arbitrarily abstraction** and level of detail which are not **guaranteed to be encoded by the model**. Our notion of *natural encoding* ensures such properties, and guarantee this matching between the layerwise model's understanding and its representation. Figure 35 shown the comparative increase of quality and informative power of our method when compared to *PolyCAM*[18].

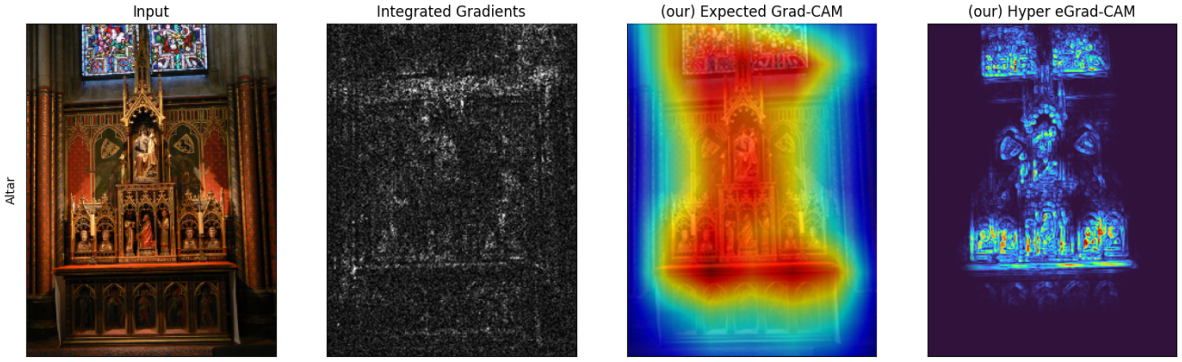


Figure 34: Side-by-side comparison of *Integrated Gradients*[63], Expected Grad-CAM and Hyper Expected Grad-CAM

6.2.3 *Localization, Noise and Clarity*

Hyper Expected Grad-CAM generates CAM which do not just highlight relevant portions of the image w.r.t. to a given class, outlining where the model's focus mostly resides, but also pinpoint with extremely accuracy, which features of the images are used towards such given prediction. In Figure 34 Expected Grad-CAM highlights relevant portions of the image that are part of the *altar* class, but cannot determinate, out of these parts, which is used to classify the image as an altar i. e., which features, given a sample, make up the *altar* class. Localization and absence of noise are extremely desirable properties within visual explanations. Hyper Expected Grad-CAM shows extremely high localization and exceptional resistance to noise due to the multiple refinements, producing ultra sharp saliencies (Figure 36). The localization capabilities, clarity, and amount of details (more on that in the next section) are unparallel compared to current methods.

6.2.3.1 *Contextual Dreaming*

Reconnecting to what discussed in the previous section, different XAI techniques (*More in Section 3.2.2*) allow to inspect the hidden layer

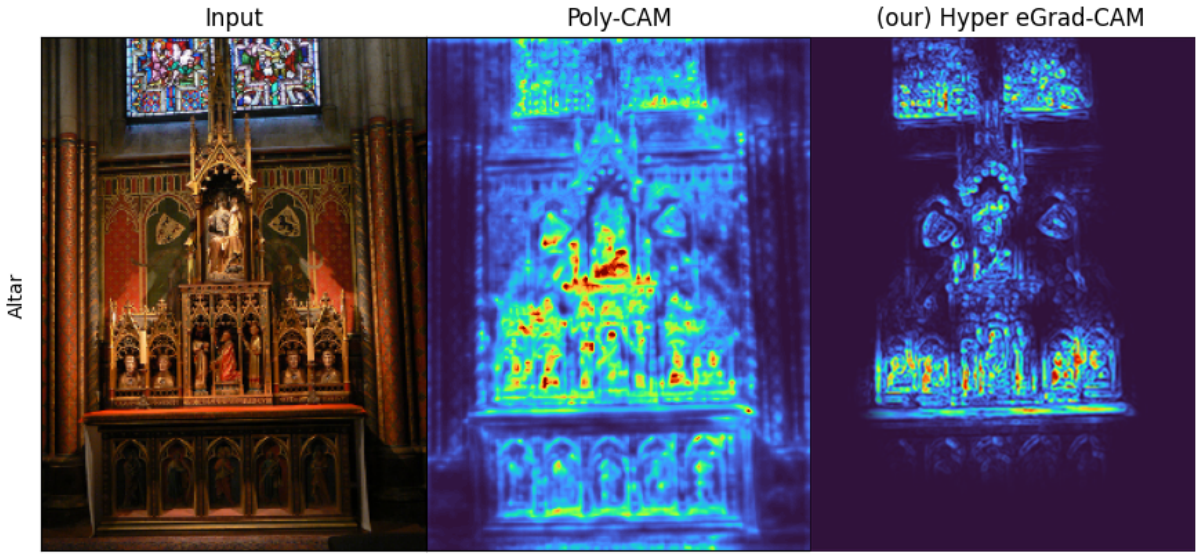


Figure 35: Side-by-side comparison of resolution, clarity, noise and localization difference between *Poly-CAM*±[18], and Hyper Expected Grad-CAM

of the network to generate, by maximization, the class-wise feature representation of relevant encoding, of different abstractions, that the model has learned i. e., what the model is dreaming, layer-wise, when thinking to a specific class. This produces layer-wise explanations of feature detectors w.r.t. a given class. In this sense, as a side effect of our notion of *natural encoding*, Hyper Expected Grad-CAM can be used to provide contextual information about the layer-wise explanation for a given class. That is, if we would like to understand and visualize the types of features that the model uses to draw predictions towards a specific class, we can produce a small *atlas* by producing Hyper Expected Grad-CAM maps for a given subset of samples. For instance, if would like to investigate, contextually, what features make up the class **"zebra"** ("n02391049") we can produce a set of explanations such as the one shown in Figure 38. This not only reveals the type of *high-level* features, as concepts (depending on the target layer), but also the type of underlying feature detectors that fire for each class at different degrees of abstraction. Moreover, returning to the example of the zebra (Figure 38), the model at this layer (features[30]) almost only completely looks at the stripes of the zebra, disregarding any texture. Because of the extremely clean (resistance to noise) map generated by Hyper Expected Grad-CAM, and its unique type of encoding, is possible to differentiate whatever the model is looking at the edge of an object, as part of some details, or its overall shape (contour), or some texture in it. The ability to distinguish, for instance, whatever the model is looking at the shape (contour) of the scales of a snake as opposed to its texture is unprecedented (Figure 36).

