Wrocław University
of Science and Technology

DOCTORAL THESIS

Methods for Interpretation and Validation of Representations Generated by Deep Convolutional Neural Networks

mgr inż. Tomasz Michał
Szandała

Supervisor
dr hab. inż. Henryk Maciejewski

WROCŁAW 2022

Doktorat był niezwykłą przygodą w trakcie, której towarzyszyło mi liczne grono osób. Teraz chciałbym Im złożyć podziękowania za to, że byli ze mną przez ostatnie lata.

Za wsparcie, rady i cierpliwość chciałbym podziękować mojej najbliższej Rodzinie: Mamie, Siostrze Violi z Kamilem i Bratu Irkowi z Anią, oraz wszystkim moim Siostrzeńcom, Bratankom i Siostrzenicy.

Mojej Cioci Zosi i Siostrom Różowym z Nysy dziękuję za wsparcie duchowe. To też wiele daje.

Moi Przyjaciele z Nokii, z Liceum i spośród Ministrantów też wielokrotnie mnie motywowali i inspirowali, za to zawsze będę Im wdzięczny. Bez Was ta przygoda byłaby trudniejsza.

Jednocześnie szczególnie podziękowania kieruję w stronę mojego Promotora, Profesora Henryka Maciejewskiego, który był wspaniałym Mentorem i, za Swoją cierpliwość i spokój, zasługuje na miano Złotego Człowieka!

Dziękuję Wam wszystkim!

Streszczenie

Uczenie maszynowe, a szczególnie sztuczne sieci neuronowe pozwoliły dokonać przełomu w analizie informacji. Obecnie są wykorzystywane we wspomaganiu decyzji nt. udzielania kredytu, jazdy autonomicznym samochodem, diagnostyce medycznej i wielu innych. Niestety, mimo ich zachwycającej skuteczności stanowią dla użytkowników czarną skrzynkę. Często jedynym wyznacznikiem jakość stworzonego modelu jest ocena poprawności udzielania odpowiedzi. W celu lepszego zrozumienia decyzji sieci neuronowych, a także ich ulepszenia, trzeba lepiej analizować generowane przez nie reprezentacje.

W poniższej pracy proponuję serię metod, które są odpowiedzią na zidentyfikowane braki w procedurach analizy i oceny reprezentacji generowanych przez Głębokie Konwolucyjne Sztuczne Sieci Neuronowe.

Pierwsza grupa metod, to te wspomagające ocenę ocenianego modelu. Oprócz zwykłej skuteczności proponuję ocenę skupienia uwagi sieci, oraz wykrywaniu potencjalnych, ukrytych cech koniecznych, ale nie wystarczających do danej klasyfikacji.

W pierwszej metodzie wprowadzam liczbę ocenę skupienia sieci na obiekcie. Pozwala ona wykryć sytuację, gdy model bierze pod uwagę tło, czyli kontekst, w którym obiekt się znajduje, a ignoruje sam podmiot klasyfikacji. W tym celu wyznaczamy obszar obiektu (tzw. ROI - Region of Interest), a następnie oceniamy stosunek ważności obszaru wewnątrz ROI do całej ilustracji. Jako ważność przyjmuje wartości z mapy ciepła uzyskanej za pomocą wybranej metody wizualizacji - także licznie opisane w tej pracy. Jeżeli uzyskana wartość średnia dla danej klasy jest zauważalnie niższa oznacza to, że klasyfikator w tym przypadku może się skupiać na kontekście, a nie na samym obiekcie.

Druga metoda do automatycznej oceny modelu służy do wykrywania sytuacji, gdy tylko część obiektu jest czynnikiem decydującym o klasyfikacji. Podobnie jak pierwsza polega na wyznaczeniu ROI, a następnie obliczeniu stosunku pola o dużej ważności do pola całego ROI. Uzyskana wartość jest kolejną liczbową oceną poprawności działania utworzonego modelu i jej niska wartość dla poszczególnych klas oznacza klasy, które należy dogłębniej przeanalizować w celu wykrycia niepoprawnej generalizacji.

W przypadku drugiej metody zwykłe techniki wizualizacji okazały się niewystarczające - były zbyt rozmyte by dostrzec skupienie sieci na cechach obiektu, zamiast całego obiektu. W tym celu opracowałem metodę Stopniowego Rozszerzania (Gradual Extrapolation - GE), która zamiast rozszerzać skokowo mapę ważności, uzyskaną w głębokiej warstwie sieci neuronowej, rozszerza się warstwa po warstwie. Dodatkowo po każdej ekstrapolacji uzyskana mapa jest mnożona przez średnie wartości aktywacji w danej warstwie. Ta procedura z pierwotnie rozmytych, niewyraźnych map, tworzy szczegółowe ilustracje, znacznie wyraźniej oddające kształty obiektu lub elementu, który zdominował klasyfikację. Ponadto metoda GE jest kompatybilna z niemal każdą techniką wizualizacji wykorzystywaną w literaturze.

W celu udowodnienia skuteczności metody GE opracowałem i zarekomendowałem w kolejnej publikacji procedure oceny metod wizualizacji. Pozwala ona liczbowo porównać obecne jak i przyszłe metody, co ma pozwolić na lepszą ich systematykę. Test opiera się na trzech kryterach: wiarygodności, interpretowalności i aplikowalności.

Wiarygodność jest obliczana jako współczynnik utraty poprawności klasyfikacji po usunięciu odpowiednio 1%, 2%, 5%, 10%, itd. najważniejszych pikseli wskazanych wg ocenianej metody. Im szybciej pewność klasyfikacji spada, tym bardziej precyzyjna jest wskazana metoda.

Interpretowalności jest definiowana jako odpowiedź na pytanie: która metoda najbardziej zawęża informacje. Po porównaniu wejściowego obrazka z uzyskaną mapą ważności wycinamy piksele, które uzyskały ważność bliską zero. Im mniej pikseli pozostanie, tym wyżej w rankingu plasuje się analizowana metoda.

Ostatnim czynnikiem jest aplikowalności, czyli sprawdzenia, czy daną metodę wizualizacji da się zastosować do wielu modeli w skończonym czasie, przy ograniczonych zasobach obliczeniowych. W celu analizy aplikuje się tę samą procedurę wielokrotnie dla różnych modeli, szczególnie tych dobrze opisanych w literaturze. Podczas przeprowadzania testu, oprócz stosowalności metody z danym modelem należy zwrócić uwagę na czas potrzebny do uzyskania wyniki oraz niezmienność wyniku między wykonaniami dla tego samego przypadku. Warto zwrócić uwagę, że ta część testu nie jest decydująca - istnieją przypadki, gdzie metoda sprawdzi się tylko dla konkretnego modelu, albo wynik metody może się różnić między wykonaniami (np. LIME). Nie dyskwalifikuje to metody, ale wymaga by adnotacja o tym znalazła się w opisie techniki.

FIA-test (Faithfulness, Interpretability and Applicability) jest próba usystematyzowania prac dotyczących wyjaśnialnej sztucznej inteligencji, a zwłaszcza technik wizualizacji - generatorów map ważności.

Ostatnią metodą zaproponowaną w dysertacji jest Mapowanie Głównych Składowych (Principal Image Sections Mapping - PRISM). Polega ona na zastosowania Analizy Głównych Składowych dla reprezentacji wygenerowanej przez dany model. Metodę należy wykonywać jednocześnie dla grup obrazów. Uzyskana macierz Głównych Komponentów może być wykorzystana na 2 sposoby. PIerwszy polega na przypisaniu trzem pierwszym komponentom kolorów czerwony, zielony i niebieski, co pozwala na wizualne porównanie najważniejszych cech występujących na danych obrazach. Dodatkowo, połączenie PRISMa z GE pozwala uzyskać ilustracje prezentujące rozmieszczenie cech wykrytych przez model na obrazach. Procedura ta jest doskonałym uzupełnieniem metody Explanation by Example, gdzie staramy się dobrać obrazy podobne do badanego i wyciągnięciu wniosków o przyczynie danej klasyfikacji. Z PRISMem wskazanie cech wspólnych jest znacznie łatwiejsze.

Oryginalny wynik PRISMa może też posłużyć do masowej analizy wielu klas. Uzyskane Główny Składowe mogą zostać użyte jako dane wejściowe do dowolnej metody klasteryzacji. Po rozmieszczeniu poszczególnych obrazów w zadanej przestrzeni i utworzeniu klastrów można dostrzec nachodzące na siebie klastry. Takie obszary sugerują, iż wskazywane klasy posiadają podzbiór cech podobnych i potencjalnie mogą być błędnie sklasyfikowane przez badany model.

Metody i procedury zaproponowane w poniższej pracy są uzupełnieniem obecnie wykorzystywanych technik. Dodaję nowe kryteria oceny jakości modelu, oraz pozwalają wykryć newralgiczne klasy, które wymagają szczególnej uwagi. Mam nadzieję, że choć część z nich trafi do standardowych narzędzi wykorzystywanych przez inżynierów sztucznych sieci neuronowych.

Abstract

Machine learning and artificial neural networks have created an unprecedented breakthrough in data analysis. Nowadays they are used as decision reinforcements in loan decisions, autonomous cars driving, medicine and many more. Unfortunately they often are just a blackbox to their users. The main criteria for a model's acceptance is its accuracy in solving a given problem. In order to better understand their reasoning and thus improve performance a new trend arose: Explainable Artificial Intelligence. For Deep Convolutional Neural Network (DCNN) it focuses on studying and analyzing the representation generated by the model.

In this dissertation I am proposing several methods that are an answer to the gaps identified in interpretation and validation of representation generated by DCNNs.

The first group of methods aim to reinforce the model's evaluation. Apart from general accuracy of the classifying model I propose to evaluate its attention focus, whether it is in spurious correlation with context or focuses on the latent feature of the actual object.

The first approach I am introducing is a new quantitative metric of how much the model focuses on the object. It allows to identify circumstances where DCNN concentrates on the context of the object, instead of the classified item. We start by locating the Region of Interest(ROI) – a rectangular area around an interesting object. Next we generate a saliency map - using one of the techniques like GradCAM, also described in this paper. Finally we calculate the ratio of saliency inside ROI, divided by the sum of saliency in the entire image. If the obtained value, average for class, is significantly lower than other it may indicate that model is focusing on the context not at the object itself.

Second method to an automatic model's evaluation is the case when the model takes into consideration only a small part of the entire object - like wheels for a car. Similar to the previous one we draw the ROI around interesting objects and obtain saliency values for the image. Finally we divide saliency inside the ROI by the area of the ROI. If the value is noticeably lower for a certain class it indicates a potential latent discriminative feature is present and may incur faulty classification in future by breaking the model's generalization.

For the second method there was no visualization technique sharp enough to see the shape of the network's focus. For the case I have come up with a technique called Gradual Extrapolation (GE), which instead of directly resizing saliency from generated representation to the input image size, extrapolates layer by layer. Moreover after each extrapolation to the preceding layer we are multiplying the obtained matrix with average activation in the given layer. This procedure results in significantly sharper saliency maps from which we are usually able to recognize shapes of salient areas. Furthermore, the devised method is compatible with most of the other state-of-the-art visualization techniques.

In order to prove the value of the proposed GE method I have defined and recommended in another publication a set of tests for visualization techniques. It allows us to quantitatively compare any method that results in a saliency map image. It is based on three criteria: faithfulness, interpretability and applicability.

Faithfulness is calculated as a factor of accuracy drop after removal of 1%, 2%, 5%, 10% etc. of the most important pixels highlighted by the analyzed method. The faster the accuracy/confidence of classification drops, the more precise the method is.

Interpretability is defined as the answer to a question: which method reduces the amount of information the most. After comparison of the initial image with the saliency map we are removing

pixels which had saliency below the chosen threshold. The less pixels are left, the more effective the analyzed method is.

Final factor is the applicability, which means evaluation of whether a given method is applicable to the most of the state-of-the-art models in a finite time, with finite computing resources. To perform this test one has to perform several analyzed method on a set of different models. After that a time used for computation should be identified as well as the invariance of the result between tests for the same instance. Note that this test is not deterministic - there are methods that are applicable only to specific cases as well there are valuable methods that are slightly volatile, like LIME. Failure in the test does not disqualify the method but requires a clear statement of the method’s limitations.

FIA-test (Faithfulness, Interpretability and Applicability) is an attempt to order works about visualization, mainly the techniques that result in a saliency map.

The final method from this dissertation is a Principal Image Sections Mapping (PRISM). This technique relies on performing Principal Component Analysis on a layer that generates the final representation. This method should be executed for a batch of images. Obtained matrix of Principal Components (PC) can be utilized in two ways. First is to focus only on 3 (usually) first PCs and assign them colors, respectively: red, green and blue. This allows users to visually compare features present in an image identified by the model. Moreover, the combination of PRISM and GE results in illustrations that can be used even by a non-technical user to identify exclusive and inclusive features. This method is an excellent addendum to the Explanation by Example, where we try to find pictures similar to the analyzed ones and then make conclusions on the model's reasoning. With PRISM it is much simpler.

Additionally the PRISM’s original output may serve as an input for multiple class analysis for chosen clustering methods like Self Organizing Maps. After depicting each instance in a finite space and assigning them to a cluster we can identify overlapping clusters. These clusters indicate that respective classes have subset of similar features and thus can be misclassified. Similar to the two first methods: they require closer insight.

Methods and procedures proposed in the paper are supplementary to the state-of-the-art techniques. I am adding new criteria to DCNNs evaluation as well as tools to identify classes that require deeper analysis in order to create more robust models. I hope that some of my findings will find a place in a regular DCNN practitioner's toolbox.

1 Introduction

Convolutional Neural Networks (CNNs) have enabled unprecedented breakthroughs in a variety of computer vision tasks including image classification(Jeyakumar *et al.*, 2020), object detection(Wu *et al.*, 2019; Pham, Pham and Dang, 2020), semantic segmentation(Girshick *et al.*, 2014), image captioning(Johnson and Karpathy, 2016) and visual question answering(Gao *et al.*, 2015). However, even with such unprecedented advancements, the lack of full understanding regarding the decisions made by deep learning models and absence of control over their internal processes act as major setbacks in safety-critical decision-making processes, such as precision medicine or autonomous cars. In response, efforts are being made to make deep learning interpretable and controllable by humans.

Although the term Explainable artificial intelligence (XAI) emerged only recently, it is quite an old topic. The earliest work on XAI can be found several decades ago, back to the 1970s(Scott *et al.*, 1977; Swartout, 1985), but at that time, Artificial Intelligence (AI) and Machine learning (ML) were still in early development and the resources were very limited(Campbell, Hoane and Hsu, 2002) therefore the answer: why model generated a given representation was more a curiosity question.
In the last decades, AI and ML started to be used almost everywhere(Ferrucci *et al.*, 2013; Barredo Arrieta *et al.*, 2020), due to drop in resource prices like computational power and storage(Granter, Beck and Papke, 2017)). Due to the common presence of AI a need for explanation of its reasoning has increased. Questions like: why is this classified as a wolf? Why did this person not receive credit? Why is this person labeled as a potential disease carrier?(Sparkes, 2015; Gunning *et al.*, 2019)
Since the problem is vast and involves numerous fields of knowledge only a subset of the problem has been researched in my thesis. I am focusing on the representations generated by Deep Convolutional Neural Network(DCNN).

1.1 The Structure of this Work

This work has been divided into several chapters. Firstly I am introducing the motivations, so the gaps I have identified as a DCNN practitioner which I believe should be answered immediately (chapter *1.2 Motivation*). Secondly (chapter *1.3 Contribution*) I am briefly summarizing the contributions I have made towards solving identified problems. Description of each contributions consists of:

- explaining which known problem it tries to address
- summary of experiments performed with method, specifying main pros and cons
- recommendation where it should be applied
- reference to the paper it has been published in.

In the next chapter - *2 State-of-the-art*, I am familiarizing the reader with state-of-the-art mandatory to better understand the whole paper. Since I am focusing on analysis of representations generated by DCNNs I am describing other methods and procedures that provide insight into model visualization. Beginning from the commonly known ones like occlusion GradCAM or LIME, to less popular jak Excitation Backpropagation, SHAP or Explanation by Example.

Subsequent chapter, *3 Proposed Methods and Procedures,* is a theoretical description of proposed methods and formulas. This chapter allows the reader to reproduce procedures that are vital parts of my contributions.

After that comes the chapter with results from the research on each method - *4 Practical Application of Proposed Methods*. Each experiment describes models (often state-of-the-art) used in the process and as results and discussion about the outcome. With each research we can also find the setbacks of given methods or potentially open problems to be solved in future.

Final chapter *5 Conclusions* briefly summarizes all main contributions from the work and contains conclusions for each procedure proposed during my research.

1.2 Motivation

In this chapter I would like to ground why research done under this dissertation is up to date and important to DCNN applications.

1.2.1 Progress of DCNN

Image classification is a fundamental issue in computer vision which is a crucial part of AI, that focuses on grouping images into predefined classes. The concept constitutes the foundation for diverse tasks, including detection, localization, and segmentation(Rawat and Wang, 2017). Meanwhile, image classification is relatively complex for automated systems due to high in-class and viewpoint-dependent object variabilities. In this context, autonomous people have to effectively classify objects from diverse viewpoints besides differentiating them from similar objects or identifying them despite numerous obstructions in a given frame. The traditional dual-stage approach that relied on feature descriptors to solve the classification problem has become unreliable due to limited accuracy (Rawat and Wang, 2017). Effective classification depended on designing a reliable feature extraction stage.

In the recent past, machine learning (ML) algorithms have become increasingly popular for evaluating relationships in data making decisions without explicit instructions. The technological concept has enabled systems that emulate the human sensory responses, such as speech and vision (Khan *et al.*, 2020). The Deep Convolutional Neural Network (CNN) is a promising Neural Network (NN) with exemplary performance in Computer Vision and Image Processing. This Artificial NN has been highly popular in computer vision tasks since 2012 when they delivered astonishing results during the ImageNet Large Scale Visual Recognition Competition (ILSVRC) (Yamashita *et al.*, 2018). The deep learning architecture enthused by the natural visual perception mechanism of the living creatures has exemplary performance in visual recognition, speech recognition, and natural language processing . Thus, CNN is becoming increasingly popular in ML and automation of receptive fields.

CNN presents an enhanced opportunity in building AI systems that require extensive computer calculation power. However, computers are ineffective in solving problems that require intuition and experience, such as considering traffic direction and obstacles, understanding spoken words, and recognizing drawing in an image (Namatēvs, 2017). In this regard, the problem requires massively parallel processing systems with complex algorithms to attain considerable reliability, accuracy and efficiency. The NN learns spatial hierarchies of features through the using underlying multiple building blocks in different transactional layers (Yamashita *et al.*, 2018).

CNN's are a subset of ANNs, which employ general matrix multiplication in one of their layers rather than convolution. The CNN architecture exhibits significant improvements compared to previous methods of image classification. The distinctive trend in advancement of CNNs entails deepening networks (Gu *et al.*, 2018). For instance, ILSVRC

developed in 2015 is 20 times deeper compared to AlexNet designed in 2012. Thus, CNN entails a sophisticated interconnection grid of simple processing units and adjusting the weights or grid parameters through the learning process to complete tasks or adapt to the operational environment (Namatēvs, 2017). Unlike simple neural networks with one or more hidden layers, CNNs have numerous layers, including realizing forms of regularization, fully-connected layers, pooling, and convolution, which compactly represent highly nonlinear and varying functions (Namatēvs, 2017).
CNN's and deep architectures feature numerous neurons and several levels of latent nonlinearity calculations. Thus, the enhancement of depth increases the complexity of CNN, challenging optimization but easing overfitting, enhancing feature representations, and improving approximation of the target function with increased nonlinearity (Gu *et al.*, 2018). The exploitation of multiple layers of nonlinear information processing greatly enhances the performance of CNN in pattern analysis, classification, but also feature extraction and transformation. Thus making it the preferred architecture in image recognition, classification, and detection tasks (Rawat and Wang, 2017; Rosato *et al.*, 2021). Effective handling of small datasets and overfitting allows CNN use in diagnostic radiology for augmenting performance and improving patient care (Yamashita *et al.*, 2018). In this regard, the growing prominence of CNN architecture in the deep learning renaissance is powered by new graphical processing units (GPUs), larger data sets, and better algorithms.
CNN's have enabled unprecedented breakthroughs in a variety of computer vision tasks, including image classification (He *et al.*, 2016), object detection (Wu *et al.*, 2019; Pham, Pham and Dang, 2020), semantic segmentation (Girshick *et al.*, 2014), image captioning (Johnson and Karpathy, 2016), visual question answering (Gao *et al.*, 2015), and natural language processing (Hou *et al.*, 2021).