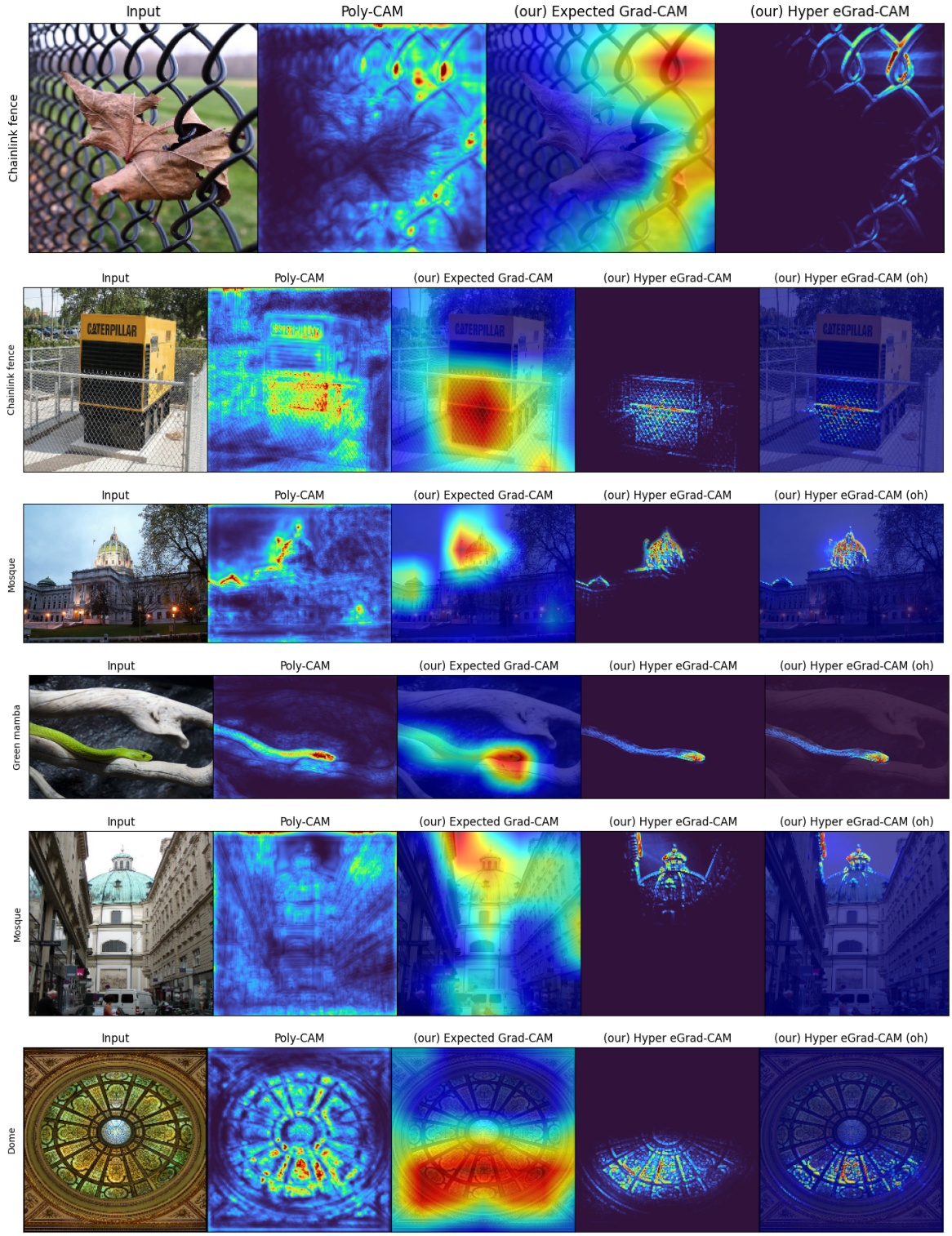


Figure 36: Side-by-side between *Poly-CAM*±[18], Expected Grad-CAM and Hyper Expected Grad-CAM
–Overimposed heatmap is also proposed for Hyper Expected Grad-CAM (**oh**).

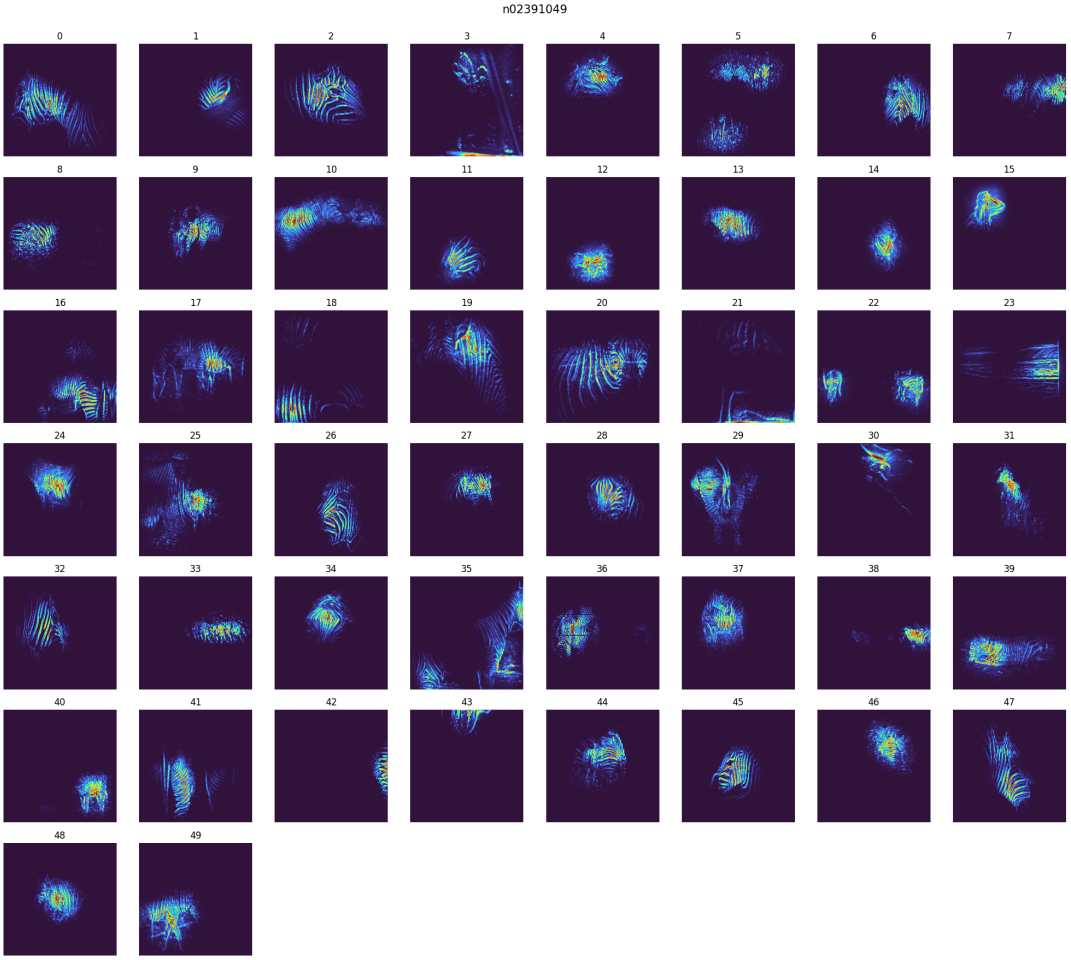


Figure 37: Contextual feature atlas generated using Hyper Expected Grad-CAM for the class "zebra" ("n02391049")