1.2.2 Need for Explainability

However, even with such unprecedented advancements, the lack of complete understanding regarding the decisions made by deep learning models and the absence of control over their internal processes act as significant setbacks. These setbacks impact mainly the safety-critical decision-making processes such as precision medicine, autonomous cars, unsafe workforce behaviors, deterioration patterns, structural defects, and latent risk factors (Hou *et al.*, 2021). CNN's involve millions of learnable parameters to estimate, necessitating GPUs for model training (Yamashita *et al.*, 2018). Nonetheless, alternatives to CNNs might be unreliable, labor-intensive, time-consuming, or sometimes risky to operate (Hou *et al.*, 2021). In response, efforts are being made to make deep learning interpretable and controllable by humans.
A range of explanation methods have been proposed in computer vision. Class Activation Maps, Gradient based techniques, LIME, Scale-Invariant Feature Transform (SIFT) and many more (in depth described in chapter 2.4 *CNN Visualization Methods*). The valuation

of the techniques depends on multiple measurable factors, such as speed, complexity, fidelity, contrastivity, usability, sparsity, sensitivity, size, multi-dimensionality, and readability(Fong and Vedaldi, 2017; Pope *et al.*, 2019; Jeyakumar *et al.*, 2020). However, many visualization methods suffer from several issues.
First concern regards to the size of the datasets used for training and validation. Model ZOO models for Pytorch are being trained with ImageNet dataset and take into consideration 1000 classes(Deng *et al.*, 2009; Szandala, 2021b). Reliability investigation of classification for all classes can be tedious or even impossible work to accomplish. Therefore it is reasonable to search for a solution that can automatically identify the ambiguous classes or at least significantly narrow down the list of potentially troublesome classes.

Taking into consideration only the high accuracy of the classification model is an insufficient indicator for network acceptance tests. There can be found in literature examples where the network has learnt spurious correlations instead of the expected object itself(Calude and Longo, 2017; Szyc, Walkowiak and Maciejewski, 2021). The most famous one is the training of bi-classificator for husky dogs and wolves(Ribeiro, Singh and Guestrin, 2016).

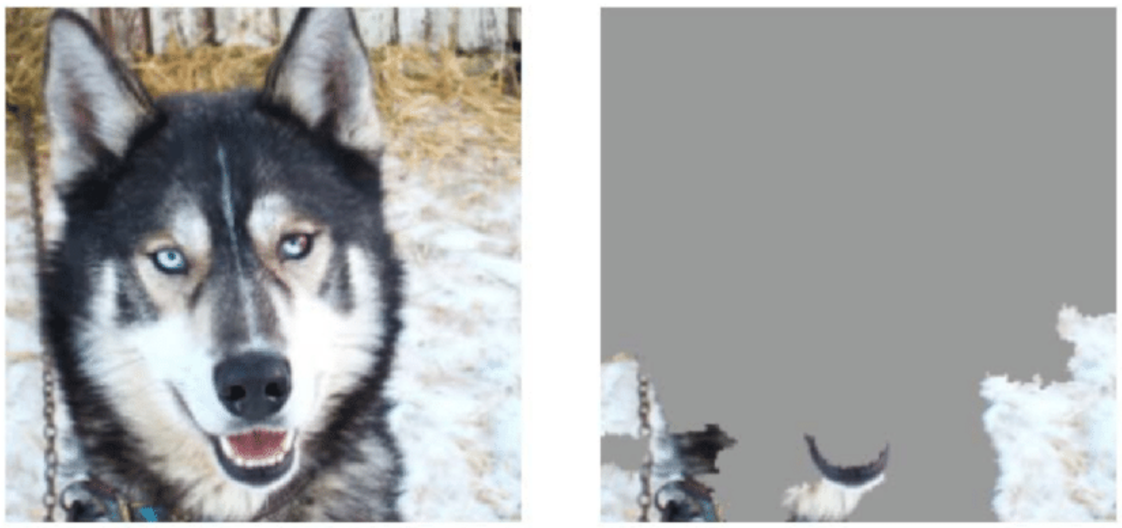


Fig. 1. Occluded image of a husky's face with only snow left is still classified as a wolf. Source: (Ribeiro, Singh and Guestrin, 2016)

Despite the network achieving remarkable accuracy after closer insight (see fig. 1.) it has been revealed that the main indicator for wolves was the snow as all wolves from the training set were depicted in the snowy environment. Moreover, even being aware of this setback, after retraining a model, this time, with wolves not only in the snowy background we have hit a problem. This time network has learnt branches and bushes as all our wolves were in the natural environment, unlike huskies, where the most were in the human-originated background, like houses (better described in chapter *4.1.1 Attention Focus Evaluation*).

Each visualization comes with a computational burden due to the use of large datasets for compelling real-time analysis. It is mandatory to evaluate whether given analysis can be performed in a finite and rather short time period but also provide usable information on a network's performance. In order to countermeasure these concerns a quantitative comparison method of explainability has to be devised.

Many of the common visualization techniques lack feature targeted precision(fig. 2.), thus might be unable to identify latent features in truth indicated given classification. One such example can be classification of Samoyed dogs (described in chapter *4.1.2 Detecting Latent Features*) where we noticed that this canine specimen, for the NN, is virtually only a 3 properly placed dots on a white background, therefore transfer of such dots to another object can potentially lead to misclassification of the example. Since most of the currently known saliency maps methods generate blurred and indistinct output there is a rising need for a solution that sharpens the outcome hence provides more detailed insight into classified objects.

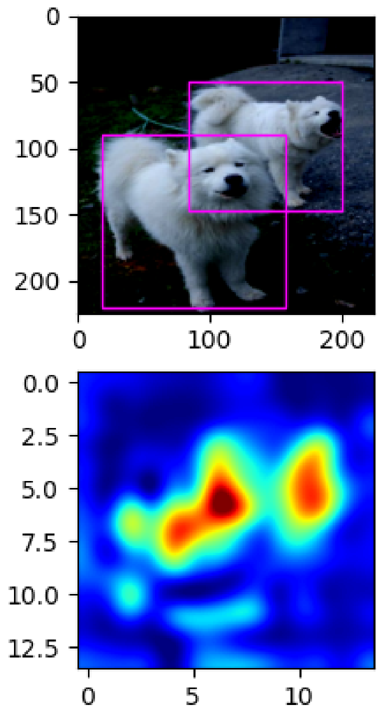


Fig. 2. Sample output of CNN representation visualization technique (GradCAM). We can see blurred heatmap that translates to the importance of given image area for classification

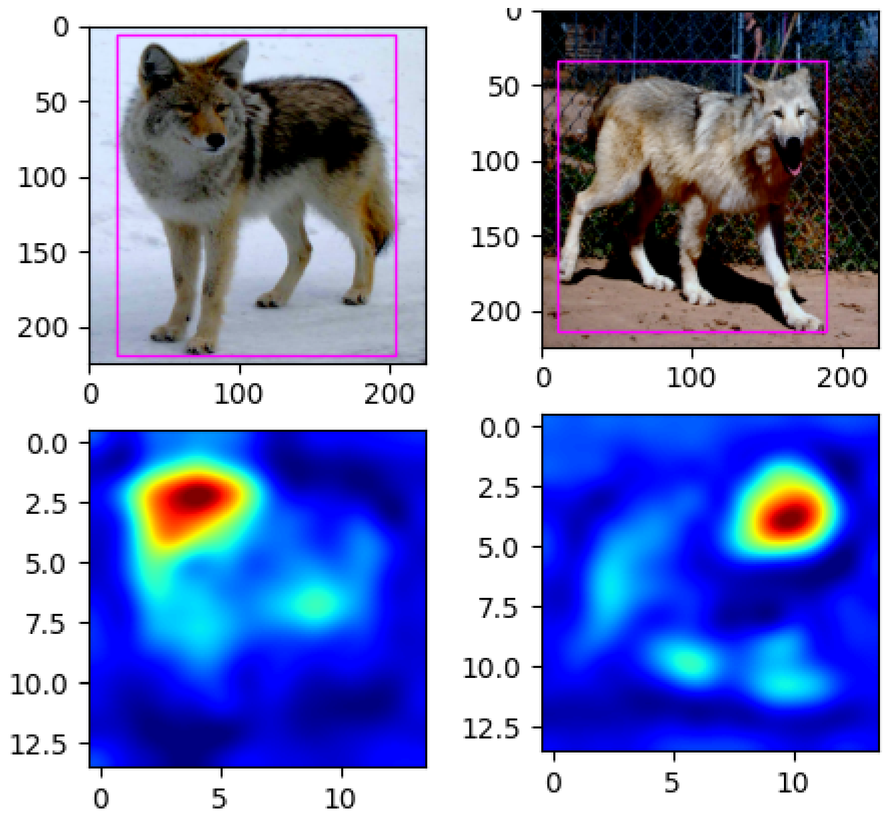


Fig. 3. Vague output from GradCAM for 2 specimens: coyote on the left and wolf on the right. In both cases a face is highlighted yet images are correctly classified. We do not see the actual difference between pictures

Another setback of commonly used methods is the lack of awareness whether highlighted features are considered to be equal by the NN model. As seen in the picture above both animals - coyote and timber wolf, are differentiated by their heads. But the questions rise: are both heads the same to the network? If so, why are they correctly classified as respective coyote and wolf? This proves the necessity for a tool for feature-level characteristic highlighting tool.

To sum up, in this work I have tried to:

- identify classes that categorization is based on a spurious correlation in the given image,
- detect examples where model is accidently trained to identify a latent feature, that might not always be a sufficient indicator in real world application,
- automate or at least significantly reduce the amount of manual examination that has to be performed during network validation,
- considerably improve the readability of outcome generated by the state-of-the-art representations analysis methods,
- devise a method to quantitatively evaluate and compare representation-analysis techniques,
- provide a technique to gain more insight into distinguishable features generated by the representations.

1.3 Contribution

In order to successfully train a Deep CNN (DCNN or DNN) to classify objects we need to feed it with thousands of images depicting each object. For networks taking part in the ImageNet contest(Zeiler and Fergus, 2014), where the dataset consists of over 14 million pictures it is virtually impossible to analyze each class to detect potential issues. There is also a vast multitude of potential issues like learning background(Xiao *et al.*, 2020), a feature unrelated to the actual object(Ribeiro, Singh and Guestrin, 2016) or only a small element of the object that is mandatory, but not sufficient(Soleimani *et al.*, 2018). In order to answer these issues I have proposed several new techniques.

1.3.1 Automatic Spurious Correlations Detection

This first problem that a CNN practitioner may encounter might be the background influence, like a camel always in a desert area, while a wolf mainly in snowy forest. For a small scale dataset it is quite simple: we may apply saliency map generating technique and arbitrarily evaluate whether the network's focus is in the background or in the anticipated object area.

In order to respond to this issue an attention focus evaluating technique has been proposed. We have introduced a procedure to obtain a metric of saliency fit to an object. This gives us a percentage of how much the model focused on an object. When this method has been applied to the entire dataset it could reveal classes whose classification relies on details loosely coupled with the actual object. Noteworthy results were:

- classification of snowboards (with custom network), where the network has learn the sky over the mountains instead people
- characteristic web of black-gold garden spider instead of spider itself - for the state-of-the-art model.

More research on the procedure is described in chapter: *4.1.1 Attention Focus Evaluation*.

However the method has one significant setback: it relies on ROI generation. Problem appears when a certain class cannot be indicated by ROI, e.g. landscape pictures like mountains, valles, lakesides. Deeper analyzed in chapter *4.1.1.3 Method’s Limitations.*

1.3.2 Quantitative Comparison of CNN Visualization Techniques

In the literature we can find a multitude of CNN visualization methods (see chapter *CNN Visualization Methods*). Some of them have specific pros and cons but most are focusing on generating a saliency map to visualize which part of the picture contributed the most to the obtained representation. A problem appears when we would like to introduce a new method. If it is not a problem-specific technique but just another saliency map generator why do we need it? Why is it better? Or at least how does it compare to already stated techniques?

A testing procedure has been designed FIAt to validate virtually any explanation technique. It consists of evaluation of Faithfulness, Interpretability and Applicability testing. The first criterion of an explanation is to reliably and comprehensively represent the local decision structure of the analyzed model. To assess such a property of the model, a proposed technique is "pixel flipping" (Samek *et al.*, 2017). The pixel-flipping procedure assesses whether removing the features highlighted by the explanation, as the most relevant, decreases the network prediction abilities.

Interpretability is in our opinion the hardest to estimate, therefore we need to get back to the root of the explanation methods concept. The aim is to reduce the information from the original object and only retain the elements that play the highest role in classification(Chen *et al.*, 2018). Based on this consideration, a test is established that computes the number of pixels of the original image remaining after its truncation only in salient areas.

Faithfulness and interpretability do not completely determine the overall usefulness of the explanation method. To characterize the usefulness, it is also necessary to determine whether the explanation method can be applied in various models and situations, at least to most state-of-the-art models, and whether the explanation can be obtained quickly enough with finite computational resources. To measure applicability, the proposed method is applied to several different models and their computational time is measured.

I have used this method to compare Gradual Extrapolation (see chapter: 1.3.3 CNN Visualization Technique Output Sharpening) with few state-of-the-art techniques. A similar approach should be taken when introducing other, saliency based methods.

It is mandatory for an AI researcher to analyze his or her model, to provide explainability for its decision. Despite a wide range of methods and techniques that mainly generate a saliency map, new procedures are being introduced. FIA-test is a method to quantitatively compare them and choose the most appropriate.

1.3.3 CNN Visualization Technique Output Sharpening

Not only background can be a spurious correlation. Potentially a small element of the object could be treated as a conclusive feature that determines given class, e.g. black widow's red hourglass pictured on an elephant could misguide the model. In order to detect this fact on a dataset we needed a new technique that focuses on an object's features in more detail.

CAMs can efficiently demonstrate the discriminative features of the input image; however, the resulting saliency map is quite fuzzy thus potentially making them useless for this problem.

Therefore, a simple yet efficient method is proposed that reduces the coarseness of the obtained heatmaps(Szandala, 2021a). Using a weighted mask multiplication technique, the sharpness of the features shown in the heatmap is improved herein. The proposed method enhances virtually any CAM technique to generate highly narrowed saliency maps; this method has been called Gradual Extrapolation and the sample result can be seen in figure 4.

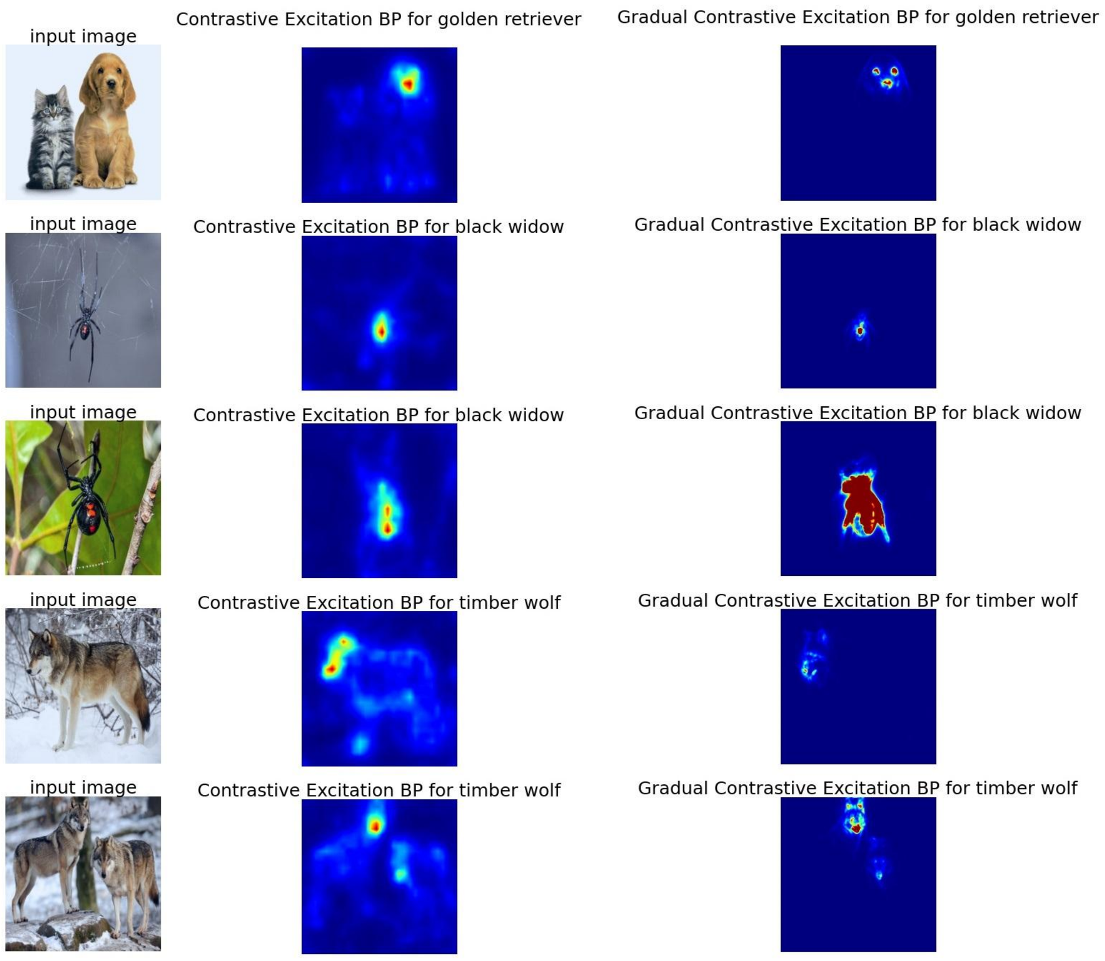

input image
Contrastive Excitation BP for golden retriever
Gradual Contrastive Excitation BP for golden retriever
input image
Contrastive Excitation BP for black widow
Gradual Contrastive Excitation BP for black widow
input image
Contrastive Excitation BP for black widow
Gradual Contrastive Excitation BP for black widow
input image
Contrastive Excitation BP for timber wolf
Gradual Contrastive Excitation BP for timber wolf
input image
Contrastive Excitation BP for timber wolf
Gradual Contrastive Excitation BP for timber wolf

Fig. 4. Contrastive Excitation Backpropagation (mid column) enhanced with Gradual Extrapolation provides visible improvement in sharpness (last column)

With the aforementioned evaluation procedure (FIAt) I have proven the usability of the technique. Experiment has been described in depth in chapter *4.2.1 CNN Visualization Performance Evaluation - FIAt*. Firstly, in the fidelity test, I was able to demonstrate that the method does not decrease original method localisation accuracy, but may even increase under certain circumstances.

Secondly, I have proven that GE enhancement is reducing the amount of information (in the form of highlighted pixels) displayed in the result of application. Also the reduction does not impact the interpretability of the outcome.

And finally I have applied the GE to a multitude of state-of-the-art networks in a finite time, therefore confirming that GE is easily applicable to most of the models and does not significantly impact the time of obtaining a saliency map.
This method is another tool in CNN practitioner's toolbox to analyze and improve his model. With GE we are able to have a closer look into the truly important features found by the network and thus adjust training procedures if the findings are not satisfying.

1.3.4 Automatic Latent Features Detection

The network may learn only a small trait if an object instead it as a whole. Unfortunately, sometimes the indicator might be an unrelated object(Lapuschkin *et al.*, 2019; Szandała and Maciejewski, 2021) or even a specific color(Szyc, 2020), but this paper aims to evaluate detect factors that are indeed important to the classification but should not be the sole clue for it. In example a factor determining black widow could be its characteristic red hourglass. Therefore any other object with it would be classified as a black widow spider. Yet these defects are recognizable merely by manual examination using Gradual Extrapolation (chapter: *1.3.3 CNN Visualization Technique Output Sharpening*). While for several classes it is achievable, for a wider range of types it can be challenging.
In this paper we aim to provide a solution to detect and thus avert learning partial correlations.
We have illustrated the usability of this method during several experiments(more details in chapter *4.1.2 Detecting Latent Features*). In the first part we have correctly indicated a class - wheeled vehicle, where the discriminative feature was a wheel. In other words: any object with a wheel was considered to be a vehicle. This hypothesis has been proven by performing an adversarial attack with a cassino roulette, which, as expected, has been classified as a vehicle.
In the next phase we used a state-of-the-art network from Pytorch’s Model ZOO and applied our method to evaluate results. It has identified a few suspicious classes, where the most noteworthy is the Samoyed class. According to our assumption: 3 black stains in a triangle manner on a white background is a Samoyed class. This hypothesis has been proven by another adversarial attack where we were thresholding a Samoyed picture until only eyes and tip of the nose left, which has been still labeled as Samoyed by the model.

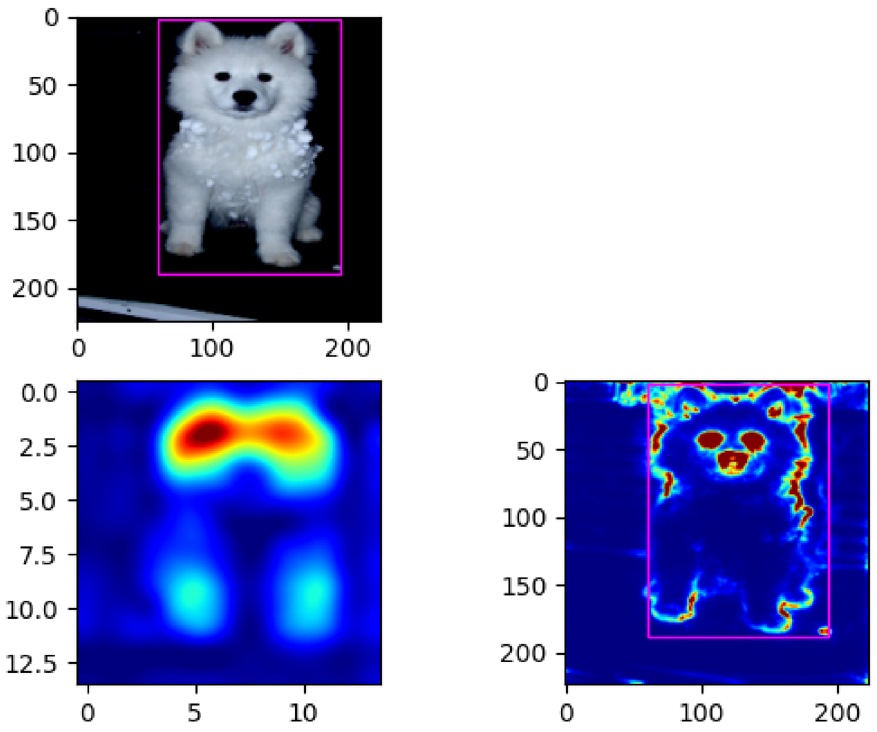


Fig. 5. Samoyed example where only a small part of ROI (violet rectangle) is an actual object. In other words we can assume only eyes and muzzle are significant to classification

Apart from this one we have also identified 3 false positives classes(e.g. can be seen in figure 6), which were highlighted by our methods but in the end were just corner cases (see chapters: *4.1.2.4 Method's Limitations: Landscape Areas* and *4.1.2.5 Method's Limitation: Hardly Descritable Objects*).

To sum up: previously mentioned contribution - the Spurious Correlation Detection (chapter: *1.3.1 Automatic Spurious Correlations Detection*) focuses on identifying classes where context plays a joyr role. Latent Features Detection is about investigation of the object itself - whether the classifier takes into consideration the object as a whole or only a singular element of this.

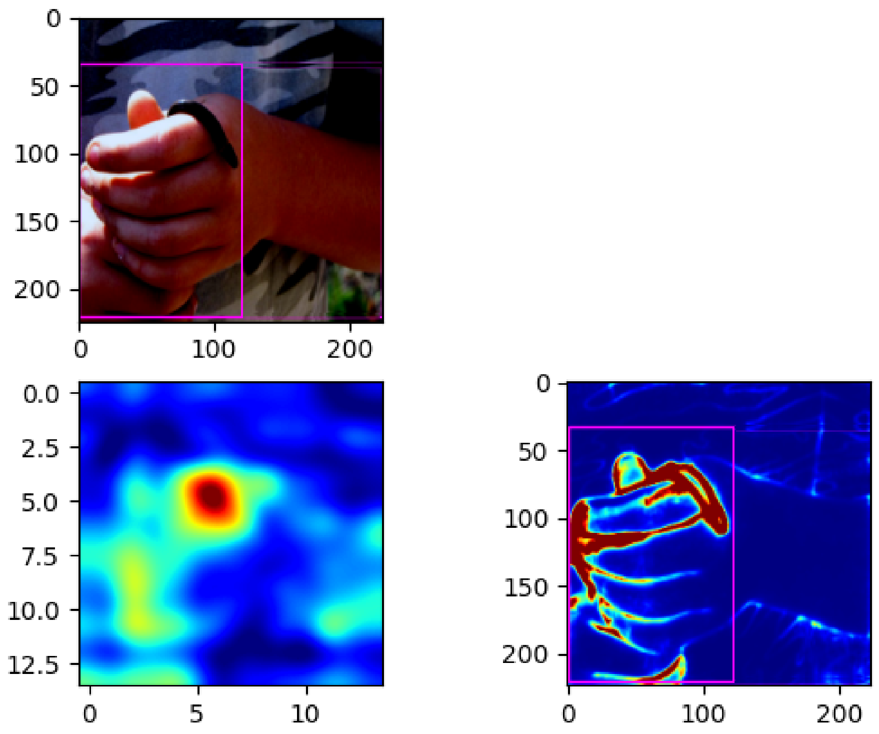


Fig. 6. Violet rectangle represents ROI for snake, but actual snake identified by Gradual Extrapolated GradCAM is mostly hidden inside hand therefore giving false positive of only a small part of the ROI being an actual object

1.3.5 Discriminative Features Identification Technique

Most representation attribution methods are based on backpropagation of the network's activations from the output back to the input. They are usually a modification of the backpropagation algorithm and, for computer vision models, take the form of a saliency map that highlights the decisive regions on the input image.

We have introduced a new method that relies on Principal Component Analysis of features detected by neural network models and interpolates them on the initial image. We call it the Principal Image Sections Mapping (PRISM). The result of the formula is an RGB colored image mask that assigns one color to each feature identified by the model. Moreover the same color will be used to highlight the same feature across all pictures processed in the same batch, of course only if it is present in other samples. This allows exposure of a comparative set of features between images processed in the same batch, and thus facilitates the Explanation by Example technique(Jeyakumar *et al.*, 2020).

During the research we are looking for a tool to examine whether two canine muzzles, e.g. coyote's and wolf's are considered to be the same by CNN's representation layer. However even the smallest networks have hundreds parameters in the representation layer, thus a procedure for reducing dimensionali was sought. The choice was Principal Component Analysis, which translates the network's final layer into a list of significance-ordered feature vectors. Moreover, processing several images at once gives output with the same features (paws, muzzles, keys, trees, etc.) being described by the same vectors.

While this output can be used for class clusterization as seen as an example in fig. 7. - thus providing insight into which classes are having resembling features, we can proceed to depicting the single image analysis in a human-comprehensive way, but truncating out the result to the most principal vectors.

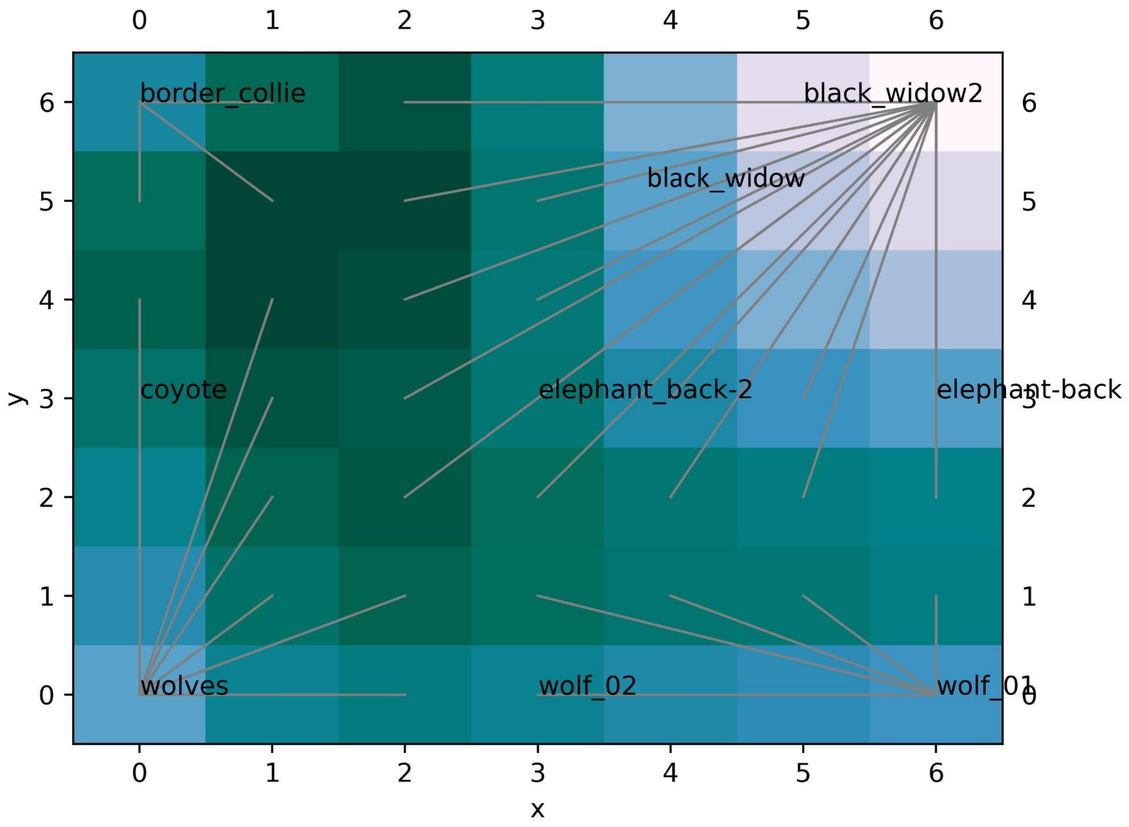


Fig. 7. Early example of clustering of 9 pictures using a Self-Organizing Map. Same classes are localized nearby. Also visible is that canines are closer to the left axis while other specimens like black widows or elephants are clearly separated from them.

The 3 first vectors can be assigned respectively red, green and blue thus providing a colored map of features identified by the model. These checkered output can be divided into inclusive and exclusive features as seen in figure 8. Furthermore, combination of this method with Gradual Extrapolation (chapter: 1.2.3 CNN Visualization Technique Output Sharpening) results in an image portraying each feature of a classified object in a different color. So processed results can be further used for identification of inclusive and exclusive features between images in a very high resolution, compared to other saliency maps like GradCAM or Excitation Backpropagation.

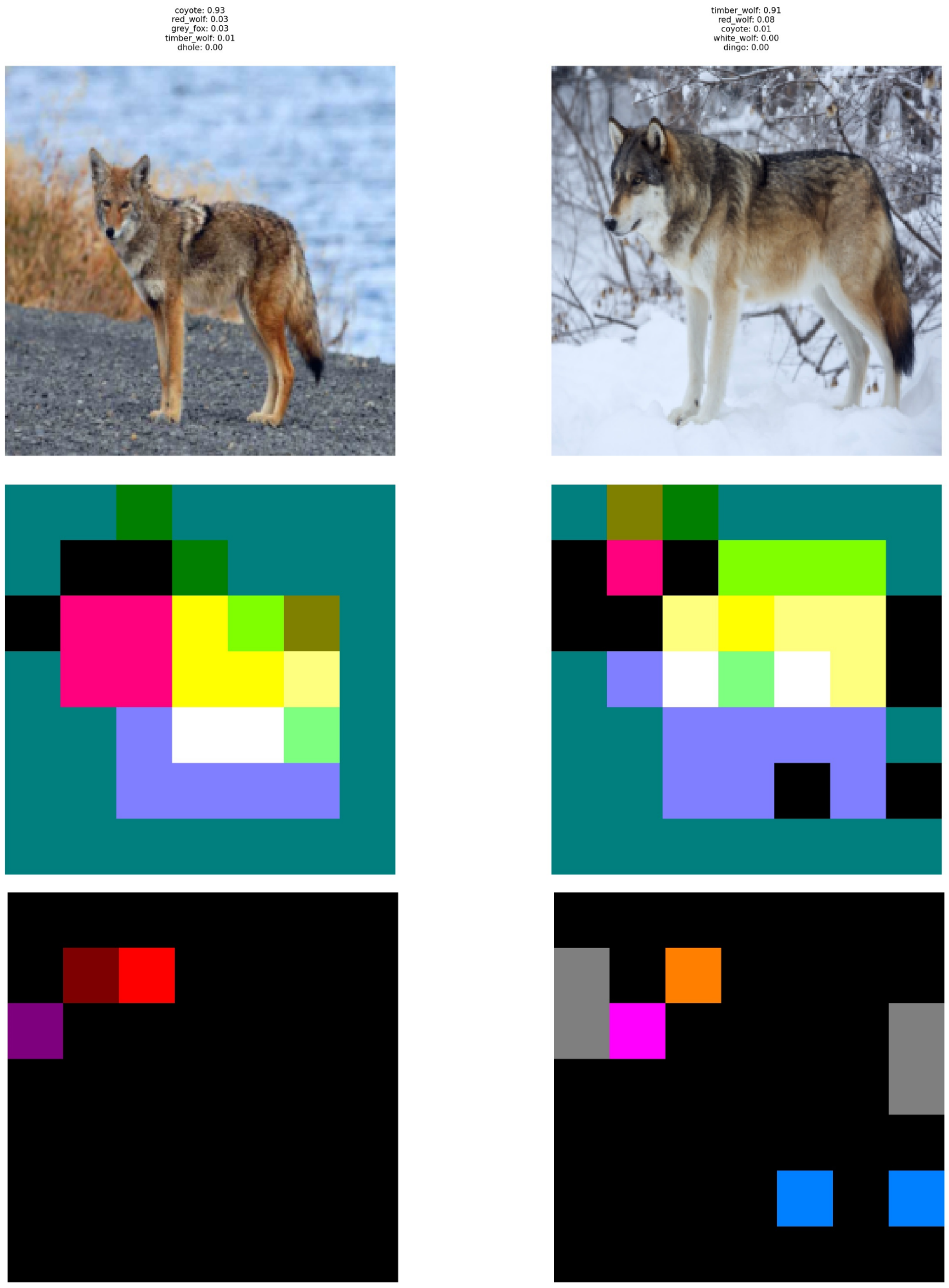


Fig. 8. Display of Principal Image Sections Mapping with split to inclusive (second row) and exclusive (third row) features for coyote (left) and timber wolf (right)

PRISM can be used both for better human-interpretation of representations generated by DCNNs and for automation of identification of ambiguous classes features. With human-interpretable usage of PRISM we can utilize results from technique Explanation by Example and depict inclusive and/or exclusive features identified by the model evn to a non-scientific viewer. On the other hand: taking PRISM output into clusterization we are able to identify potentially indistinct classes and thus adjust the training to avoid possible vulnerabilities in real-world application of the model.

1.3.6 Publicly Available Code

Some of the works have been coded in Python and made publicly available as GitHub repositories.

- TorchPRISM - module implementing PRISM method using Python and PyTorch framework:
  https://github.com/szandala/TorchPRISM
  State on 6th June 2022: 12 Stargazers

- Gradual Extrapolation - sample implementation of Gradual Extrapolation meth using Python and PyTorch framework:
  https://github.com/szandala/gradual-extrapolation
  State on 6th June 2022: 5 Stargazers

2 State-of-the-art

2.1 Convolutional Neural Networks

CNN is a neural network that exploits spatial or temporal data correlation. The model entails weight sharing, multi-tasking, automatic feature extraction, and hierarchical learning(Guo *et al.*, 2016; Abbas, Ibrahim and Jaffar, 2019; Khan *et al.*, 2020). The underlying topology has multiple learning stages comprising a combination of nonlinear processing units, convolutional layers, subsampling layers, and input scanners at the start, as illustrated in Figure 2. The model involves a feedforward multilayered hierarchical network with several layers depending on the bank of convolutional kernels to multiple transformations(LeCun, Kavukcuoglu and Farabet, 2010). The convolution process is crucial for extracting valuable features from correlated data points on an image(Rawat and Wang, 2017). Meanwhile, the activation function of the nonlinear processing unit assumes the convolutional kernels' output as input for embedding nonlinearity in the feature space and learning abstractions. The activation function produces diverse activation patterns used for different responses, including learning diverse semantics found on an image (Rawat and Wang, 2017; Szandała, 2020a). The output of the nonlinear processing unit is parsed to subsampling for result summarizing and developing input invariant to geometrical distortions(LeCun, Kavukcuoglu and Farabet, 2010). In the meantime, automatic feature extraction ability ingrained in CNN eliminates the need for a distinct feature extractor (Najafabadi *et al.*, 2015). Hence, the model can exhaustively learn representation from raw pixels.

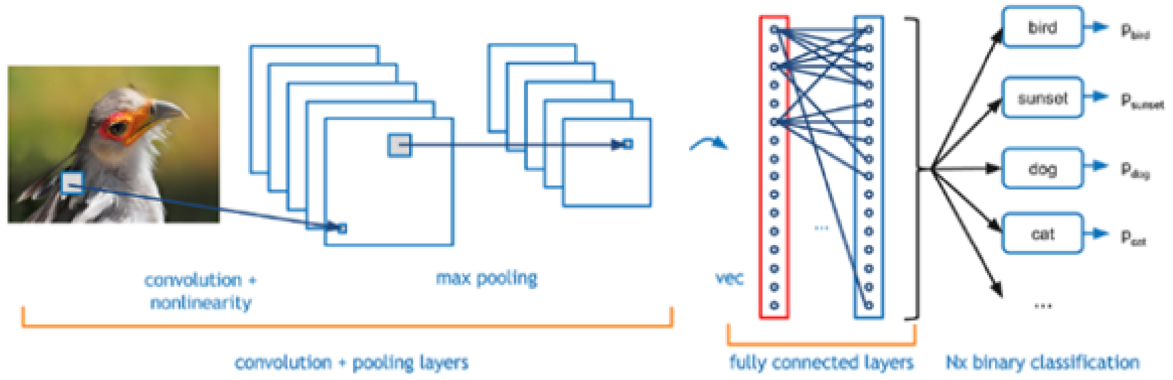


Fig. 9. A conceptual diagram for CNN. Image is being processed by multiple convolutional and pooling layers until a simplified representation is generated. The obtained representation, that goes into fully connected layers, is the main focus of my research. Source: (Shahid and Channappayya, 2021)

The grid pattern embedded in CNN allows learning of spatial hierarchies of features using up and down patterns. The mathematical construct has three layers, where convolution and pooling layers are responsible for feature extraction, while the fully connected layer performs classification by mapping the extracted features into output(Yamashita *et al.*, 2018). Figure 3 illustrates that the stack of mathematical operations in the convolution layer completes the linear operation. This layer stores pixel values in a two-dimensional grid in the kernel, a small grid of parameters while a loss function in forwarding propagation on learnable parameters and training dataset compute model's performance under different kernels and weights(Yamashita *et al.*, 2018). In this context, CNN is highly reliable for image processing due to applying an optimizable feature extractor to all image positions. The kernel optimization parameters constitute training aimed at constraining variances between outputs and ground truth labels(Yamashita *et al.*, 2018). Effective training involves different optimization algorithms, including backpropagation and gradient descent.