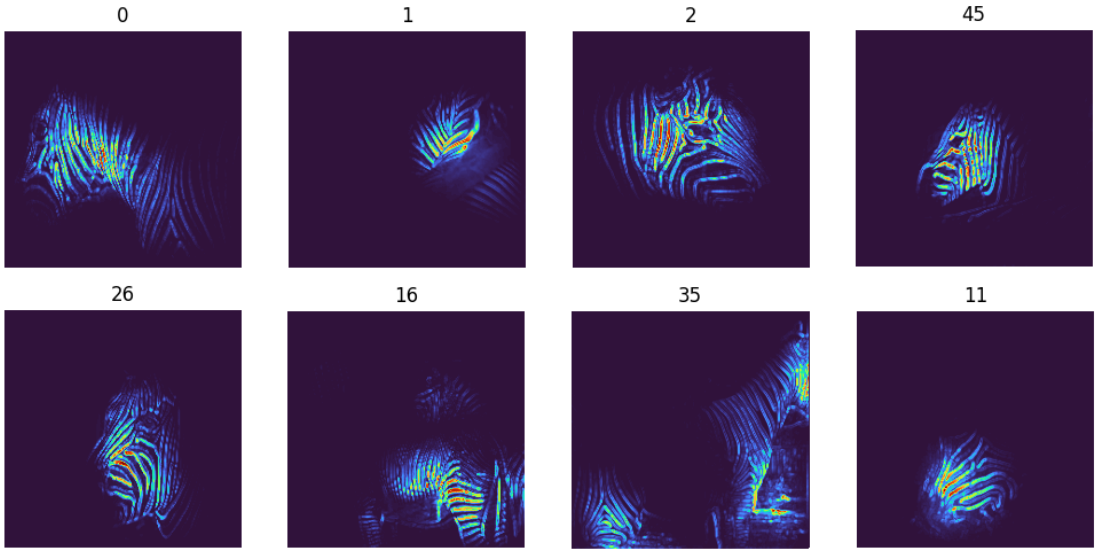
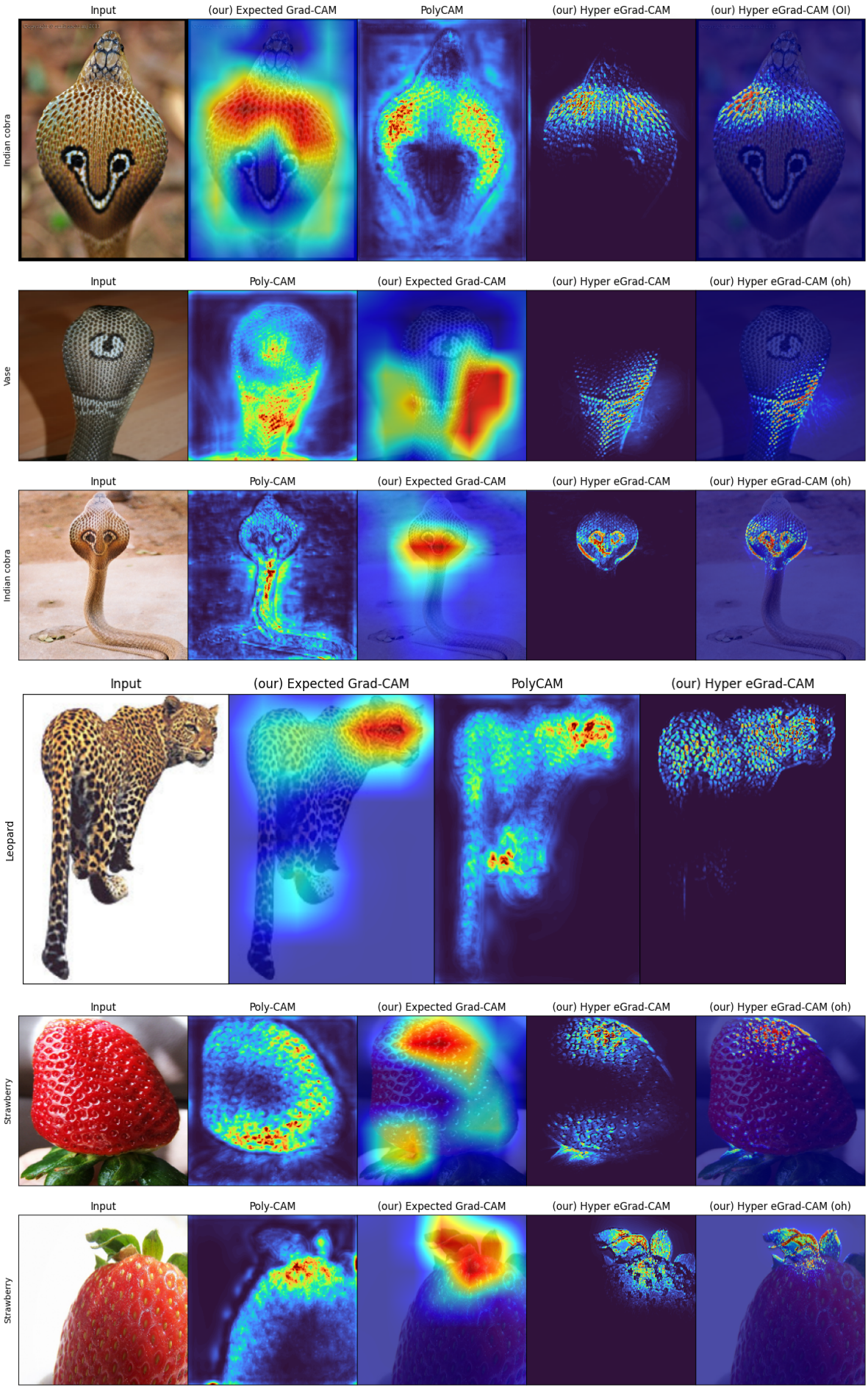


Figure 38: Contextual feature atlas generated using Hyper Expected Grad-CAM for the class "zebra" ("n02391049") - Zoom-In View


Input
(our) Expected Grad-CAM
PolyCAM
(our) Hyper eGrad-CAM
(our) Hyper eGrad-CAM (OI)
Indian cobra
Input
Poly-CAM
(our) Expected Grad-CAM
(our) Hyper eGrad-CAM
(our) Hyper eGrad-CAM (oh)
Vase
Input
Poly-CAM
(our) Expected Grad-CAM
(our) Hyper eGrad-CAM
(our) Hyper eGrad-CAM (oh)
Indian cobra
Input
(our) Expected Grad-CAM
PolyCAM
(our) Hyper eGrad-CAM
Leopard
Input
Poly-CAM
(our) Expected Grad-CAM
(our) Hyper eGrad-CAM
(our) Hyper eGrad-CAM (oh)
Strawberry
Input
Poly-CAM
(our) Expected Grad-CAM
(our) Hyper eGrad-CAM
(our) Hyper eGrad-CAM (oh)
Strawberry