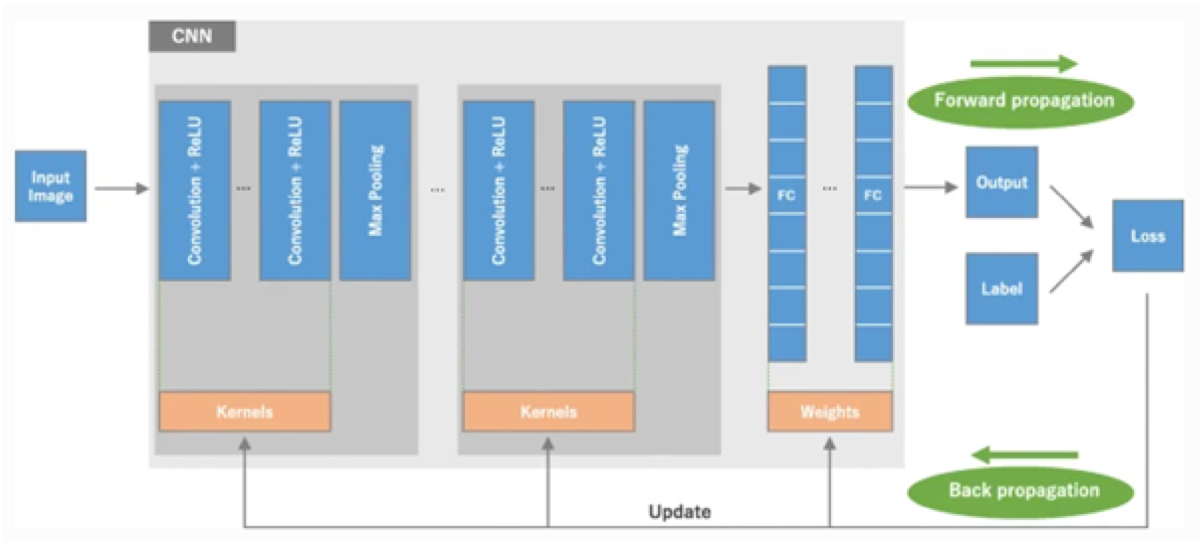


Figure 10. The training process of a CNN. In order to train a classification network we have to process through it hundreds, thousands or even millions of images and propage loss (difference between expected and received output) back to adjust weights in each hidden layer. Source: (Liu *et al.*, 2015)

Gradient backpropagation is a popular and effective network optimization algorithm. The technique involves computing the gradient of the model's loss with respect to the weights. The gradient is a vector that contains a value for each weight, reflecting how much a slight change in that weight will affect the output, essentially describing which weights are most important for the loss.

This single step through the network gives important values for each pixel, which is displayed in a heatmap. The enhancement of the method to a guided backpropagation blocks the backward flow of gradients from neurons whenever the output is negative, leaving only those gradients that result in increased output, which ultimately results in a less noisy interpretation. Since it only requires one pass through the network and produces a cleaner output, especially around the object’s edges, it became the preferred technique over occlusion, which is relatively slow and resource-intensive. However, the gradient backpropagation has a major issue in that it does not work well when there is more than one class of object present in the image.

To solve this issue, novel Class Activation Maps and their enhancements (Zhou *et al.*, 2016; Selvaraju *et al.*, 2017; Chattopadhay *et al.*, 2018) are introduced. Here the authors found that the quality of interpretations improved when the gradients were taken at each filter of the last convolutional layer instead of at the class score. The results yield faster than occlusion and better than gradient backpropagation as it is targeted, therefore highlighting the most significant areas.

2.2 CNN Development History

CNN is a deep learning architecture inspired by the visual perception of living creatures. In 1959, Hubel and Wiesel in 1959 found that visual cortex cells of animals are receptive to light(Hubel and Wiesel, 1968). Subsequently, Kunihiko Fukushima used the findings in 1980 to propose a neocognitron, which became a CNN predecessor (Fukushima and Miyake, 1982). Meanwhile, LeCun and colleagues published the modern framework in 1990 (LeCun *et al.*, 1990). The LeNet-5 model was a multi-layer artificial neural network that could effectively classify handwritten digits after training with the backpropagation algorithm (Gu *et al.*, 2018). LeNet-5 obtained original image representations, envisioning recognition of visual patterns from raw pixels with little-to-none preprocessing. Meanwhile, the evolution of the model to the shift-invariant artificial neural network (SIANN) enhanced character recognition from images, but limited computing power and the absence of extensive training data challenged its effectiveness (Gu *et al.*, 2018). As a result, SIANN was ineffective for complex problems, such as video classification and large-scale images.

The Hubel and Wiesel model designed several brain functions, particularly the primary visual cortex (PVC). Goodfellow and colleagues posit that integrating PVC function in a convolutional network entails defining features in 2D maps to reflect the 2D structure (Goodfellow, Bengio and Courville, 2017). The design of convolutional network units should copy PVC simple cells by having a small and spatially localized receptive field (Namatēvs, 2017). Additionally, PVC has numerous complex cells that respond to features, which inspire the pooling of CNNs. In this context, neocognitron was the first artificial NN to incorporate all the neurophysiological models of PVC by articulating alternating downsampling and convolutional layers. The neocognitron architecture was a feedforward, supervised, and gradient-based deep learning network. Meanwhile, the backpropagation network of 1986 allowed the training of a NN with one or two hidden layers (Namatēvs, 2017). Backpropagation algorithms foster learning by enabling CNNs to change weights according to the target(Liu *et al.*, 2017; Khan *et al.*, 2020). The current deep learning renaissance emerged in 2006 when an NN outperformed the radial basis function (RBF) kernel on the MNIST benchmark. Thus, the advancement of deep CNNs (DCNNs) is attracting intense interest in classification tasks, chiefly after ILSVRC classified approximately 1.2 million images into 1000 classes(Rawat and Wang, 2017). Subsequently, GPUs implementation of feedforward neural networks (FNNs), field-programmable gate arrays (FPGA), and digital signal processing (DSP) are popular computational resources for training during the execution of the eccentric patterns of DCNNs (Namatēvs, 2017). Meanwhile, DCNN functions similar to the neocortex in the human brain, dynamically learning features from the raw data (Yamashita *et al.*, 2018). The hierarchical feature extraction ability ensures DCNNs have access to low, mid, and high-level features. In this context, high-level features are abstract topographies obtained from combining low- and mid-level features.

2.3 Common CNN Modifications to Improve Performance

CNN architectures involve numerous variants with similar basic components. For instance, the popular LeNet-5 entails the convolutional layer has several convolution kernels for mapping and learning feature representations; pooling layer placed between two convolutional layers that deliver shift-invariance by reducing the resolution of the feature maps and fully-connected layers for performing high-level reasoning, which can be replaced by 1 × 1 convolution layer(Gu *et al.*, 2018). Meanwhile, CNN's are open to different types of improvements across the layers (Szandała, 2021). The generalized linear model (GLM) constitutes a convolution filter that initiates abstraction when latent concepts are linearly separable. In this regard, CNN may embrace tiled convolution to learn rotationally and scale-invariant features when weight-sharing mechanisms drastically decrease parameters (Gu *et al.*, 2018). Meanwhile, transposed convolution associates a single activation with multiple output activations to articulate super-resolution (Dong *et al.*, 2016), visual question answering (Das *et al.*, 2017), semantic segmentation (Noh, Hong and Han, 2015; Pham, Pham and Dang, 2020), localization(Zhou *et al.*, 2016), recognition (Zhang, Lee and Lee, 2016), and visualization(Zeiler and Fergus, 2014). The improvement of the convolution layer may also entail dilated convolution, network in network, and inception module.

Pooling is a crucial concept in CNN that reduces the computational burden by decreasing connections between convolutional layers. The critical improvements in the layer entail L_p pooling that guarantees better generalization than max-pooling (Gu *et al.*, 2018). Mixed pooling combines max pooling and average pooling to address the overfitting problems. Stochastic pooling is a dropout-inspired pooling method that randomly picks the activations using multinomial distribution to promote the utilization of non-maximal activations of feature maps (Zeiler and Fergus, 2013). Meanwhile, spectral pooling enhances the pooling layer by cropping the input representation in the frequency domain to reduce dimensionality. Spatial pyramid pooling produces a fixed-length representation regardless of the input sizes, while multi-scale orderless pooling enhances the invariance of CNNs and preserves the discriminative power (Gu *et al.*, 2018).

The use of a suitable activation function enhances CNN performance. In this regard, tasks are embracing different activation functions to achieve varying performance. Rectified linear unit (ReLU) is a popular non-saturated activation function and piecewise linear function that preserves the positive part while truncating the negative aspect to zero (Nair and Hinton, 2010). The simple max() operation allows a network to swiftly obtain sparse representations, inducing sparsity in the hidden units. Thus, ReLU is efficient for training deep networks without pre-training but the discontinuity at 0 challenges the performance of backpropagation. The problem is solved using Leaky ReLU that compresses the negative part rather than mapping it to a constant zero, allowing a small and non-zero gradient when the unit is not active. The Parametric Rectified Linear

Unit (PReLU), rather than compressing negative parts, adaptively learns the parameters of the rectifiers to improve accuracy while Randomized Leaky Rectified Linear Unit samples the parameters of negative aspects from a uniform distribution in training before fixing them during testing. Meanwhile, other popular activation functions with higher classification accuracies include Exponential Linear Unit and Maxout(Szandała, 2021).

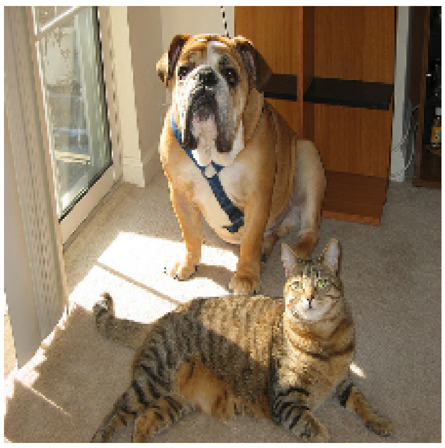
The accuracy of specific CNN tasks relies on the selection of an appropriate loss function. In this context, hinge loss trains classifiers with a large margin, such as Support Vector Machine (SVM). In contrast, Softmax loss combines multinomial logistic loss and softmax that turns predictions into non-negative values and normalizes them to obtain a probability distribution over classes (Tang, 2013). Contrastive loss is prevalent for training Siamese networks that entail a weakly supervised scheme for learning a similarity measure from pairs of matching or non-matching data instances (Bromley *et al.*, 1993; Chopra, Hadsell and LeCun, 2005). Meanwhile, Triplet loss considers three instances per loss function: negative, positive, and anchor instances (Schroff, Kalenichenko and Philbin, 2015). Kullback-Leibler Divergence (KLD) is a common non-symmetric measure of information loss in the objective function of various autoencoders (AEs) (Mehta and Majumdar, 2017), including sparse AE (Lee *et al.*, 2007), denoising AE (Vincent *et al.*, 2008) and variational AE (VAE) (Kingma and Welling, 2013). The method, when placed symmetrically, becomes Jensen-Shannon Divergence, which is highly effective for the Generative Adversarial Networks.

DCNNs significantly suffer from the overfitting problem that is effectively managed through regularization. This approach modifies the objective function through additional terms that castigate the model complexity. In this context, the l_p-norm regularization function is a weight decay that enhances optimizations and attractiveness by exploiting the sparsity effect of the weights. In contrast, Tikhonov regularization is popular for rewarding invariance to noise in the inputs (Bertero, De Mol and Viano, 1980). Meanwhile, dropout is highly effective for overfitting when applied to the fully connected layers (Hinton *et al.*, 2012). The method prevents a network from depending on any combination of neurons, leading to enhanced accuracy even without specific information. The efficiency of dropout is enhanced by DropConnect that randomly sets the weight matrix to zero rather than the outputs of neurons to zero. In this regard, optimizing CNNs can further be enhanced through data augmentation, weight initialization, stochastic gradient descent, shortcut connections, and batch normalization (Szandała, 2021).

2.4 CNN Visualization Methods

The deep networks are prevalent in computer vision but suffer from a critical challenge due to the failure to provide visual output. In this context, it is almost impossible to identify the part with crucial influence on the output despite improving different layers and techniques. As a result, the demand for explaining or visualizing the decision-making process in networks is increasing (Das *et al.*, 2017; Selvaraju *et al.*, 2017; Szandala, 2021a). The popular methods have been evolving from occlusion, gradient backpropagation, original CAM to Grad-CAM. Also model-agnostic methods has been proposed like LIME, LRP or SHAP. Meanwhile, Grad-CAM has several derivatives, including Grad-CAM++, full Grad-CAM, and Excitation Backpropagation. All the techniques, excluding occlusion and gradient backpropagation, generate a heatmap or a stain that marks the most significant area. However, the computation only involves a single chosen convolutional layer to mean that the reliability of a given saliency map increases with depth (Szandala, 2021a). Thus, extrapolation is essential for visualizing the obtained map on a referenced image. Nonetheless, output tends to be a low resolution because the deepest layers have the most miniature convolutional masks.

2.4.1 Occlusion



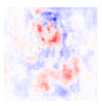
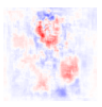
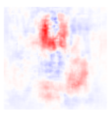
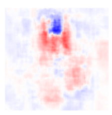
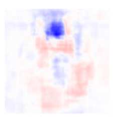
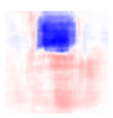
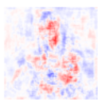
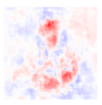
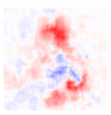
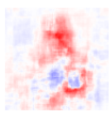
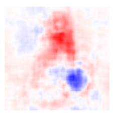
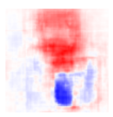
Patch size	10x10	15x15	25x25	35x35	45x45	90x90
"boxer" sensitivity						
"tiger cat" sensitivity						

Fig. 11. Generated the occlusion sensitivity map from the image of cat and dog (above) based on logit scores. The red and blue regions indicate a relative increase and decrease from non-occluded scores respectively: **the blue regions are vital!** Source: (Springenberg *et al.*, 2014)

The most straightforward method for determining a model's attention focus is the occlusion. We hide a small part of the image and study the output. The more it has changed from the original, the more significant the given area was. The most notable setback is the speed of this method. We have to analyze the image multiple times, once for each occluded region. The second limitation is that it does not take into account interdependencies between regions. The most notable application of occlusion has been done by Sameer Singh and his team during implementation of husky vs wolves distinguishing model(Ribeiro, Singh and Guestrin, 2016). Method has pointed to the researchers that the network is learning the presence of snow in pictures with wolves and lack of snow in images with huskies. However, without processing the occluded image dozens or even hundreds of times, occlusion is useless, therefore more sophisticated methods have been proposed.

2.4.2 Deconvnet

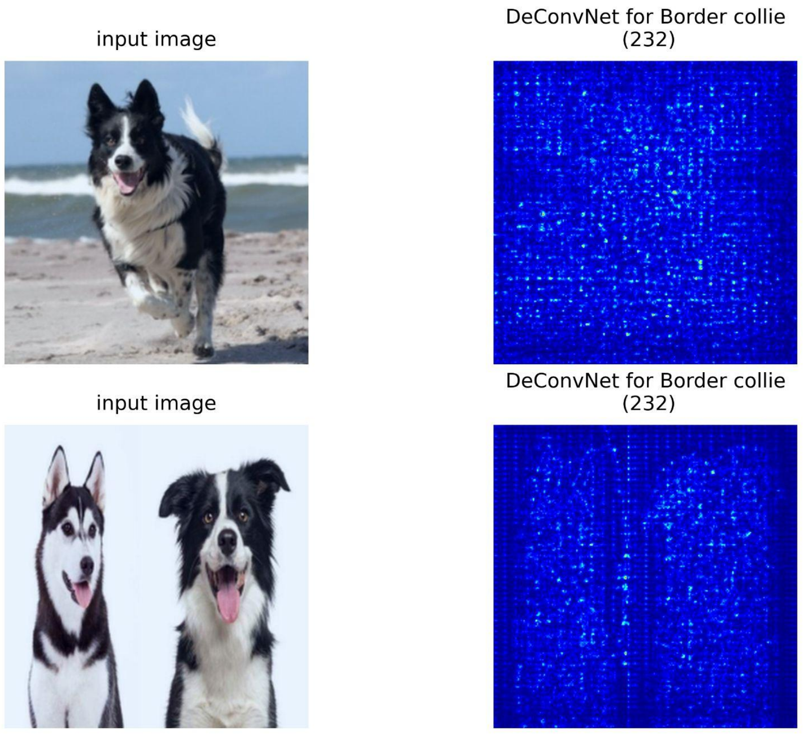

input image
DeConvNet for Border collie (232)
input image
DeConvNet for Border collie (232)

Fig. 12. Application of deconvolution network. It due to coarseness of output it provides a useable output only for backgroundless images

Another, simple in concept, method is performing deconvolution on a network(Zeiler and Fergus, 2014). The deconv operation is defined as the inverse of the convolution operation, i.e. change the input data from a 2×2 matrix into an output matrix with the shape 3×3 in our example. The deconv operation does not guarantee that we will have the same values in the output as the original matrix.

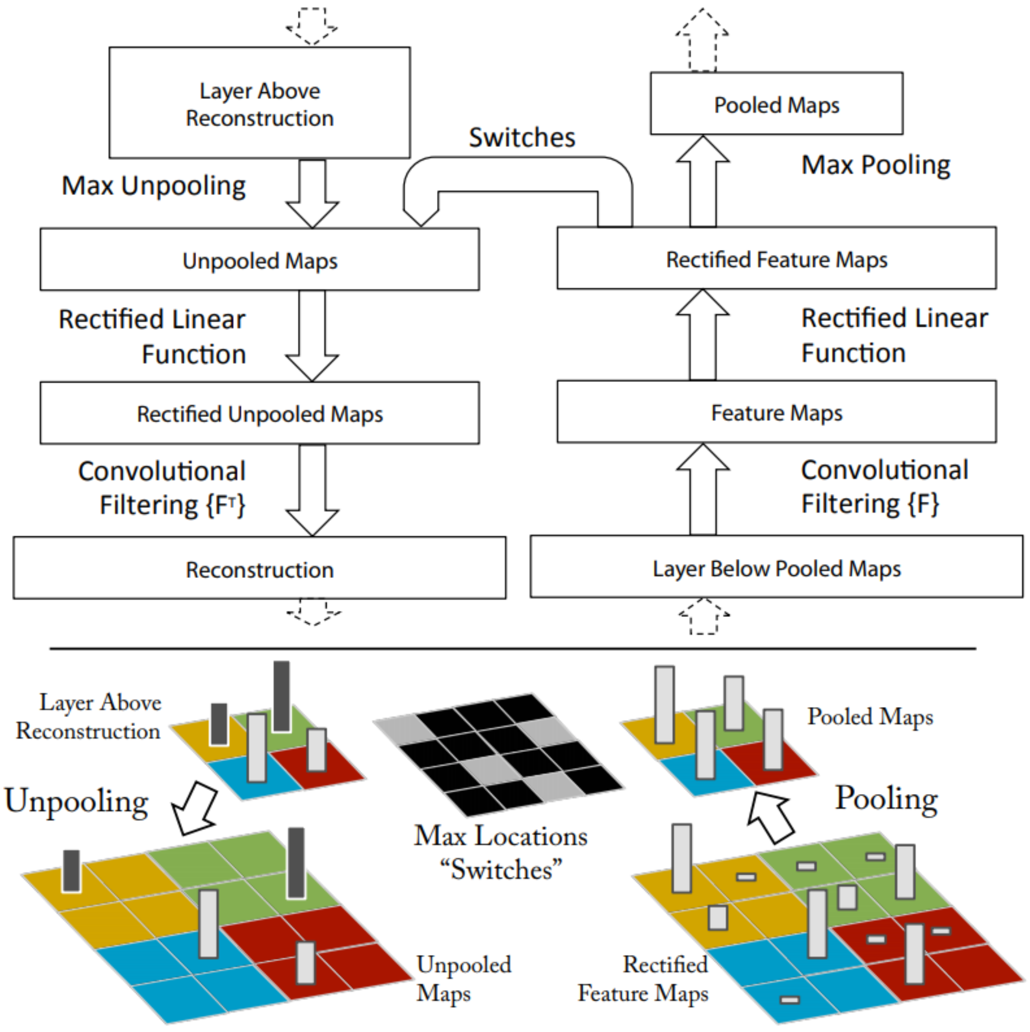


Fig. 13. Convolution and deconvolution procedure

To visualize a convolution network, a deconvolution network (deconvnet) (Yu *et al.*, 2016) is attached to each of its layers, providing a continuous path back to image pixels. To start, an input image is presented to the convnet and features are computed throughout the layers. To examine a given convnet activation, we set all other activations in the layer to zero and pass the feature maps as input to the attached deconvnet layer. Then we successively unpool, rectify and finally filter to reconstruct the activity in the layer beneath that gave rise to the chosen activation. This is then repeated until input pixel space is reached.

2.4.3 Gradient Backpropagation

The next group of noteworthy methods rely on gradient backpropagation. Here we compute the gradient of the model's loss with respect to the weights. The gradient is a vector that contains a value for each weight, reflecting how much a small change in that weight will affect the output, essentially telling us which weights are most important for the loss.

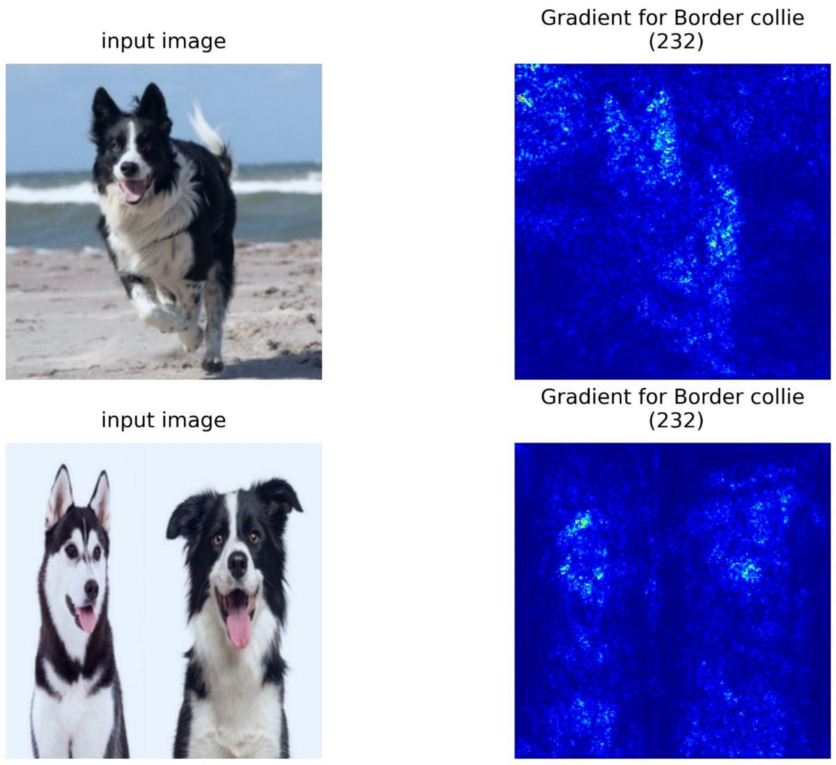

input image
Gradient for Border collie (232)
input image
Gradient for Border collie (232)

Fig. 14. Gradient backpropagation for two pictures. Very poor readability as we can only guess objects the network has focused on

This single step through the network gives us an importance value for each pixel, which we display in the form of a heatmap. The enhancement of the method, guided backpropagation, blocks the backward flow of gradients from neurons whenever the output is negative, leaving only those gradients that result in increased output, which ultimately results in a less noisy interpretation. Since it only requires one pass through the network and produces a cleaner output, especially around the edges of the object, it became preferred over occlusion.

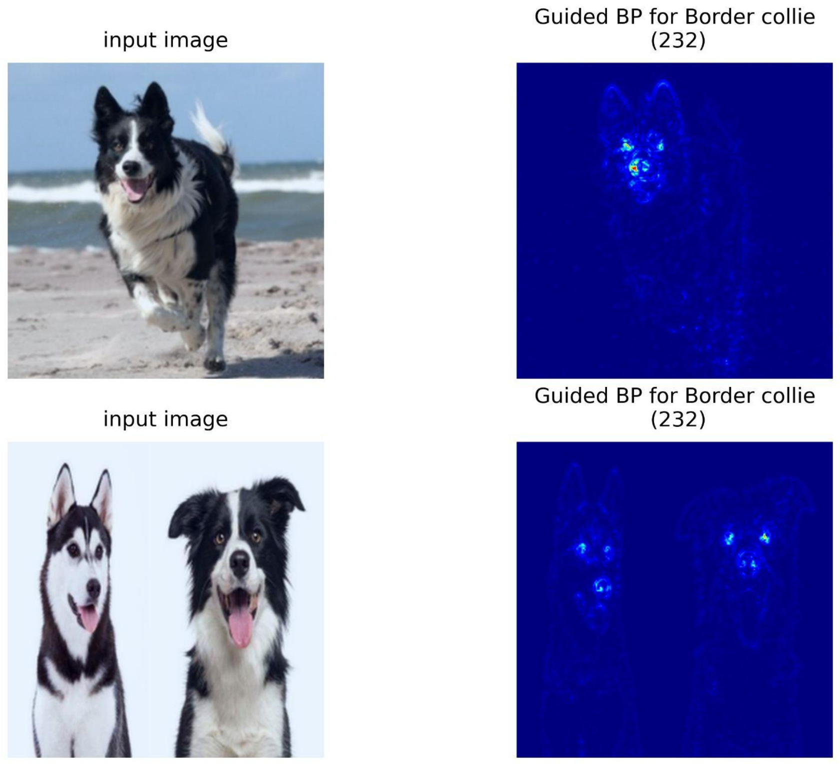

input image
Guided BP for Border collie (232)
input image
Guided BP for Border collie (232)

Fig. 15. Guided backpropagation output - a visible improvement over Gradient Backpropagation as we can clearly recognize shapes. Unfortunately lack of discrimination between Border collie and husky

However, a major issue with gradient backpropagation is that it does not work well when there is more than one class of object present in the image.

2.4.4 Class Activation Maps and Derivatives

The original CAM approach modifies the training model to enhance the visualization of the decision-making process in CNNs. Chattopadhay (Chattopadhay *et al.*, 2018) notes that the technique discards all fully connected layers at the top of a network before introducing a global average pooling (GAP) layer that is succeeded by a single fully connected layer. Using a convolution layer with network architecture similar to GoogLeNet or ImageNet allows GAP on the convolutional feature maps and utilization of features on the fully-connected layer to generate the desired output. The simplified connectivity structure produces a stain of important regions by projecting weights of the output layer onto the convolutional feature maps (Zhou *et al.*, 2016). The modification generates a streamlined architecture, enabling CAM to produce a heatmap for all the output classes. The original CAM approach sums the last convolutional layer activations by rescaling weights of the newly introduced fully connected layer associated with the selected class (Szandala, 2021a). Nonetheless, CAM requires a GAP layer in architecture, and it is only effective for visualizing the final layer heatmap.

When we see the heatmaps of different layers we see that different parts of the image are being activated. This is because earlier layers have filters which are looking at various parts of the image. But the class prediction largely depends on the activations of the layers just before prediction and hence, visualizing heatmaps of the layer just before fully connected layer answers which portion of the input image led to a particular classification.

In order to obtain Grad-CAM (eq. 1) of width *u* and height *v* for any class *c*, we first compute the gradient of the score for class *c*, y^c (before the softmax), with respect to feature map activations A^k of a convolutional layer, i.e. $\partial y^c / \partial A^k$. These gradients flowing back are global-average-pooled over the width and height dimensions (indexed by *i* and *j* respectively) to obtain the neuron importance weights α^c_k.

$$\alpha^c_k = \frac{1}{Z}\sum_i\sum_j \frac{\partial y^c}{\partial A^k_{ij}} \qquad (1)$$

Where y^c is defined as (eq. 2):

$$y^c = \sum_k w^c_k \frac{1}{Z}\sum_i\sum_j A^k_{ij} \qquad (2)$$

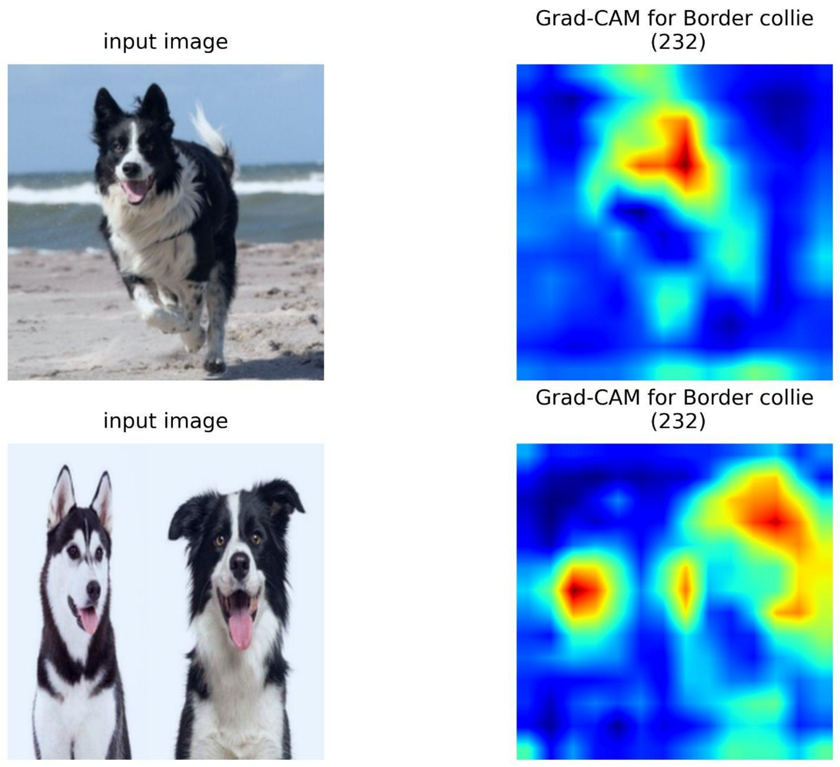

input image
Grad-CAM for Border collie (232)
input image
Grad-CAM for Border collie (232)

Fig. 16. Grad-CAM output for 2 images. We gain discrimination between husky and Border collie at the loss of sharpness in the output

The gradient weighted CAM (Grad-CAM) addresses the limitation of CAM on network architecture. The technique delivers an improved localization effect on salient features by eliminating GAP restrictions and visualizing any layer in network heatmaps [62]. Zhou, Selvaraju et al. (Zhou *et al.*, 2016; Selvaraju *et al.*, 2017) posits that Grad-CAM generalizes CAM to any deep neural network architecture, allowing utilization in networks of diverse ranges while eliminating the need to modify network architectures. Meanwhile, Grad-CAM is similar to CAM in that it generates heatmaps by summing output channels' weighted average using the weights of the fully connected layer associated with the output class. However, Matcha (Matcha *et al.*, 2020) notes that the technique has a distinct approach of generating weights that involves computing the gradient of the output class value associated with each channel in the feature map of a given layer. The results are a gradient channel map that represents the corresponding channel's gradient in a feature map. The subsequent process in GAP of gradient channel map to obtain critical weights of each channel in the feature map, which functions as the heatmap. Nonetheless, Grad-CAM does not identify all the class instances if an image has more than one instance of a single class present (Chattopadhay *et al.*, 2018).
Grad-CAM++ solves the limitation of Grad-CAM by only considering the positive values and ignoring negative values when calculating the GAP of the gradient channel map. The technique provides enhanced visual explanations of CNN model predictions due to improved object localization and explains multiple object instances in a single image [64]. The guided backpropagation in Grad-CAM++ allows disregarding of the negative gradients while enabling the utilization of a ReLU on top of each value of the gradient channel map output (Chattopadhay *et al.*, 2018). Meanwhile, positive gradients only allow the identification of pixels with a positive impact on the class output. In this context, considering negative values neutralizes some positive values in the gradient channel, leading to the loss of valuable information. Thus, Grad-CAM++ calculates higher-order derivatives to improve the localization accuracy of Grad-CAM, chiefly when an image has several occurrences of the same objects (Szandala, 2021a). In another improvement instance, Smooth GradCAM++ embraces a smooth gradient function that smoothens the gradients obtained using Grad-CAM++, leading to sharpened heatmaps (Smilkov *et al.*, 2017; Pan *et al.*, 2020). However, the technique creates multiple noisy versions of the same input image due to image augmentation.

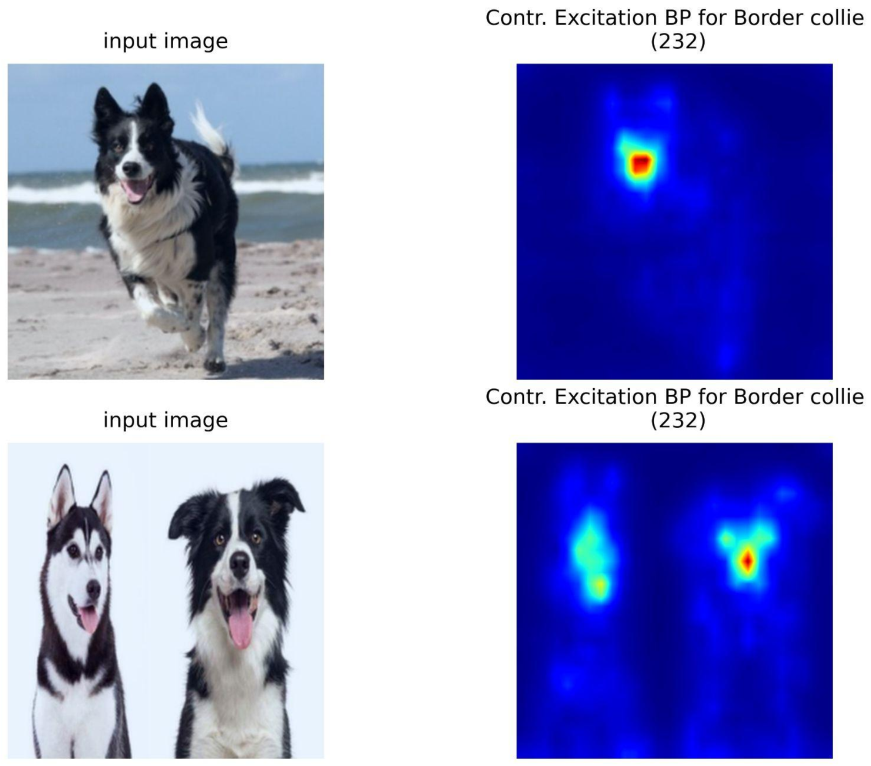

input image
Contr. Excitation BP for Border collie (232)
input image
Contr. Excitation BP for Border collie (232)

Fig. 17. Contrastive Excitation Backpropagation. An improvement over Grad-CAM with notably better focus localisation and partial shapes restoration

The excitation backpropagation proposes a unique approach in explaining the decision-making process in CNNs. The intuitively simple technique provides empirically effective explanations, which entails passing top-down signals downwards in the network hierarchy through a probabilistic Winner-Take-All process (Zhang *et al.*, 2018). Excitation Backprop efficiently calculates the winning probability of each neuron by integrating bottom-up and top-down information. Meanwhile, the technique generates interpretable attention maps at intermediate convolutional layers, preventing a complete backward sweep (Zhang *et al.*, 2018). Thus, excitation backpropagation leverages enhanced information available in a network and produces soft attention maps that effectively capture elusive differences between top-down signals. Nonetheless, the method ignores the nonlinearities in the network backward pass and generates heatmaps that preserve the evidence for or against a network class predicted (Szandala, 2021a). In this context, the excitation backpropagation delivers a visually sharp contrastive variant by subtracting evidence against a certain class from the evidence for that class. Although the approach guarantees a higher visualization accuracy than different CAMs, it only highlights critical regions rather than the object itself.

2.4.5 Guided Grad-CAM - Example of Combination of Methods

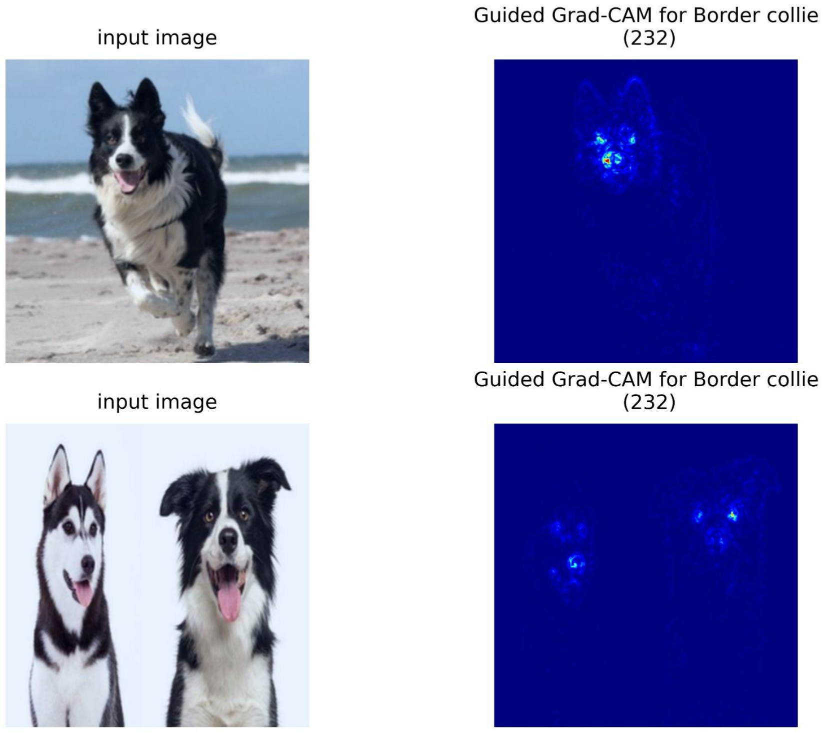

input image
Guided Grad-CAM for Border collie (232)
input image
Guided Grad-CAM for Border collie (232)

Fig. 18. Guided Grad-CAM, a combination of Grad-CAM and Guided Backpropagation which results in a recognizable objects that were significant for classification

Guided Grad-CAM, a combination of two previously described techniques where we multiply output from Guided Backpropagation (provides visible shapes of the objects) with output from Grad-CAM (provides saliency of given area for classification). A good example for effectiveness of combinations of two separate approaches.

2.4.6 Saliency Maps Summary

Saliency methods elucidate the predictions obtained using DCNN and create post-hoc explanations of image classifier outputs. However, the approaches are unreliable for insensitive explanations of the object shape, which sometimes contributes to the factual factor of a model prediction (Kindermans *et al.*, 2019). In this regard, saliency methods aim at inferring insights of a learned function by the model when ranking the explanatory power of constituent inputs. The approaches focus on computing the area in the deep layer of the network, meaning that projecting the deep map to the original image considerably reduces accuracy and omits essential traits of the classification reasoning (Szandala, 2021a). Saliency methods estimate the relevance of each pixel using the classification output score, which is displayed as a saliency map with highlights of essential pixels (Szandała, 2020b; Tomsett *et al.*, 2020). Nonetheless, the methods do not correctly attribute a constant vector shift applied to the input. From this perspective, different techniques used in explaining visualization in CNNs are sometimes dedicated to limited application constituting distinct pros and cons.