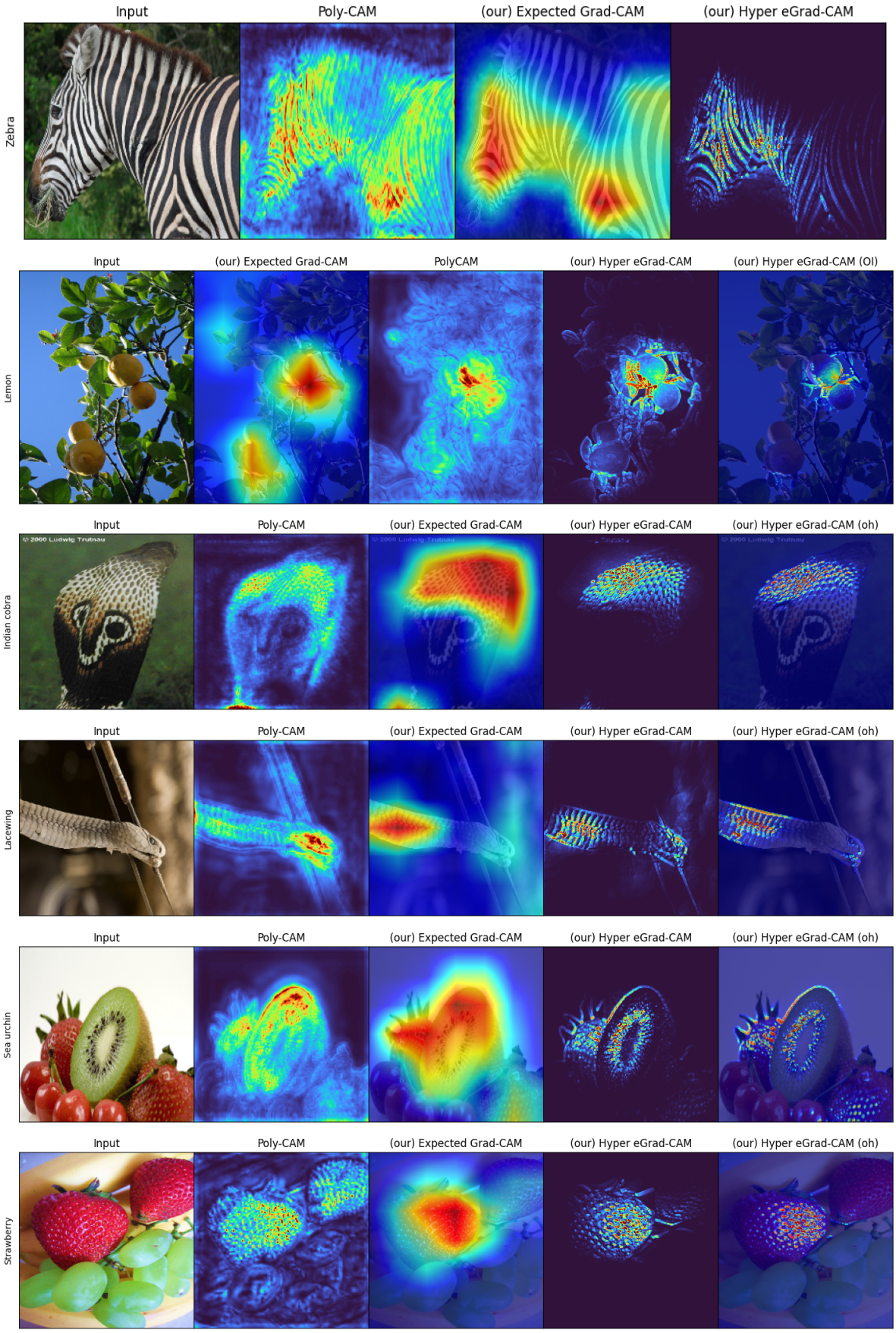


Figure 39: Side-by-side between *PolyCAM*[18], Expected Grad-CAM and Hyper Expected Grad-CAM
- Overimposed heatmap is also proposed for Hyper Expected Grad-CAM (**oh**).

7 CONCLUSION

This thesis deeply delved into the core foundation and underpinning details that corroborate the current state of *visual explanations* within the field of XAI. Our in-depth investigation has been split into two parts, each separately tackling different facets of the contemporary state of *visual explanations* which also inspired the name of this study: *(I) Towards Gradient Faithfulness* and *(II) Beyond Faithfulness*.

In the first segment, we addressed the current gradient-related issues that afflicts not solely Grad-CAM but also every newer technique built on it. Prior methods attempted to solve such relevant issues, by providing different augmentation involving provably *different-from-reference* techniques, but due to their poor performances, have been deemed inadequate within the field of XAI halting every work from been carried out in this direction. As a result, newer publications that have emerged have uniquely focused on *non-gradient* CAM approaches, significantly influencing the industry in recent years. We addressed these issues by developing three novel *gradient-based* CAM formulations, namely Expected Grad-CAM, Expected Grad-CAM++ and Guided Expected Grad-CAM, aimed at replacing traditional Grad-CAM and every newer formulation built upon it, providing a *gradient-safe* backbone explainer. This has been achieved by altering the original gradient computation with a modified and adapted technique, derived from the proven *difference-from-baseline* approach *Expected Gradients* (EG)[20], that involves a *path attribution method* of which baseline is sampled from a distribution. Since our formulation operates on the gradient computation, rather than the recombination and, consequently, use of the partial derivatives as weighting factors of the resulting CAM, our technique is considerably relevant as it allows any existing technique based on Grad-CAM to be rewritten in terms of our method Expected Grad-CAM, providing an immediate, *out-of-the-box* increase in quantitative performances by offering a *gradient-safe backbone*. We validated our findings on a 5 to 10 times larger study, when compared to prior works, on the ILSVRC2012 public dataset. Through a comprehensive comparison of our method against nine of the most recent and performing explainers across six of the most well-established and relevant quantitative metrics, the results exceeded our expectations. Expected Grad-CAM, across all the quantitative metrics, resulted in the *third-best explainer*, when including our approaches, or *second-best* when not counting our more advanced proposition. This is extremely remarkable, considering that, despite Expected Grad-CAM only addresses the gradient-issues present within Grad-CAM, it was able to

outperform newer and more advanced methods both *gradient*-based and *non-gradient*-based techniques. Our comprehensive evaluations shed light on the extent and impact that these gradient-issue have on the explanations, revealing a much larger problem than previously outlined in prior works due to their limited evaluation study. As part of the core aim of this paper is grounded in *real-life* application and industry-driven needs, in addition of the strong theoretical foundations of each method, significant emphasis and effort has been dedicated to enhance our proposition performances, parallelization, and scalability. In an effort to provide, not just theoretically sounding explainers, but concrete and *deployable* solution, we implement a set of optimization spanning from *dynamic programming* techniques to the implementation of caching, providing our method with execution times comparable to existing method within popular and *production-ready* libraries. Notable results considering the amount of work done compared to such methods. Ultimately, by visual assessment, still considered an extremely important evaluation for *visual explanations* within XAI, we confirmed that Expected Grad-CAM and its variants produced similar explanations to the original formulation, when no gradient issues were present, but outperformed the original method when issues arise, ultimately producing more localized and less noisy maps.

In the second part we adopt our own proposition, namely Expected Grad-CAM, to devise a completely new approach that challenges the current state and formulation of *visual explanation* and *faithfulness* as a whole. As *faithfulness* describes the extent to which an explanation accurately reflects the *underlying* DNN mechanics and dynamics involved toward a given prediction; that is, a *faithful* explanation should therefore capture the most relevant and important features that *contributes* to the model prediction. *Importance* and *feature relevance* themeselves are extremely contentious notions within XAI, and are differently defined and encoded with respect to different techniques. Generally, as *faithfulness* describes the scale and degree at which an explanation adheres to the model's inner workings, then the notion of *importance*, or *relevancy*, encodes the conditions at which it occurs. However, neither of these quantities provides a notion on how such a quantity should be encoded or represented. By rethinking faithfulness in terms of our notion of *natural encoding*, by formalizing a set of properties and constraints that an *informative* and *human-interpretable* saliency should respect, we devised a new method, Hyper Expected Grad-CAM. This latest and more advanced proposition is a CAM techniques which produces both *high-resolution*, as intended in prior works, and *class-discriminative* saliencies without fusing other methods. This results in explanation which do not suffer from the issue of typical *high-resolution* methods i. e., with gradients w.r.t. , to the input, nor from the *class-discriminative* CAM-based masking

approaches. This was achieved by generating a new type of *hybrid saliencies* which follows our notion of *faithfulness* and *natural encoding*. In contrast with prior works, Hyper Expected Grad-CAM leveraged two novel ideas which go in the opposite direction of prior works:

- Resolution is not just pixels. *Frequency Decomposition is all you need.*
- Saliencies are not informative nor faithful: they do not follow the *natural encoding*.

By following and implementing the above concepts, our approach is capable of not only generating *class-discriminative* and extremely high resolution maps, according to our notion of resolution (*More in Section 4.8.5*), where each saliencies is the composition of the conditioned progressive build-up of the model's understanding up to any given arbitrary layer, but the feature representation itself follows our notion of *natural encoding*. Conversely to prior methods, where each saliency only encodes the feature importance as a single value, often encoded as the pixel-intensity, that is the map is a composition of sparse circular activation, Hyper Expected Grad-CAM generate saliencies which are composed of the individual make-up of the uncompressed and progressively reconstructed model's understanding, conditioned and gradient-weighted. This implies, that the map is a truthful representation of the underlying model's intent as each "*atomic detail*", within the saliency, is the highest encoding that the model as learned, rather than an arbitrary set of *high-concept* obtained by element-wise multiplying the saliency with the input, as present in current methods. Current *explanation methods* are not designed for *human-interpretability*, and current metrics also do not evaluate *interpretability* either. Ultimately, such methods have been found to be deceptive in their explanations and not truthfully reflecting the model's intentions and understanding of the input as they depict a set of arbitrarily *higher-level concept* which are present within the original image as *relevant*, despite the model's encoding. As a result, they lead to misleading interpretation of such maps, conveying the model is focusing on abstracts that it might not genuinely comprehend as it does not contain feature detectors of such sort. Despite current metrics do not cover nor evaluate interpretability or the extent and cohesive increase of information provided by our proposition, Hyper Expected Grad-CAM, scored remarkable results across each quantitative metrics, yielding a 0.15 increase in insertion, when comparing the *highest scoring* (non-gradient-based) explainer available within the field, and 0.11 when consider *insertion-deletion*.

7.1 RESEARCH QUESTIONS ANSWERS

In this section all the gathered findings are summarized in regards to the research questions initially formulated.

RQ1. To what extent does the original formulation of Grad-CAM suffers from saturating and vanishing gradients and can a *gradient-based* CAM method be formulated that does not suffer from such limitations?

By conducting a 5 to 10 times larger evaluation study with a up to 5 times more iterations (w.r.t. to *Ins/Del* metrics), when compared to previous works, was possible to establish that the gradient issues that affect Grad-CAM, and therefore any approach built upon it, are more widespread and of greater extent than previous papers have outlined. We proposed three novel *gradient-based CAM* formulations, namely Expected Grad-CAM, Expected Grad-CAM++ and Guided Expected Grad-CAM which tackle and address intricate and persistent *vanishing* and *saturating* gradient problems. This was achieved by reshaping the conventional gradient computation by incorporating a customized and adapted technique inspired by the well established and provably *Expected Gradients*'s *difference-from-reference* approach.

RQ2: Is it possible to create a pure *gradient-based* CAM technique which offers *high-resolution* and *class-discriminative* explanations without combining any other method?

In the second segment of our thesis we built upon our prior proposition (Expected Grad-CAM) and devise a novel *CAM* method that produces both *high-resolution* and *class-discriminative* explanation without fusing other methods, while addressing the issues of both *gradient* and *CAM* methods altogether. Hyper Expected Grad-CAM, by challenging the current state and formulation of *visual explanation* and *faithfulness* produces a new type of *hybrid* saliencies which satisfy the notion of *natural encoding* and *perceived resolution*. By rethinking *faithfulness* and *resolution* is possible to generate saliencies which are more *detailed, localized* and *less noisy*, but most importantly that are composed of only concepts that are encoded by the *layerwise* models' understanding.

7.2 FUTURE WORK

Due to the limitations and scope of this thesis, as well as the novelty of the proposed approaches, much work can be done in many different directions. As discussed in previous section, existing gradient-based CAM methods can be reformulated in terms of Expected Grad-CAM, potentially contesting current SoA explainer. Regarding Hyper Expected

Grad-CAM, as discussed in their relative sections, due to the novelty of the approach and the notions it introduces, it is an uncharted territory, and consequently many parts have not been adequately explored or optimized. For instance, the intermediary layer selection, during *frequency decomposition* is selected using a $1 + 1$ policy, however, there could be better strategies that can exploit a larger pool of network information. For our approach, we used only a single *filter* per step; however, by evaluating our results, an improvement could be dictated by applying a secondary *band-pass-filter* such as a *Gabor filters* widely used in *texture analysis*. The selection of the number of stages can also be improved, using a different metric rather than IoU, which could potentially yield better or the same performances, but with a lower number of stages. At each stage of the *feature dependency extraction* the kernels that determine the neighborhood's extraction can also be tuned to better represent the aspect ratio of the original input, which, when preserved, could lead to higher performances. More metrics, which better cover, explainability and *human-interpretability*, according to our notion of *faithfulness* and *natural encoding* can also be constructed. In conclusion, these are only some of the directions that future work can be built upon and are truly only a glimpse of the potential path available for exploration as the novelty of our contribution offers ample space for further investigation.

BIBLIOGRAPHY

[1] Khouloud Abdelli, Helmut Grießer, and Stephan Pachnicke. A Hybrid CNN-LSTM Approach for Laser Remaining Useful Life Prediction. *26th Optoelectronics and Communications Conference (2021), paper S3D.3*, page S3D.3, 7 2021. doi: 10.1364/OECC.2021.S3D.3. URL https://opg.optica.org/abstract.cfm?uri=OECC-2021-S3D.3.

[2] Abhinav Saxena, Kai Goebel, Don Simon, and Neil Eklund. Turbofan Engine Degradation Simulation Data Set, 2023. URL https://data.nasa.gov/Aerospace/CMAPSS-Jet-Engine-Simulated-Data/ff5v-kuh6.

[3] Mounia Achouch, Mariya Dimitrova, Khaled Ziane, Sasan Sattarpanah Karganroudi, Rizck Dhouib, Hussein Ibrahim, and Mehdi Adda. On Predictive Maintenance in Industry 4.0: Overview, Models, and Challenges. *Applied Sciences 2022, Vol. 12, Page 8081*, 12(16):8081, 8 2022. ISSN 2076-3417. doi: 10.3390/APP12168081. URL https://www.mdpi.com/2076-3417/12/16/8081/htmhttps://www.mdpi.com/2076-3417/12/16/8081.

[4] Amina Adadi and Mohammed Berrada. Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI). *IEEE Access*, 6:52138–52160, 9 2018. ISSN 21693536. doi: 10.1109/ACCESS.2018.2870052.

[5] Giduthuri Sateesh Babu, Peilin Zhao, and Xiao Li Li. Deep convolutional neural network based regression approach for estimation of remaining useful life. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9642:214–228, 2016. ISSN 16113349. doi: 10.1007/978-3-319-32025-0{_}14.