2.4.7 Layer-wise Relevance Propagation

Geirhos et al. (Geirhos *et al.*, 2020)mention that the development of deep neural networks has prompted interest in interpreting deep neural networks forecasts in medicine. Layer-wise relevance propagation (LRP) is one of the outcomes. LRP is a technique for calculating scores for image pixels and image sections, which indicate the effect of each image segment on the classifier's estimate for a single trial image(Karim *et al.*, 2020). A profound neural system is a feed-forward chart of essential computational neurons performing a single task. The interconnection of many of these simple components and the accessibility of an effective method for learning the model gives a deep network its complexity (error backpropagation) (Seibold, Hilsmann and Eisert, 2021). In a feed-forward pass, the productivity of a deep neural network is derived by assessing these neurons. Ideally, LRP is a decomposition method that creates a relevant heatmap that meets the intended conservation property when applied in a backward pass.

Bohle et al. (Böhle *et al.*, 2019) propose employing LRP to show CNN choices for Alzheimer's disease based on MRI facts in this context. Abnormal cell death is a mark of Alzheimer's disease, particularly in the medial temporal lobe. LRP creates a heatmap in the input space, similar to other visualization approaches, displaying the value or significance of each pixel leading to the last classification consequence. The LRP approach can directly highlight favorable additions to the input space's system organization. LRP appears to offer a lot of promise to aid physicians in understanding neural network choices for detecting Alzheimer's disease and maybe other conditions based on structural MRI data.

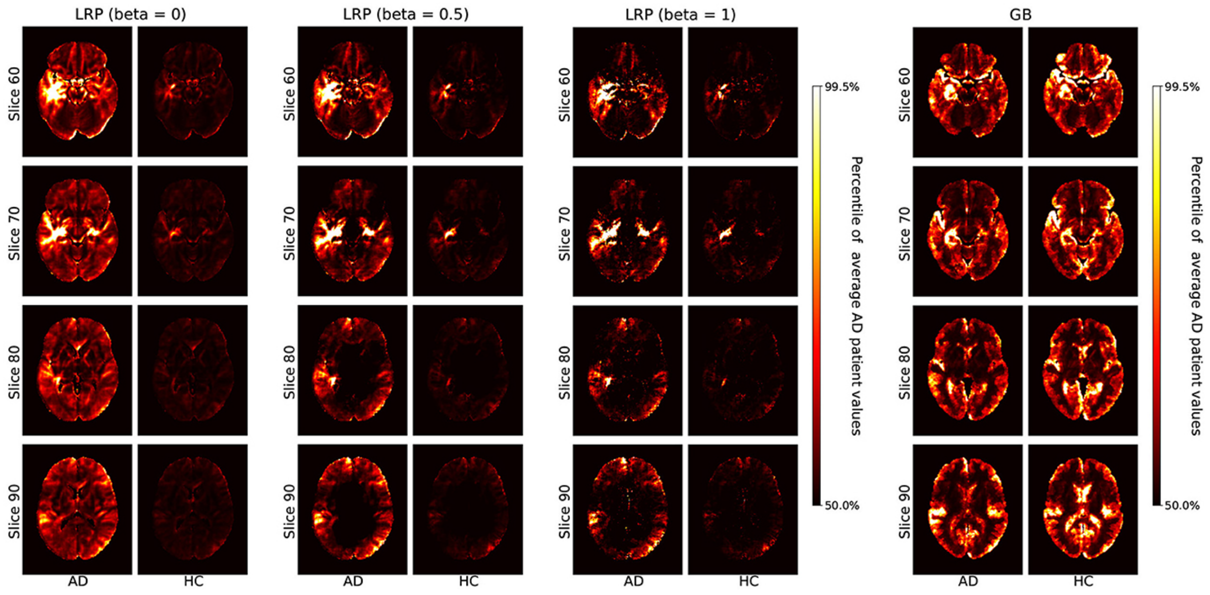


Fig. 19. LRP output for different values of beta parameter. Source: (Böhle *et al.*, 2019)

It is also worth noting that by propagating the forecast backwards through the model using various rules, LRP can explain SOTA predictions in terms of its input attributes. These can be implemented efficiently and modularly, and even complex models can have high-quality explanations because of parameter adjustment. Furthermore, LRP may be generalized beyond DNNs to other model types via Neuralization-Propagation (NEON), broadening its use to many different cases that demand explainable machine learning solutions (Montavon *et al.*, 2022). However, there are a number of considerations against using LRP. For example, there is limited empirical evidence and no comparison to other SOTA approaches. In addition, numerous LRP regulations were left out, for reasons that are unclear. Above all, human evaluation of explanation quality is still required, which can be time-consuming and error-prone. Worse, there are no defined evaluation standards; instead, only integrity and understandability are used, which are both biased.

2.4.8 LIME - Local Interpretable Model-agnostic Explanations

Trust is essential when acting on a projection or selecting whether or not to deploy a new model(Nagendran *et al.*, 2020). Such knowledge also helps to provide perceptions into the framework, which can help to turn an undependable prediction model into a reliable one. Ribeiro et al. (Ribeiro, Singh and Guestrin, 2016) state that LIME is a unique clarification system that studies an interpretable model around the forecast to explain any classifier's guesses in an interpretable and authentic version. Simply said, LIME is a method that can accurately describe any classifier or regressor's predictions by approximating them locally with an interpretable model(Lundberg *et al.*, 2020).

The abbreviation of LIME itself should give you an intuition about the core idea behind it. LIME is:

- Model agnostic which means that LIME is able to explain any black-box classifier you can think of.
- Interpretable, which means that LIME provides you a solution to understand why your model behaves the way it does.
- Local, which means that LIME tries to find the explanation of your black-box model by approximating the local linear behavior of your model.

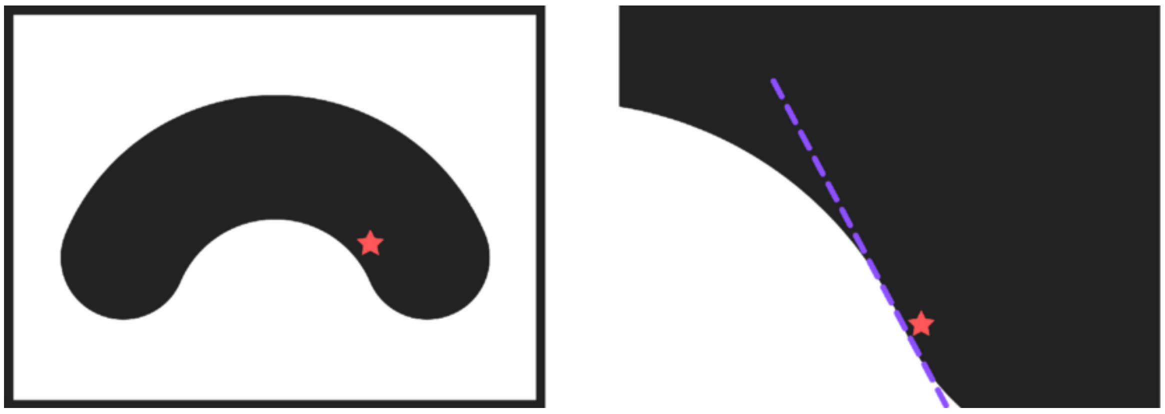


Fig. 20. Sample feature space of bi-classifier (left) where red star locates the object. LIME tries to create a linear classifier(right) to perform explanations

Let’s say you have a feature space with non-linear boundaries as shown in the left image above. The data points residing inside the black curve will be classified as A and otherwise, will be classified as B. Now we want to predict the class of a data point denoted by the red star in the above image. Instead of looking at the global behavior (left image), LIME will go into the vicinity area of the red star point such that it becomes very local that a linear classifier could explain your model’s prediction (right image).
Procedure is as follows:

1. The first step that LIME would do is to create several artificial data points that are close to the data denoted by the red star. Note if our input data is an image, LIME will generate several samples that are similar with our input image by turning on and off some of the super-pixels of the image.
2. Next, LIME will predict the class of each of the artificial data points that has been generated using our trained model.
3. The third step is to calculate the weight of each artificial data to measure its importance. To do this, first the cosine distance metric is usually applied to calculate how far the distance of each artificial data point with respect to our original input data. Next, the distance will be mapped into a value between zero to one with a kernel function. The closer the distance, the closer the mapped

value to one, and hence, the bigger the weight. The bigger the weight, the bigger the importance of a certain artificial data point.

4. The last step is fitting a linear regression model using the weighted artificial data points. After this step, we should get the fitted coefficient of each feature, just like the usual linear regression analysis. Now if we sort the coefficient, the features that have larger coefficients are the ones that play a big role in determining the prediction of our black-box machine learning model.

LIME may be used to create a classifier structure that categorizes tabular statistics, pictures, or transcripts. Surprisingly, the abbreviation LIME might provide insight into the basic concept. LIME is model agnostic, which means it can elucidate any black-box classifier and is model-independent. LIME is also interpretable and local, which means it can help you figure out why a model performs the way it does and find an explanation for a black-box framework by approximating its local linear behavior respectively.
Furthermore, LIME explains an estimate that even non-experts may use feature engineering to associate and advance an unreliable framework. According to Ferrando Piera and Urbano (Ferrando and Lorenzo-Seva, 2017), a good model explainer should have many desirable characteristics. It should, for starters, offer a qualitative comprehension of the relationship amid the input features and the feedback. In other words, it ought to be easy to understand. Second, unless an explanation contains a thorough description of the model itself, it may not be able to be completely accurate(Deramgozin *et al.*, 2021). It must, however, be at least locally realistic, i.e., it ought to replicate the representation's behavior in the area of the occurrence being projected. Moreover, the explainer should explain any model without making assumptions about it. Besides, the interpreter should give the user a representative set to present to have a general understanding of the concept. LIME uses a representation that people can understand regardless of the model's essential features; thus, the feature is an interpretable representation (Zhang *et al.*, 2019). The sort of data that people are working with will determine how interpretable a representation is. In a nutshell, LIME tries to relate a model's prediction to human-understandable attributes.
LIME is not free from setbacks. When employing this approach, one of the most serious concerns is LIME instability. Simply put, if one runs LIME on a single individual and repeats the procedure numerous times with the same parameters and settings, they can get completely different results. The generating process of LIME causes it to be unstable. Each call to LIME generates a new dataset because the points are produced at random (Visani, Bagli and Chesani, 2020). Individuals may end up with different models for different LIME calls since the dataset is utilized to train the LIME Linear Model. Instability is detrimental to LIME because it contributes to a lack of confidence. Experts have debated whether or not the Generation Step should be eliminated because it is the source of Instability. However, there may be numerous locations with very few training

points as a result of this, and the linear approximation will be quite rough in those areas. Ideally, the sampling stage goes a long way toward ensuring that the ML function is well-covered, as more points increase the model's accuracy.
Sample failure of LIME can be seen below(Visani *et al.*, 2022). Author's used LIME to explain machine decisions on loan rejection. Three times the same decision has been analyzed and this is the model's motivation explanation according to LIME.

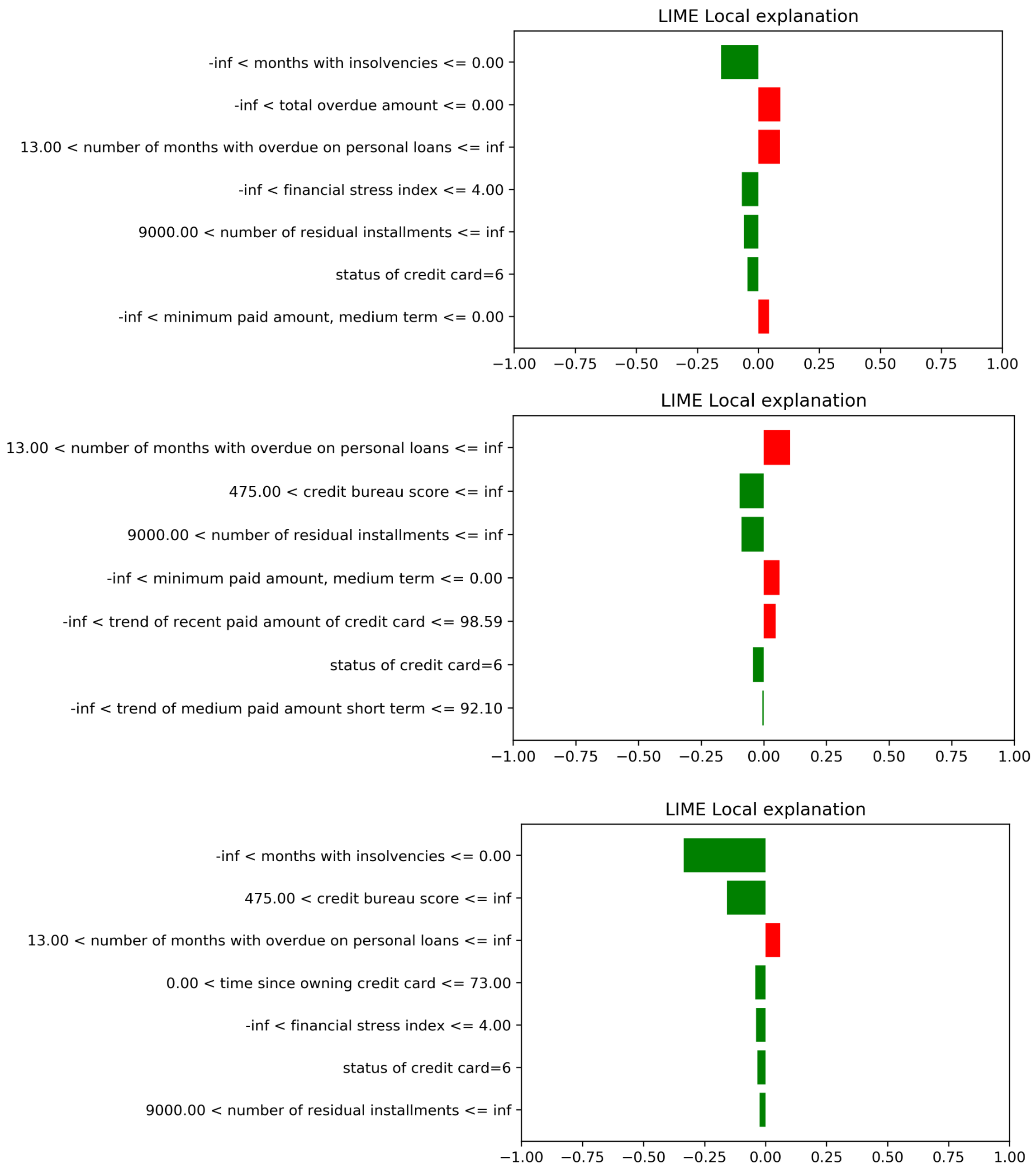


Fig. 21. results for LIME's explanation of loan decisions made by the ML model. Source: (Visani *et al.*, 2022)

2.4.9 SHAP - SHapley Additive exPlanations

In numerous instances, comprehending why an approach produces a detailed prediction is as significant as the exactness of the forecast. However, complex frameworks that even professionals labor to read, such as deep learning models, often attain the maximum precision for massive up-to-date data, generating a strain between precision and interpretability. Therefore, numerous approaches for supporting users in reading the estimates of difficult models have recently been offered (Biran and Cotton, 2017). However, it is occasionally uncertain how these approaches are linked and when one method is favored to another. Lundberg and Lee (Lundberg and Lee, 2017) suggest SHAP - SHapley Additive exPlanations, a combined paradigm for inferring guesses, to overcome this problem. For each prediction, SHAP assigns an essential value to each feature (Parsa *et al.*, 2020; Slack *et al.*, 2020). Its novel features include discovering a new session of additive feature significance actions and hypothetical studies signifying that this class has a sole answer with a set of favorite attributes. The novel course combines six existing approaches, which is remarkable as four of the class's most current approaches lack the projected needed abilities (Lundberg and Lee, 2017). Laconically, SHAP value estimate approaches are better matched with human intuition as judged by user studies and more effectively differentiate between model output classes than several existing methods.

There are a variety of approaches that can be used to improve model interpretability. One of the most common ways of expressing the model and understanding how your data's properties relate to the outputs is to utilize SHAP Values. A mechanism drawn from coalitional game theory allows the "payout" to be distributed fairly across the features. SHAP Values have the distinct advantage of providing both global and local interpretability (Lundberg *et al.*, 2020). Image classification tasks, for example, can be explained by the scores assigned to each pixel on a predicted image, which show how much that pixel helps to classify that image into a specific category. In addition, tabular data can represent a large amount of information. According to Fidel (Fidel, Bitton and Shabtai, 2020), SHAP Deep Explainer can tell which input feature contributes to the model output and the magnitude in a picture of connected neural networks. The summary map in the SHAP global interpretation program displays the most relevant elements and their level of influence on the model. It is an all-encompassing interpretation.

Both SHapley Additive exPlanations (SHAP) and LIME are additive and model-agnostic approaches for explaining individual predictions. According to Aas et al. (Aas, Jullum and Løland, 2021), SHAP seeks to explain model prediction for a given input by computing each feature's contribution to the prediction. SHAP achieves this goal by employing Shapley Values, which are derived from game theory. Shapley Values are a method of rewarding game players depending on their contribution to the overall profit. The concept has been transferred to SHAP as a technique of determining which of the attributes contributes the most to the model's final forecast. It is defined as the feature value's average marginal contribution over all feasible coalitions. Each attribute is assigned a value of 0 or 1, indicating if it is present in the coalition. In the end a function is used to map these vectors to the feature space; for images, the function fills in gray super-pixels with a vector value of 0.

2.4.10 Explanation by Example

The limitation of different methods encouraged enhanced evaluation of the method Explanation by Examples. Chen and colleagues (Chen *et al.*, 2018) in 2019 introduced the prototypical part network that dissects an image by finding prototypical parts and combines the prototypical evidence to formulate a final classification. The explanation-by-example is the preferred explanation style across applications spanning image, text, audio, and sensory domains by the average non-technical end-users (Jeyakumar *et al.*, 2020). The technique is not effectively grounded to make deterministic conclusions but provides an obvious choice to non-technical consumers for pointing to a set of images in a training set with a similar response to the examined one. Thus, nearest training examples present users with an opportunity to match features across similarly mapped ground-truth examples and test input.

The mapping of the explanation domain to visual domains is relatively straightforward in explanation-by-example frameworks because they generate several examples as an explanation. Jeyakumar et al. (Jeyakumar *et al.*, 2020) indicate that visualization of generated examples is similar to the visualization of data instances. In this regard, the researchers utilized ExMatchina, an open-source implementation of Explanation-by-Example, to access the nearest matching data samples for representative examples by comparing feature activations found in the last convolutional layer. Subsequently, explanation-by-example generates identical examples to the test input, whereby image and audio domains repeatedly offer nearest training examples that are intuitive and semantically similar to the mapping of training data, associated labels, and model inference (Jeyakumar *et al.*, 2020). The technique only requires configuring parameters in the model layer to generate explanations rather than fine-grained hyperparameters tuning. However, the efficiency of the Explanation-by-Example significantly relies on the quality of the training data, while lack of similar examples leads to subpar explanations. The method also increases the risk of exposing personally identifiable information, mainly when operating on sensitive training data.

Nonetheless, Explanation by Example was the motivation to create PRISM (see chapter *3.4 Principal Image Sections Mapping*) and merge it with GE. This idea enhances Explanation by Example by highlighting inclusive and exclusive features in the form understandable by the human reader.

2.5 CNN Models Training Modification Based on Explainability

Object recognition models are usually trained to minimize loss on a given dataset, and evaluated by the accuracy they achieve on the corresponding test set classification. In this paradigm, model performance can be improved by incorporating any generalizing correlation between images and their labels into decision-making. However, the actual model reliability and robustness depend on the specific set of correlations that is used, and on how those correlations are combined. Indeed, outside of the training distribution, model predictions can deviate wildly from human expectations either due to relying on correlations that humans do not perceive(Jacobsen *et al.*, 2018; Jetley, Lord and Torr, 2018), or due to overusing correlations, such as texture(Geirhos *et al.*, 2018), color(Szyc, 2020), background(Ribeiro, Singh and Guestrin, 2016; Xiao *et al.*, 2020), that humans do use. Characterizing the correlations that models depend on thus has important implications for understanding model behavior, in general.