[6] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador Garcia, Sergio Gil-Lopez, Daniel Molina, Richard Benjamins, Raja Chatila, and Francisco Herrera. Explainable Artificial Intelligence (XAI): Concepts, Taxonomies, Opportunities and Challenges toward Responsible AI. *Information Fusion*, 58:82–115, 10 2019. ISSN 15662535. doi: 10.48550/arxiv.1910.10045. URL https://arxiv.org/abs/1910.10045v2.

[7] Alexandros Bousdekis, Katerina Lepenioti, Dimitris Apostolou, and Gregoris Mentzas. Decision Making in Predictive Maintenance: Literature Review and Research Agenda for Industry 4.0.

IFAC-PapersOnLine, 52(13):607–612, 1 2019. ISSN 2405-8963. doi: 10.1016/J.IFACOL.2019.11.226.

[8] Sravan Kumar Challa, Akhilesh Kumar, and Vijay Bhaskar Semwal. A multibranch CNN-BiLSTM model for human activity recognition using wearable sensor data. *Visual Computer*, 38(12):4095–4109, 12 2022. ISSN 01782789. doi: 10.1007/S00371-021-02283-3/METRICS. URL https://link.springer.com/article/10.1007/s00371-021-02283-3.

[9] Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N. Balasubramanian. Grad-CAM++: Improved Visual Explanations for Deep Convolutional Networks. *Proceedings - 2018 IEEE Winter Conference on Applications of Computer Vision, WACV 2018*, 2018-January:839–847, 10 2017. doi: 10.1109/wacv.2018.00097. URL https://arxiv.org/abs/1710.11063v3.

[10] Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N. Balasubramanian. Grad-CAM++: Generalized gradient-based visual explanations for deep convolutional networks. *Proceedings - 2018 IEEE Winter Conference on Applications of Computer Vision, WACV 2018*, 2018-January:839–847, 5 2018. doi: 10.1109/WACV.2018.00097.

[11] Arun Das and Paul Rad. Opportunities and Challenges in Explainable Artificial Intelligence (XAI): A Survey. *CoRR*, abs/2006.11371, 6 2020. doi: 10.48550/arxiv.2006.11371. URL https://arxiv.org/abs/2006.11371v2.

[12] Saurabh Desai and Harish G. Ramaswamy. Ablation-CAM: Visual explanations for deep convolutional network via gradient-free localization. *Proceedings - 2020 IEEE Winter Conference on Applications of Computer Vision, WACV 2020*, pages 972–980, 3 2020. doi: 10.1109/WACV45572.2020.9093360.

[13] B.S. Dhillon. *Engineering Maintenance : A Modern Approach*. CRC Press, 2 2002. ISBN 9780429132209. doi: 10.1201/9781420031843. URL https://www.taylorfrancis.com/books/mono/10.1201/9781420031843/engineering-maintenance-dhillon.

[14] Finale Doshi-Velez and Been Kim. Towards A Rigorous Science of Interpretable Machine Learning. -, 2 2017. doi: 10.48550/arxiv.1702.08608. URL https://arxiv.org/abs/1702.08608v2.

[15] Rachel Lea Draelos and Lawrence Carin. Use HiResCAM instead of Grad-CAM for faithful explanations of convolutional neural networks. 11 2020. URL https://arxiv.org/abs/2011.08891v4.

[16] Mengnan Du, Ninghao Liu, and Xia Hu. Techniques for Interpretable Machine Learning. *Communications of the ACM*, 63(1):

68–77, 7 2018. ISSN 15577317. doi: 10.48550/arxiv.1808.00033. URL https://arxiv.org/abs/1808.00033v3.

[17] David J. Edwards, Gary D. Holt, and F. C. Harris. Predictive maintenance techniques and their relevance to construction plant. *Journal of Quality in Maintenance Engineering*, 4(1):25–37, 1998. ISSN 13552511. doi: 10.1108/13552519810369057/FULL/XML.

[18] Alexandre Englebert, Olivier Cornu, and Christophe De Vleeschouwer. Poly-CAM: High resolution class activation map for convolutional neural networks. 4 2022. URL https://arxiv.org/abs/2204.13359v2.

[19] Dumitru Erhan, Y Bengio, Aaron Courville, and Pascal Vincent. Visualizing Higher-Layer Features of a Deep Network. *Technical Report, Univeristé de Montréal*, 3 2009.

[20] Gabriel Erion, Joseph D. Janizek, Pascal Sturmfels, Scott M. Lundberg, and Su In Lee. Improving performance of deep learning models with axiomatic attribution priors and expected gradients. *Nature Machine Intelligence*, 3(7):620–631, 6 2019. ISSN 25225839. doi: 10.48550/arxiv.1906.10670. URL https://arxiv.org/abs/1906.10670v2.

[21] FranÃ§ois-Guillaume Fernandez. TorchCAM: class activation explorer. https://github.com/frgfm/torch-cam, 3 2020. URL https://github.com/frgfm/torch-cam.

[22] Ruigang Fu, Qingyong Hu, Xiaohu Dong, Yulan Guo, Yinghui Gao, and Biao Li. Axiom-based Grad-CAM: Towards Accurate Visualization and Explanation of CNNs. 8 2020. URL https://arxiv.org/abs/2008.02312v4.

[23] Jacob Gildenblat and contributors. PyTorch library for CAM methods. https://github.com/jacobgil/pytorch-grad-cam, 2021.

[24] Leilani H. Gilpin, David Bau, Ben Z. Yuan, Ayesha Bajwa, Michael Specter, and Lalana Kagal. Explaining Explanations: An Overview of Interpretability of Machine Learning. *2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 80–89, 1 2018. doi: 10.1109/DSAA.2018.00018.

[25] Daniel G Goldstein, Jake M Hofman, Microsoft Research JENNIFER WORTMAN VAUGHAN, Forough Poursabzi-Sangdeh, Jennifer Wortman Vaughan, and Hanna Wallach. Manipulating and Measuring Model Interpretability. *Conference on Human Factors in Computing Systems - Proceedings*, 67, 2 2018. doi: 10.1145/3411764.3445315. URL https://arxiv.org/abs/1802.07810v5.

[26] David Gunning. DARPA's explainable artificial intelligence (XAI) program. *AI Magazine*, 40:44–58, 3 2019. doi: 10.1145/3301275.3308446. URL https://dl.acm.org/doi/10.1145/3301275.3308446.

[27] 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, 6 2018. ISSN 10495258. doi: 10.48550/arxiv.1806.10758. URL https://arxiv.org/abs/1806.10758v3.

[28] Mohammad A.A.K. Jalwana, Naveed Akhtar, Mohammed Bennamoun, and Ajmal Mian. CAMERAS: Enhanced Resolution And Sanity preserving Class Activation Mapping for image saliency. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 16322–16331, 6 2021. ISSN 10636919. doi: 10.1109/CVPR46437.2021.01606. URL https://arxiv.org/abs/2106.10649v1.

[29] Mohd Javaid, Abid Haleem, Ravi Pratap Singh, Shanay Rab, and Rajiv Suman. Significance of sensors for industry 4.0: Roles, capabilities, and applications. *Sensors International*, 2:100110, 1 2021. ISSN 2666-3511. doi: 10.1016/J.SINTL.2021.100110.