Previously described methods for visualization are designed to explain specified cases, thus are hardly usable for evaluation of models taught on hundreds or thousands of images. The first problem that comes to our mind is the influence of background. In order to identify this bias Xiao et al. (Xiao *et al.*, 2020) proposed a special pre-fabrication of test dataset, that cuts out actual objects from the base picture thus generating 2 images: only background and only object. If the model classifies correctly the sole objects, then it has learnt the correct features. On the other hand, if the model does not identify only the background as the original object, we may conclude that the network is not biased towards the background.

3 Proposed Methods and Procedures

During the last few years I have designed and described several novel methods that could help CNN practitioners in explaining and evaluating their models.

3.1 Attention Focus Evaluation

The proposed method is based on three simple concepts. The first is to determine what the network should consider as a discriminative feature. Secondly, we have to go through and discover what the network is actually learning. And finally, we need to evaluate how much of the learned correlations are tied to the expected area of the image.

The first phase is the generation of the Regions-of-Interest (ROI). For this purpose, we can employ an expert that will manually indicate ROI, but in this work we used the Detectron2 framework(Wu *et al.*, 2019). It is Facebook AIResearch's next-generation software system that uses state-of-the-art object detection algorithms. It is also a common practice to use a base model pre-trained on a large image set, such as ImageNet(Deng *et al.*, 2009), as the feature extractor part of the network. Detectron2 framework allows us to obtain coordinates of two points that mark the top left and bottom right of the ROI. For the purpose of this research, we did not pay attention to which class object has been recognized.

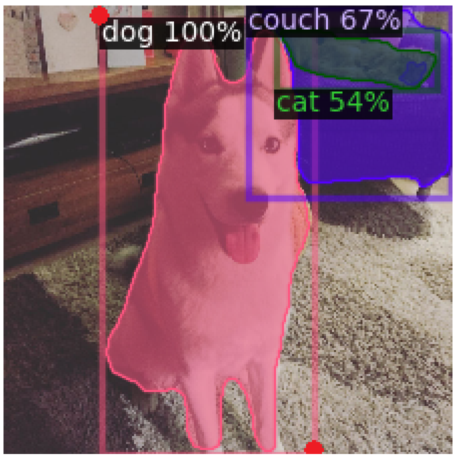


Fig. 22. Detectron2 processed image with highlighted 3 regions of interests

Detectron2 produces images with highlighted ROIs (see fig. 1) and returns a tuple for each image that contains 4 coordinates: x and y for the top left corner and x and y for the bottom right corner of the ROI.

Saliency map generation is the subsequent stage, although it can be performed in parallel to ROIs generation if we have sufficient computing power. For this purpose we have used the popular GradCAM method(Chattopadhay *et al.*, 2018). To obtain the class-discriminative localization map, Grad-CAM computes the gradient of y^c (score for class c) with respect to feature maps A of a convolutional layer. These gradients flowing back are global-average-pooled to obtain the importance weights for pixel k with respect to class c: α^c_k (eq. 3).

$$\alpha^{c}_{k} = \frac{1}{Z}\sum_{i}\sum_{j}\frac{\partial y^{c}}{\partial A^{k}_{ij}} \qquad (3)$$

The importance weights create a saliency map as a matrix of values between 0.0 to 1.0 which corresponds to the importance of a particular pixel. Graphical representation of this map is a picture with colors from blue (lowest importance) to red (highest importance).
The final step is to compute how many classification-important pixels are inside ROI in relation to all important pixels in the image (eq. 4). In the proposed method we sum values of saliency map inside ROI and divide them by the sum of values over the entire map. If there are more ROIs detected, we calculate the ratio for each separately and in the end choose the highest value.

$$m_{fit} = \frac{\sum_{x,y}^{roi}\alpha^{c}_{k}}{\sum_{x,y}^{image}\alpha^{c}_{k}} \qquad (4)$$

This proportion gives us fit measurement. The higher value we obtain for a given class, the higher confidence that the network has learned the object, not the background.

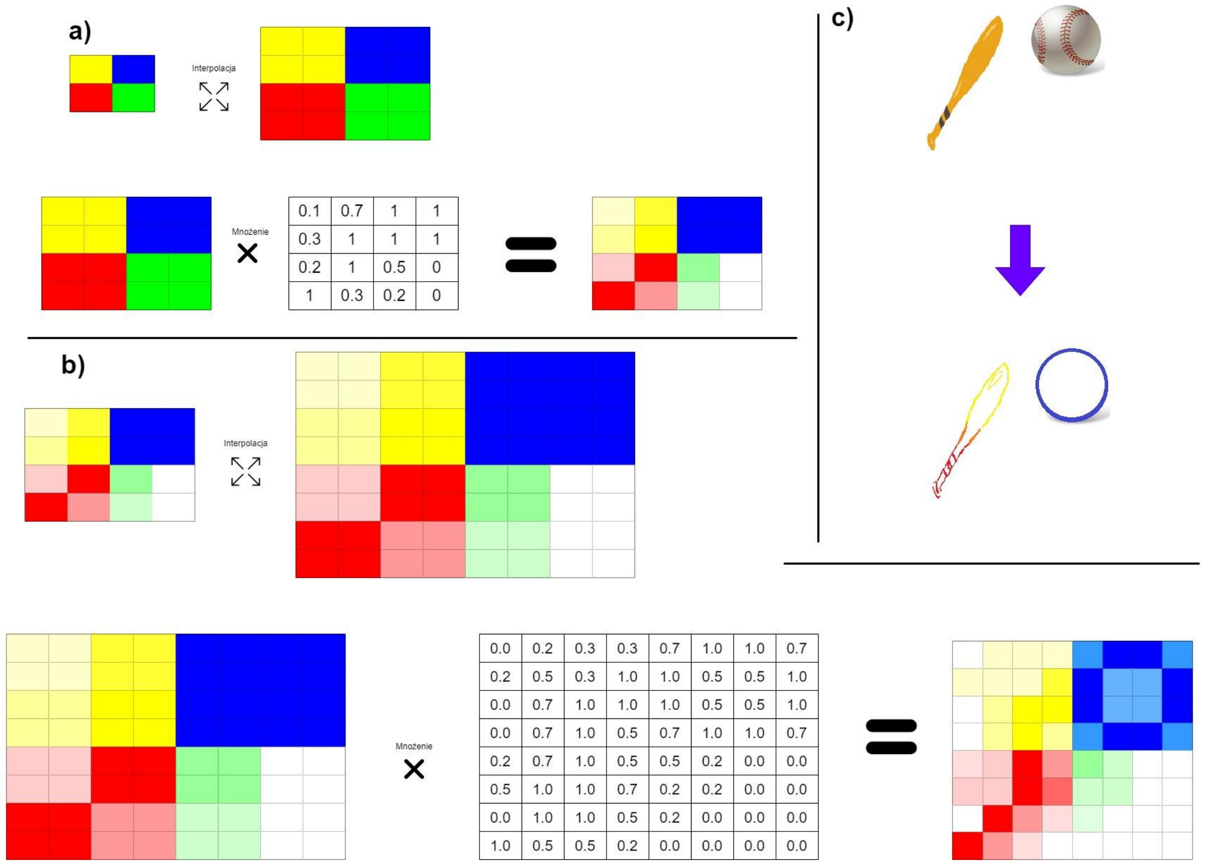
3.2 Gradual Extrapolation

Gradual extrapolation is based on a simple concept. Visualization techniques generate a saliency map at the last layer, therefore it has the size of the last layer: 7x7 (VGG-16) which is much less than the input: 224x224. Normally the map is extrapolated to the initial size thus being fuzzy due to approximation during resizing. In GE the saliency maps are expanded layer by layer and multiplied by the matrix representing the weights of the contributions from the given layer. In other words: we have the small map, 7x7, we size it to match the pre-last layer: 14x14 and then we augment the cells by original outputs from the 14x14 layer. We repeat this procedure until the original size is achieved.
Here (eq. 5), the selection of the contribution matrix is important. A simple matrix of mean values (m) is selected, and all masks of the given layer are summed with respect to the channels. For instance, assume that there are C channels, each with a mask of height h and width w. Then, one matrix of dimensions h × w is obtained as follows:

$$\forall_i \in h,\ \forall_j \in w:\ m_{ij} = \frac{\sum_{c=1}^{C} x_{cij}}{C} \qquad (5)$$

This process is repeated for each MaxPool layer until the input dimensions are restored. The concept is superficially represented in figure 23.

In other words, for each preceding layer, a matrix of importance values is determined. Therefore, simply the mean values of the single cells across all channels in the selected layer are selected. However, this unsophisticated approach can be improved and refined with more advanced techniques.



0.1	0.7	1	1
0.3	1	1	1
0.2	1	0.5	0
1	0.3	0.2	0

0.0	0.2	0.3	0.3	0.7	1.0	1.0	0.7
0.2	0.5	0.3	1.0	1.0	0.5	0.5	1.0
0.0	0.7	1.0	1.0	1.0	0.5	0.5	1.0
0.0	0.7	1.0	0.5	0.7	1.0	1.0	0.7
0.2	0.7	1.0	0.5	0.5	0.2	0.0	0.0
0.5	1.0	1.0	0.7	0.2	0.2	0.0	0.0
0.0	1.0	1.0	0.5	0.2	0.0	0.0	0.0
1.0	0.5	0.5	0.2	0.0	0.0	0.0	0.0

Fig. 23. Schematic concept of gradual extrapolation. Section a) and b) represents a gradual extrapolation transition from an image × 8. Section c) shows the concept of transition between image and its restoration using gradual extrapolation

To elucidate this process, the contribution matrix is divided by its maximal value, which restricts the possible values within the unit interval range. As the method can be applied to a model of any depth, restricting the values to 0 and 1 also prevents method overflow.

3.3 Latent Features Detection

This method is a combination of Attention Focus Evaluation and Gradual Extrapolation(GE). In short: we use GE to narrow down the saliency map to the smallest significant area, which gives us a potential part of the entire object that serves as the main discriminator of a given class.

This method also consists of three steps. The first is to determine what the network should consider as a discriminative feature according to our (expert's) best knowledge. Secondly, we have to process the image through the model and discover what the network is actually using for classification and narrow it down as much as possible . And finally, we need to evaluate how much of the actually contributed area takes the expected area of the image.

ROI identification is done as well using Detectron2.

Secondly we process the images with GradCAM thus obtaining virtually the same result as previously, but this time we enhance the saliency map with GE. Gradual Extrapolation is based on a concept that the map is expanded to the size of the preceding layer and then multiplied by the matrix representing the weights of the contributions from the given layer. Thus the obtained output is an object shaped form that can be considered the most significant part of the image, for a given class.

The final step is to compute the ratio of how many classification-important pixels are inside ROI in relation to all pixels in the ROI. In the proposed method we count the number of active pixels (above 0.1) inside ROI and divide them by the sum of values over the entire ROI (eq. 6). If there are more ROIs detected we calculate the ratio for each separately and in the end choose the lowest value.

$$m_{DFR} = \frac{\sum_{x,y}^{roi} 1 \Rightarrow p_{x,y} > 0.1}{\sum_{x,y}^{roi} 1} \qquad (6)$$

Threshold for omitting values below 0.1 has been arbitrarily chosen in order to filter out potential noise in saliency. Our empirical studies proved its effectiveness. This proportion gives us significant area measurement. The higher value we obtain for a given class, the higher confidence that the network has learned the entire object, not the latent feature.

If the ROI has not been specified on the image, the entire picture was taken into consideration.

3.4 Principal Image Sections Mapping

We proposed a method that relies on a simple concept: if we look at the final convolutional layer, the Deep Convolutional Neural Networks are only complex representation generators, therefore our focus should be on explaining: what contributed to the given representation vector the most. Here comes Principal Component Analysis, which allows us to reorganize this map into significance-sorted vectors, which could be consequently analyzed further.

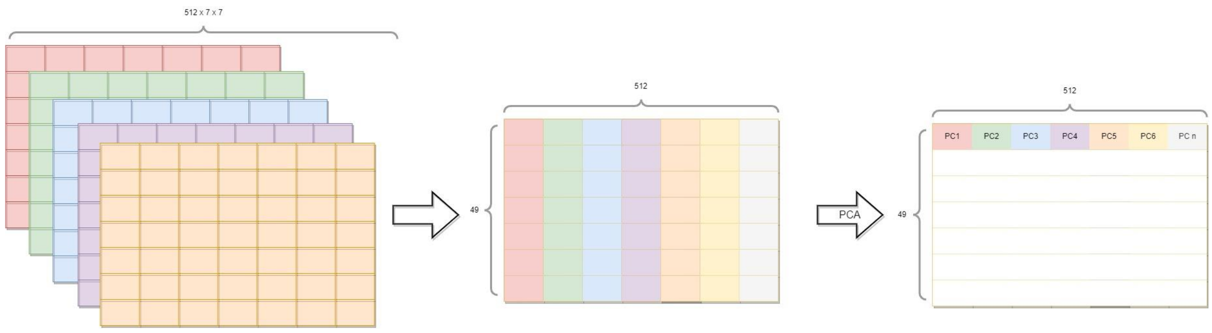


Fig. 24. Procedure of PRISM formula: first we convert maps produced in the chosen layer into vectors. In this case from 512 maps of size 7x7 into 512 vectors of length 49. On this 512x49 matrix we perform PCA

PRISM has been designed around this idea. It computes the PCA for the last convolutional layer and truncates the results after the third Principal Component thus receiving an RGB map of features as seen in picture 1.

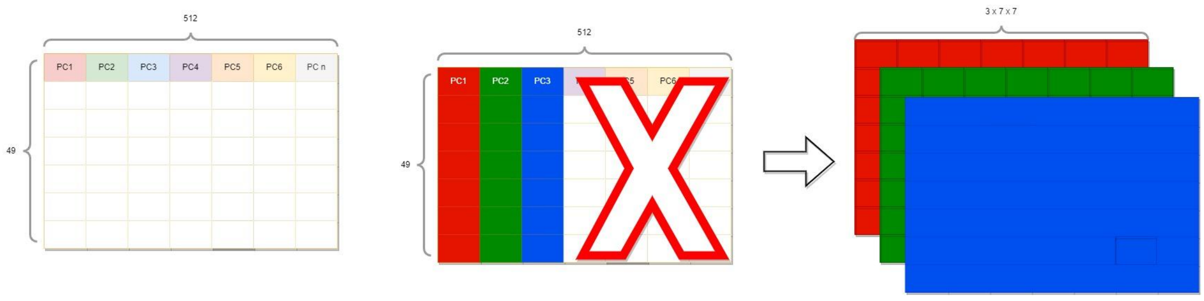


Fig. 25. While having the matrix of principal components we can remove all components beyond third one, restore each one to the original shape (1x49 into 7x7) and finally assign a base color to each one

Assume $\mathbb{A}$ is a set of outputs from each convolutional layer in a network. We choose a single output, preferably from the final layer, but there is no obstacle to study any other layer. It has a shape of $n \times c \times h \times w$:

n - number of images in a batch

c - number of channels in layer

h,w - height and width of each mask in this layer.

v - multiplication of n*h*w.

Instead of straightforward PCA for dimensionality reduction, we use Singular Value Decomposition (SVD) for obtaining Principal Components vectors. Since SVD is applicable only to two dimensional data, we have to reshape the original four dimensional batch into the two dimensional matrix (A’) and then center it by subtracting the mean. (eq 6 and 7).

$$A_{|\mathbb{A}|}^{n\times c\times h\times w} \xrightarrow{reshape} A_{|\mathbb{A}|}^{v\times c} = A' \qquad (6)$$

$$A'' = A' - mean(A') \qquad (7)$$

Now we can compute the SVD and then the PCA outcome (eq. 8). Last step is to reshape the obtained matrix back to its original four-dimensional form (eq. 9).

$$U \times S \times V^{T} = svd(A'') \qquad (8)$$

$$A_{PCA} = U \times S \qquad (9)$$

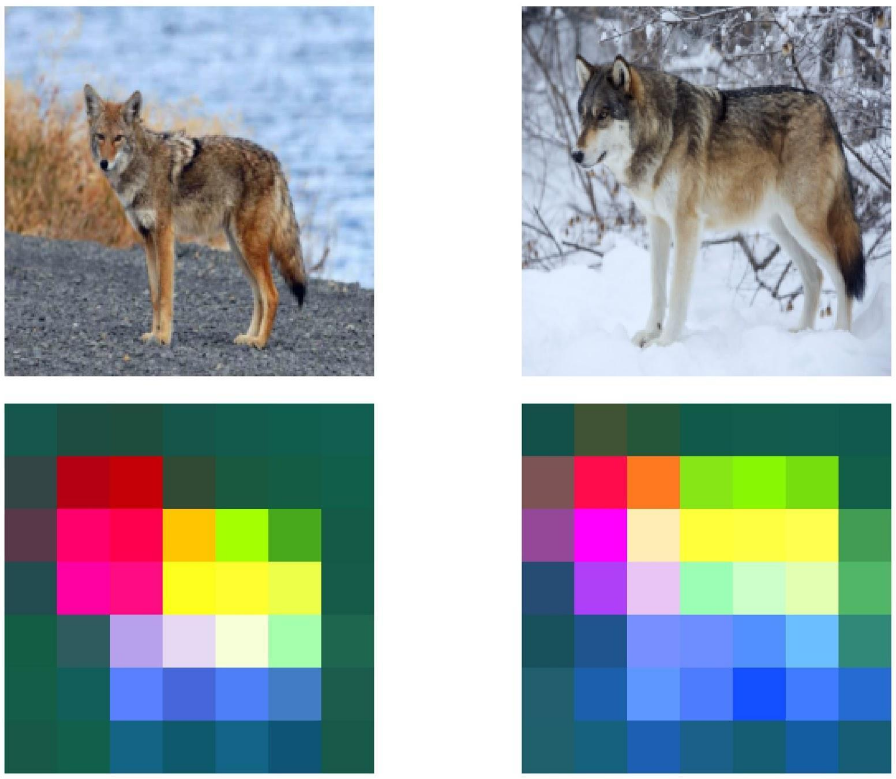


Fig. 26. Raw output from PRISM marks each cell from the processed layer with respective color according to PCA value. Similar colors on both images indicated same filters activation in the network thus we can conclude they are the same feature according to the examined model

This procedure results in a checkered representation of the processed images (fig. 3.). This Could be further processed using Gradual Extrapolation producing a human-recognizable picture with colored features found by the network as seen on figure 27.

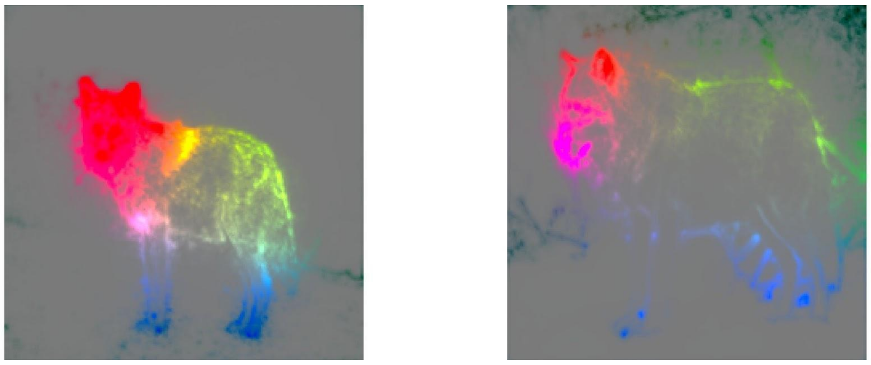


Fig. 27. Images from figure before but with applied Gradual Extrapolation to attach colors to a human-recognizable elements in the image

4 Practical Application of Proposed Methods

4.1 Semi-Automatic Model Evaluation

With the proposed procedure we have performed several experiments. Two to prove its effectiveness in detecting background skewed classes and one to generate an adversarial attack on the model using knowledge obtained from research.