[30] Peng Tao Jiang, Chang Bin Zhang, Qibin Hou, Ming Ming Cheng, and Yunchao Wei. LayerCAM: Exploring hierarchical class activation maps for localization. *IEEE Transactions on Image Processing*, 30:5875–5888, 2021. ISSN 19410042. doi: 10.1109/TIP.2021.3089943.

[31] Pallabi Kakati, Devendra Dandotiya, and Bhaskar Pal. Remaining Useful Life Predictions for Turbofan Engine Degradation Using Online Long Short-Term Memory Network. *ASME 2019 Gas Turbine India Conference, GTINDIA 2019*, 2, 1 2020. doi: 10.1115/GTINDIA2019-2368. URL /GTINDIA/proceedings-abstract/GTINDIA2019/83532/1073796.

[32] 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. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11700 LNCS:267–280, 11 2017. ISSN 16113349. doi: 10.48550/arxiv.1711.00867. URL https://arxiv.org/abs/1711.00867v1.

[33] Jay Lee, Yan Chen, Hassan Al-Atat, Mohamed Abuali, and Edzel Lapira. A systematic approach for predictive maintenance service design: Methodology and applications. *International Journal of Internet Manufacturing and Services*, 2(1):76–94, 2009. ISSN 17516056. doi: 10.1504/IJIMS.2009.031341.

[34] Miguel Lerma and Mirtha Lucas. Grad-CAM++ is Equivalent to Grad-CAM With Positive Gradients. -, pages 113–120, 5 2022. doi: 10.48550/arxiv.2205.10838. URL https://arxiv.org/abs/2205.10838v1.

[35] Xiang Li, Qian Ding, and Jian Qiao Sun. Remaining useful life estimation in prognostics using deep convolution neural networks. *Reliability Engineering & System Safety*, 172:1–11, 4 2018. ISSN 0951-8320. doi: 10.1016/J.RESS.2017.11.021.

[36] Q. Vera Liao, Daniel Gruen, and Sarah Miller. Questioning the AI: Informing Design Practices for Explainable AI User Experiences. *Conference on Human Factors in Computing Systems - Proceedings*, 1 2020. doi: 10.1145/3313831.3376590. URL http://arxiv.org/abs/2001.02478http://dx.doi.org/10.1145/3313831.3376590.

[37] Min Lin, Qiang Chen, and Shuicheng Yan. Network In Network. *2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings*, 12 2013. doi: 10.48550/arxiv.1312.4400. URL https://arxiv.org/abs/1312.4400v3.

[38] Zhong Qiu Lin, Mohammad Javad Shafiee, Stanislav Bochkarev, Michael St. Jules, Xiao Yu Wang, and Alexander Wong. Do Explanations Reflect Decisions? A Machine-centric Strategy to Quantify the Performance of Explainability Algorithms. *CoRR*, abs/1910.07387, 10 2019. doi: 10.48550/arxiv.1910.07387. URL https://arxiv.org/abs/1910.07387v2.

[39] Bin Lu, David B. Durocher, and Peter Stemper. Predictive maintenance techniques. *IEEE Industry Applications Magazine*, 15(6): 52–60, 2009. ISSN 10772618. doi: 10.1109/MIAS.2009.934444.

[40] Sina Mohseni, Niloofar Zarei, and Eric D. Ragan. A Multidisciplinary Survey and Framework for Design and Evaluation of Explainable AI Systems. *ACM Transactions on Interactive Intelligent Systems*, 11(3-4):46, 11 2018. ISSN 21606463. doi: 10.48550/arxiv.1811.11839. URL https://arxiv.org/abs/1811.11839v5.

[41] Menaka Narayanan, Emily Chen, Jeffrey He, Been Kim, Sam Gershman, and Finale Doshi-Velez. How do Humans Understand Explanations from Machine Learning Systems? An Evaluation of the Human-Interpretability of Explanation. *CoRR*, abs/1802.00682, 2 2018. doi: 10.48550/arxiv.1802.00682. URL https://arxiv.org/abs/1802.00682v1.

[42] Daniel Omeiza, Skyler Speakman, Celia Cintas, and Komminist Weldermariam. Smooth Grad-CAM++: An Enhanced Inference Level Visualization Technique for Deep Convolutional Neural

Network Models. *CoRR*, abs/1908.01224, 8 2019. doi: 10.48550/arxiv.1908.01224. URL https://arxiv.org/abs/1908.01224v1.

[43] Vitali Petsiuk, Abir Das, and Kate Saenko. RISE: Randomized Input Sampling for Explanation of Black-box Models. *British Machine Vision Conference 2018, BMVC 2018*, 6 2018. URL https://arxiv.org/abs/1806.07421v3.

[44] P. Poor, J. Basl, and D. Zenisek. Predictive Maintenance 4.0 as next evolution step in industrial maintenance development. *Proceedings - IEEE International Research Conference on Smart Computing and Systems Engineering, SCSE 2019*, pages 245–253, 3 2019. doi: 10.23919/SCSE.2019.8842659.

[45] Zhongang Qi, Saeed Khorram, and Li Fuxin. Visualizing Deep Networks by Optimizing with Integrated Gradients. *AAAI 2020 - 34th AAAI Conference on Artificial Intelligence*, pages 11890–11898, 5 2019. doi: 10.1609/aaai.v34i07.6863. URL http://arxiv.org/abs/1905.00954http://dx.doi.org/10.1609/aaai.v34i07.6863.

[46] Michael Ridley. Explainable Artificial Intelligence (XAI). *Information Technology and Libraries*, 41(2), 6 2022. ISSN 2163-5226. doi: 10.6017/ITAL.V41I2.14683. URL https://ejournals.bc.edu/index.php/ital/article/view/14683.

[47] Andreja Rojko. Industry 4.0 Concept: Background and Overview. *International Journal of Interactive Mobile Technologies (iJIM)*, 11 (5):77–90, 7 2017. ISSN 1865-7923. doi: 10.3991/IJIM.V11I5.7072. URL https://online-journals.org/index.php/i-jim/article/view/7072.

[48] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C Berg, Li Fei-Fei, O Russakovsky, J Deng, H Su, J Krause, S Satheesh, S Ma, Z Huang, A Karpathy, A Khosla, M Bernstein, A C Berg, and L Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3):211–252, 9 2014. ISSN 15731405. doi: 10.1007/s11263-015-0816-y. URL https://arxiv.org/abs/1409.0575v3.

[49] Wojciech Samek, Thomas Wiegand, and Klaus-Robert Müller. Explainable Artificial Intelligence: Understanding, Visualizing and Interpreting Deep Learning Models. *CoRR*, abs/1708.08296, 8 2017. doi: 10.48550/arxiv.1708.08296. URL https://arxiv.org/abs/1708.08296v1.

[50] Sam Sattarzadeh, Mahesh Sudhakar, Konstantinos N. Plataniotis, Jongseong Jang, Yeonjeong Jeong, and Hyunwoo Kim. In-

tegrated Grad-CAM: Sensitivity-Aware Visual Explanation of Deep Convolutional Networks via Integrated Gradient-Based Scoring. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2021-June:1775–1779, 2 2021. ISSN 15206149. doi: 10.48550/arxiv.2102.07805. URL https://arxiv.org/abs/2102.07805v1.

[51] Sule Selcuk. Predictive maintenance, its implementation and latest trends. *http://dx.doi.org/10.1177/0954405415601640*, 231(9):1670–1679, 1 2016. ISSN 20412975. doi: 10.1177/0954405415601640. URL https://journals.sagepub.com/doi/abs/10.1177/0954405415601640.

[52] 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, 10 2016. doi: 10.1007/s11263-019-01228-7. URL http://arxiv.org/abs/1610.02391http://dx.doi.org/10.1007/s11263-019-01228-7.

[53] Oscar Serradilla, Ekhi Zugasti, Jon Rodriguez, and Urko Zurutuza. Deep learning models for predictive maintenance: a survey, comparison, challenges and prospect. *Applied Intelligence*, 52(10):10934–10964, 10 2020. ISSN 15737497. doi: 10.48550/arxiv.2010.03207. URL https://arxiv.org/abs/2010.03207v1.

[54] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning Important Features Through Propagating Activation Differences. *34th International Conference on Machine Learning, ICML 2017*, 7:4844–4866, 4 2017. doi: 10.48550/arxiv.1704.02685. URL https://arxiv.org/abs/1704.02685v2.

[55] Luis P. Silvestrin, Mark Hoogendoorn, and Ger Koole. A Comparative Study of State-of-the-Art Machine Learning Algorithms for Predictive Maintenance. *2019 IEEE Symposium Series on Computational Intelligence, SSCI 2019*, pages 760–767, 12 2019. doi: 10.1109/SSCI44817.2019.9003044.

[56] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 9 2014. URL https://arxiv.org/abs/1409.1556v6.

[57] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. *2nd International Conference on Learning Representations, ICLR 2014 - Workshop Track Proceedings*, 12

2013. doi: 10.48550/arxiv.1312.6034. URL https://arxiv.org/abs/1312.6034v2.

[58] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. SmoothGrad: removing noise by adding noise. *CoRR*, abs/1706.03825, 6 2017. doi: 10.48550/arxiv.1706.03825. URL https://arxiv.org/abs/1706.03825v1.

[59] Leslie N. Smith and Nicholay Topin. Super-Convergence: Very Fast Training of Neural Networks Using Large Learning Rates. page 36, 8 2017. ISSN 1996756X. doi: 10.1117/12.2520589. URL https://arxiv.org/abs/1708.07120v3.

[60] David Solís-Martín, Juan Galán-Páez, and JoaquÃn Borrego-Díaz. On the Soundness of XAI in Prognostics and Health Management (PHM). *Information (Switzerland)*, 14(5), 3 2023. ISSN 20782489. doi: 10.3390/info14050256. URL https://arxiv.org/abs/2303.05517v1.

[61] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for Simplicity: The All Convolutional Net. *3rd International Conference on Learning Representations, ICLR 2015 - Workshop Track Proceedings*, 12 2014. doi: 10.48550/arxiv.1412.6806. URL https://arxiv.org/abs/1412.6806v3.

[62] Suraj Srinivas and FranÃ§ois Fleuret. Full-Gradient Representation for Neural Network Visualization. *Advances in Neural Information Processing Systems*, 32, 5 2019. ISSN 10495258. doi: 10.48550/arxiv.1905.00780. URL https://arxiv.org/abs/1905.00780v4.

[63] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic Attribution for Deep Networks. *34th International Conference on Machine Learning, ICML 2017*, 7:5109–5118, 3 2017. doi: 10.48550/arxiv.1703.01365. URL https://arxiv.org/abs/1703.01365v2.

[64] Michael van Lent, William Fisher, and Michael Mancuso. An Explainable Artificial Intelligence System for Small-unit Tactical Behavior. In *AAAI Conference on Artificial Intelligence*, 2004.

[65] Giulia Vilone and Luca Longo. Explainable Artificial Intelligence: a Systematic Review. *CoRR*, abs/2006.00093, 5 2020. doi: 10.48550/arxiv.2006.00093. URL https://arxiv.org/abs/2006.00093v4.

[66] 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. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2020-June:111–119, 10 2019.

ISSN 21607516. doi: 10.1109/CVPRW50498.2020.00020. URL https://arxiv.org/abs/1910.01279v2.

[67] Jihong Yan, Yue Meng, Lei Lu, and Lin Li. Industrial Big Data in an Industry 4.0 Environment: Challenges, Schemes, and Applications for Predictive Maintenance. *IEEE Access*, 5:23484–23491, 10 2017. ISSN 21693536. doi: 10.1109/ACCESS.2017.2765544.

[68] Mengjiao Yang Been Kim Google Brain and Google Brain. Benchmarking Attribution Methods with Relative Feature Importance. *CoRR*, abs/1907.09701, 7 2019. doi: 10.48550/arxiv.1907.09701. URL https://arxiv.org/abs/1907.09701v2.

[69] Chih Kuan Yeh, Cheng Yu Hsieh, Arun Sai Suggala, David I. Inouye, and Pradeep Ravikumar. On the (In)fidelity and Sensitivity for Explanations. *Advances in Neural Information Processing Systems*, 32, 1 2019. ISSN 10495258. doi: 10.48550/arxiv.1901.09392. URL https://arxiv.org/abs/1901.09392v4.

[70] Xiaochun Yin, Zengguang Liu, Deyong Liu, and Xiaojun Ren. A Novel CNN-based Bi-LSTM parallel model with attention mechanism for human activity recognition with noisy data. *Scientific Reports 2022 12:1*, 12(1):1–11, 5 2022. ISSN 2045-2322. doi: 10.1038/s41598-022-11880-8. URL https://www.nature.com/articles/s41598-022-11880-8.

[71] Matthew D. Zeiler and Rob Fergus. Visualizing and Understanding Convolutional Networks. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8689 LNCS(PART 1):818–833, 11 2013. ISSN 16113349. doi: 10.48550/arxiv.1311.2901. URL https://arxiv.org/abs/1311.2901v3.

[72] Matthew D. Zeiler, Graham W. Taylor, and Rob Fergus. Adaptive deconvolutional networks for mid and high level feature learning. *Proceedings of the IEEE International Conference on Computer Vision*, pages 2018–2025, 2011. doi: 10.1109/ICCV.2011.6126474.

[73] Shuai Zheng, Kosta Ristovski, Ahmed Farahat, and Chetan Gupta. Long Short-Term Memory Network for Remaining Useful Life estimation. *2017 IEEE International Conference on Prognostics and Health Management, ICPHM 2017*, pages 88–95, 7 2017. doi: 10.1109/ICPHM.2017.7998311.

[74] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning Deep Features for Discriminative Localization. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016-December:2921–2929, 12 2015. ISSN 10636919. doi: 10.48550/arxiv.1512.04150. URL https://arxiv.org/abs/1512.04150v1.

[75] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning Deep Features for Discriminative Localization. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016-December:2921–2929, 12 2015. ISSN 10636919. doi: 10.48550/arxiv.1512.04150. URL https://arxiv.org/abs/1512.04150v1.

[76] Tiago Zonta, Cristiano AndrÃ© da Costa, Rodrigo da Rosa Righi, Miromar JosÃ© de Lima, Eduardo Silveira da Trindade, and Guann Pyng Li. Predictive maintenance in the Industry 4.0: A systematic literature review. *Computers & Industrial Engineering*, 150:106889, 12 2020. ISSN 0360-8352. doi: 10.1016/J.CIE.2020.106889.