4.1.1 Attention Focus Evaluation

This research has been presented during the ICCS 2021 conference and published in a paper entitled Automated Method for Evaluating Neural Network's Attention Focus(Szandała and Maciejewski, 2021).

4.1.1.1 Small-scale Evaluation for ImageNet-9

Here we have utilized transfer learning of ResNet50(Schott *et al.*, 2018) and expect it to recognize 9 subjectively chosen classes. The subset ImageNet-9 consists of classes: bird, camel, dog, fish, insect, musical_instrument, ship, snowboard, wheeled_vehicle. Each class is often associated with certain environments like camel-desert, insect-plants, ship-water, snowboard-mountains, etc. Each class has been represented by approximately 500 images. It appears that transfer learning is quite an efficient method since 8 object types were recognized correctly as the actual object. Only one class, the snowboard, took our attention. It appears that the presence of the sky implicated this recognition.
Using the CAM-in-ROI method we have noticed that only 28% of significant pixels were inside the ROI's rectangle, while for e.g. camels this value oscillated around 74%.

Table 1. ROI-fitting value and percentage of correctly classified images over ImageNet-9.. Note that images where ROI has been not found were excluded

Class	Average fitting	Model accuracy
bird	77.75 %	88.25 %
camel	74.20 %	22.70 %
dog	85.28 %	96.18 %
fish	67.87 %	62.82 %
insect	66.31 %	77.10 %
musical_instrument	59.26 %	81.80 %
ship	68.19 %	88.08 %
snowboard	28.01 %	82.28 %
wheeled_vehicle	83.77 %	94.26 %

Table 1 shows average importance, according to GradCAM, found inside ROI. The most outstanding is the snowboard class where less than one-third of saliency fits inside indicated ROI. Human conducted scrutiny revealed that for this certain class a top area associated with the sky is signal classification as the snowboard (fig. 28).
It is worth noticing that despite a low fitting ratio for the snowboard, the model’s classification accuracy does not differ significantly from other classes. This leads to a conclusion that incorrect correlation may be undetectable by standard means.

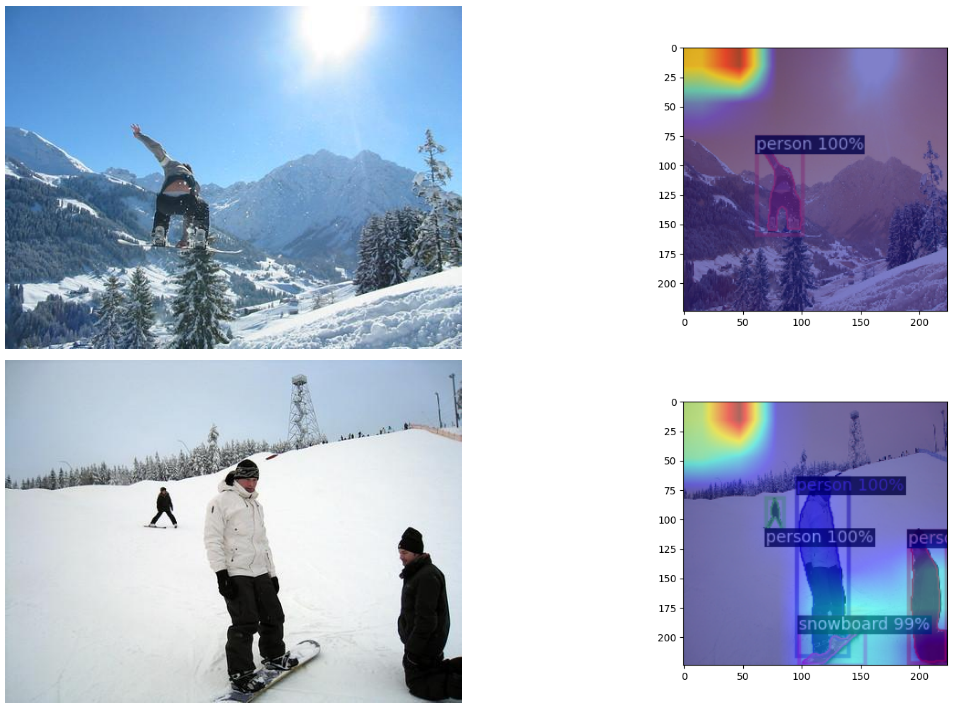


Fig. 28. Images classified as snowboard and their respective saliency maps with ROIs

This proof of concept demonstrated the practicality of the proposed method on a small subset, where human validation could be done. The method has correctly identified the snowboard classification as defective and other correlations as satisfactory.

4.1.1.2 Full-scale Evaluation on Entire ImageNet

Xiao et al. stated that state-of-the-art pretrained networks appear to be immune to background skewed learning(Xiao *et al.*, 2020). We have decided to scrutinize this statement by applying CAM-in-ROI measurement to the samples of all classes found in the ImageNet dataset and their classification using VGG architecture. We have prepared a batch of approximately 50 images of each class listed in the ImageNet set. Chosen results are displayed in table 2.

Table 2. ROI-fitting value over full ImageNet. Only selected classes were displayed

Class	Average fitting
Japanese_spaniel	96.73%
Persian_cat	93.04%
black_and_gold_garden_spider	11.08%
valley	7.75%
lakeside	7.50%
seashore	2.46%
bell_cote	1.53%
alp	1.12%
breakwater	0.68%
mosque	0.25%
rapeseed	0.00%

4.1.1.3 Method's Limitations

Some of the classes, like seashore, alp or lakeside can be considered object-less, since Detectron2 hardly ever recognized any object in the picture. Or there could have been highlighted an object that has only subsidiary importance. As we may see on the fig. 29: ROI targets a person, while the network focuses, correctly, on the surroundings and classifies the image as a valley. Furthermore, some objects like bell cote are not known to the Detectron2. These issues caused many false positives in our results, although these may be overcome by applying different, more compelling ROI detecting tools.

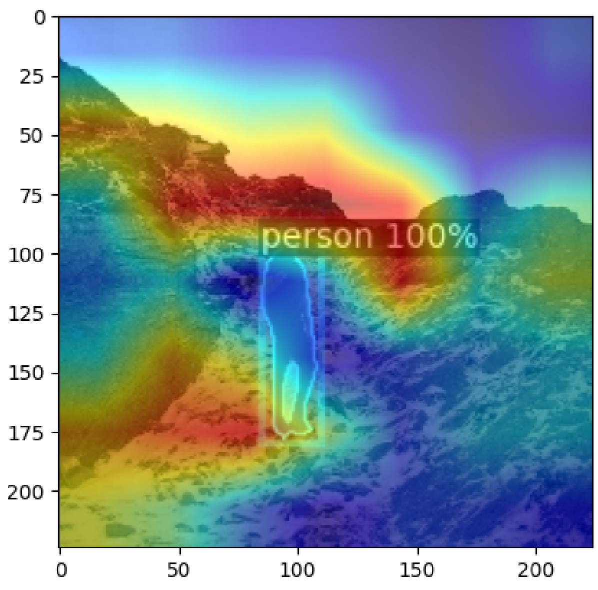


Fig. 29. An image classified correctly as a valley, despite the person detected in the foreground. Note that a tiny saliency region is included inside ROI thus giving as non-zero fitting measure

Further we must acknowledge one more setback: our technique will turn out ineffective for detecting incorrect perception of the object. Namely, like the aforementioned association of the cab object with yellow(Geirhos *et al.*, 2018).

4.1.1.4 Generating Adversarial Attack Based on Method Findings

Nonetheless, our method has greatly limited the number of classes that require a closer look. Therefore we had to manually inspect only the lowest fitting classes. By reviewing images from the lowest fits we can highlight one distinctive class: black_and_gold_garden_spider. The black-and-gold-garden-spider often appears on the spider web background. While ROI indicates the insect quite well, the saliency map marks the spider web as the most significant area on the image, therefore resulting in only 11% average fitting. This leads us to the conclusion that the network assigns the label black_and_gold_garden_spider to an image that displays an object on a spider's web background.

Enhanced with knowledge about background influence on a certain class we can forge an image that has a spider's web in the background and something else in the foreground. We have downloaded a web image as well as a deer's head and merged them into one picture. Before the fusion, each picture was separately correctly classified: web as web and deer as a hartebeest (fig. 30). The blended image, as expected, has been classified as black-and-gold-garden-spider.

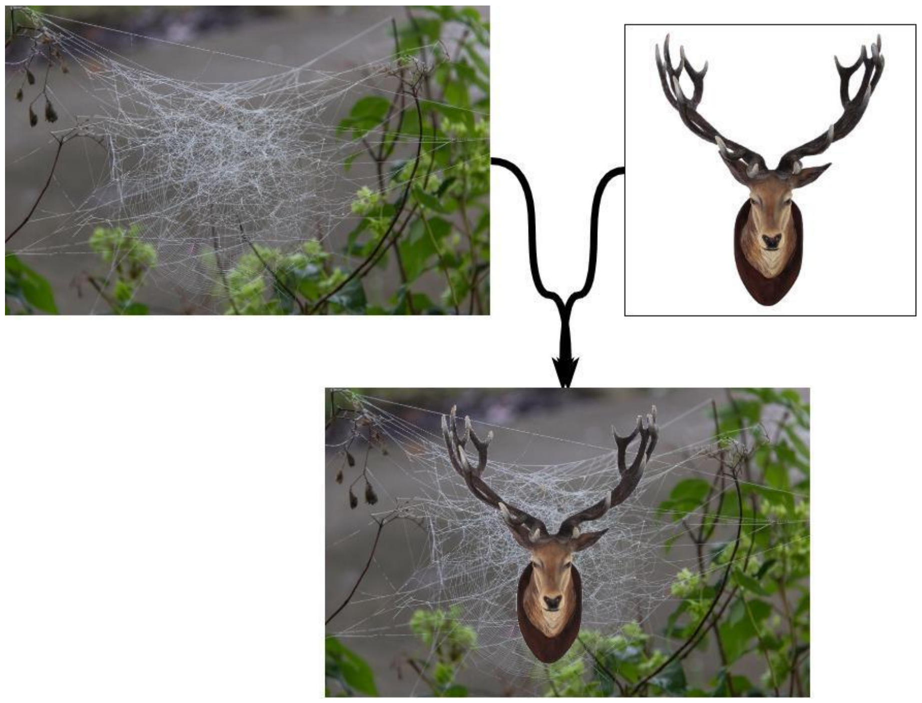


Fig. 30. Source images of spider's web and a deer's head that, according to results from 3rd experiment, should result in classification as black_and_gold_garden_spider

By knowing the flaws in the network's reasoning, we were able to conduct an adversarial attack on it. Its success ultimately proves the potency of the proposed method.

4.1.1.5 Summary

Name	Automatic Network’s Attention Focus Evaluation
Method’s application	
Apart from testing accuracy, when evaluating a model, we should also test whether the model is learning the actual object, not the context of the object.	
Procedure	
1. Train the model as usual 2. Perform ROI indication on each image from validating set 3. Obtain saliency map for each image 4. Measure ration of saliency inside ROI to saliency over entire image 5. Calculate average for each class 6. Manually inspect classes with lowest average ratio value 7. If any class appears to clearly learn background instead object - adjust training	
Benefits	**Limitations**
● Complementary evaluation for accuracy/error rate ● Identifies classes in case which model is learning context instead object	● ROIs generator might not know all objects known to the evaluated model ● Requires a lot of additional computation to generate ROIs and saliency maps for each image

4.1.2 Detecting Latent Features

Unlike the previous approach this time we assume that the network is classifying correctly and the discriminative element is the actual object. However in this case we would like to evaluate whether the object as a whole has contributed or only a part of it.

4.1.2.1 Small-scale Evaluation on ImageNet-9

For this research we used transfer learning with the VGG-16 model which was targeted to solely 9 classes. The subset ImageNet-9 consists of classes: bird, camel, dog, fish, insect, musical_instrument, ship, snowboard, wheeled_vehicle. Each chosen class is often associated with their respective environments like camel-desert, insect-plants, ship-water, snowboard-mountains, etc. Each class has been represented in a training dataset with 500 images. Transfer learning is an efficient method since almost all object types were recognized correctly as the actual object.

Table 3. ROI active area value and percentage of correctly classified images over ImageNet-9

Class	Average fitting	Model accuracy
bird	37.8 %	91.00 %
camel	19.9 %	72.72 %
dog	30.5 %	90.44 %
fish	50.3 %	92.71 %
insect	31.8 %	82.11 %
musical_instrument	27.7 %	84.77 %
ship	51.5 %	89.48 %
snowboard	57.1 %	79.27 %
wheeled_vehicle	**13.4 %**	**91.19 %**

Table 3, second column shows average activity, according to GE GradCAM, found inside ROI. The most outstanding is the wheeled_vehicle where less than one-seventh of ROI is considered significant for the classification. Taking individual look into wheeled_vehicle dataset we notice that indeed the model focuses on the wheels, less on the vehicle as a whole.

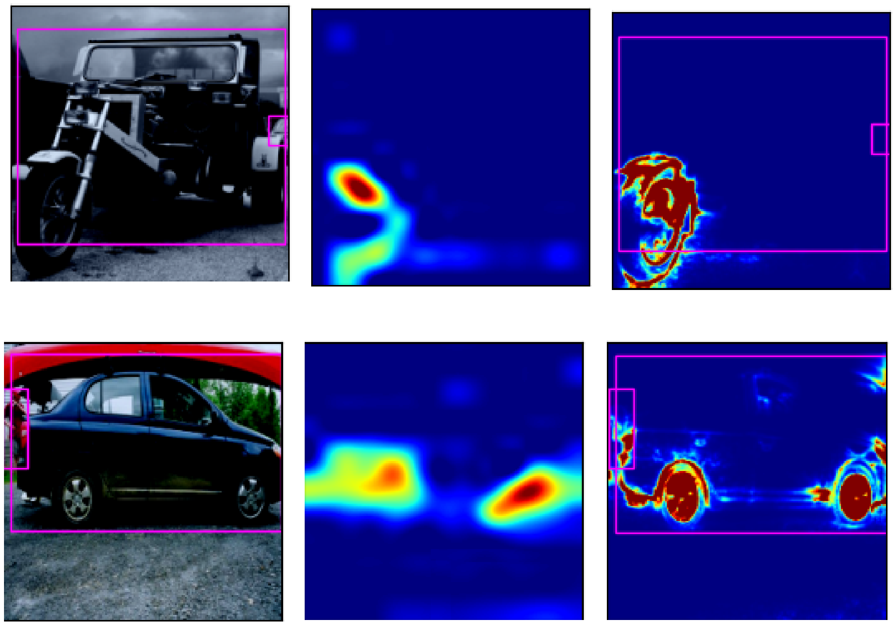


Fig. 31. Analysis of transfer learned model on wheeled_vehicle class. It show classifier's focus on the wheels

4.1.2.2 Adversarial Attack Using Findings from ImageNet-9 Experiment

Enhanced with this knowledge we could find an image depicting a wheel-like object as seen in figure 32 and expect it to be classified as a wheeled vehicle. The choice was a roulette image which has been tagged as expected with 88% certainty.

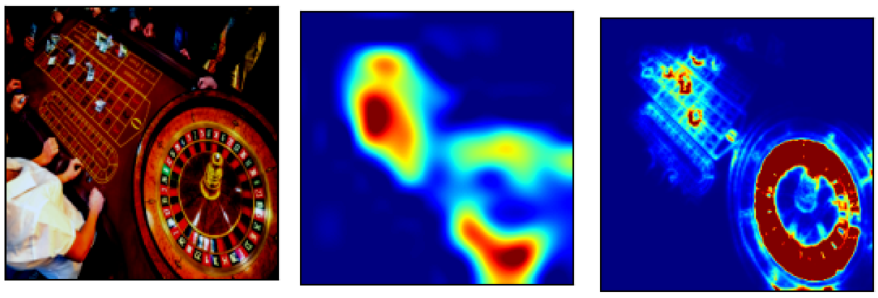


Fig. 32. Using network to classify this image resulted in labeling it as wheeled_vehicle due to roulette's wheel presence

4.1.2.3 Full-scale Evaluation on the Entire ImageNet

After confirming that method works on a subset we could proceed to the state-of-the-art model, VGG-16 with all 1000 ImageNet classes. Each was represented by approximately 50 images. The application of the method has been presented in a table below.

Table 4. Sorted by ROI activity results from ImageNet, using VGG-16. Note that only arbitrary chosen classes were displayed for readability

Class	Mean active area
lakeside	61.0 %
diamondback	59.5 %
seashore	56.5 %
tick	46.1 %
…	
ringneck_snake	15.4 %
Samoyed	15.1 %
nect_brace	14.6 %
shoji	13.1 %

With a closer look we have generalized conclusions and divided them into 4 cases.

4.1.2.4 Method’s Limitations: Landscape Areas

The first result that caught our attention is the lakeside class. Very high influential value originates from method’s rule: if there are no ROIs in the picture, we consider the image as a one, big ROI. This decision resulted in very high mean value as landscapes provide an evenly spread saliency map, even with Gradual Extrapolation application.

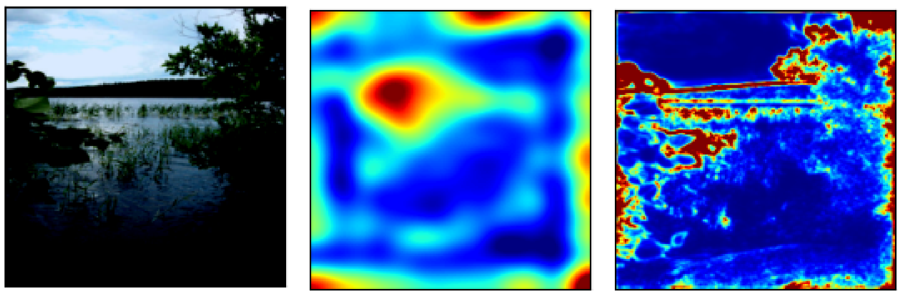


Fig. 33. Network saliency distribution for a lakeside image

4.1.2.5 Method’s Limitation: Hardly Descritable Objects

Lowest score has been achieved by shoji, the Japanese paper/wooden doors. Surprisingly the network correctly aligns attention with the doors that the problem is in the ROI detector. This type of door is hard to define therefore ROI-marking is unable to indicate them. Moreover, they are often depicted inside a house, which is full of other, known to ROI-generating tool, objects. These issues result in false positive score as unimportant items are marked with ROI, while the network correctly focus on a doors as seen in a picture below.

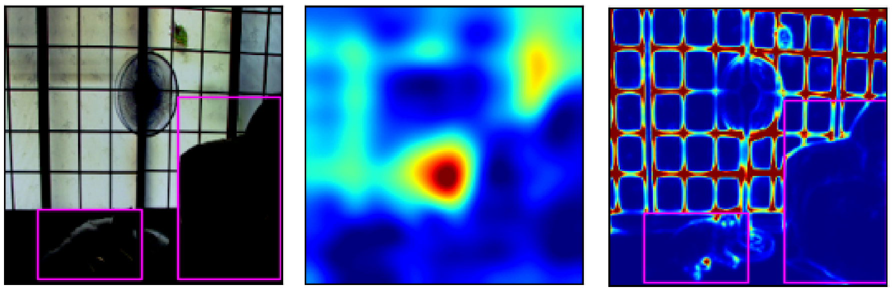


Fig. 34. Sample output from applying proposed method to shoji, where we can notice that ROI focused on a foregrounds while the actual object was in the background

4.1.2.6 Generating Adversarial Attack Based on Method's Findings

Most of the animals were located high when sorted by mean activity area, surprisingly only the Samoyed distinguished from this pattern. After inspection of results for this class we deduced that Samoyed has 3 black stains on a white area: one for nose, two for eyes.

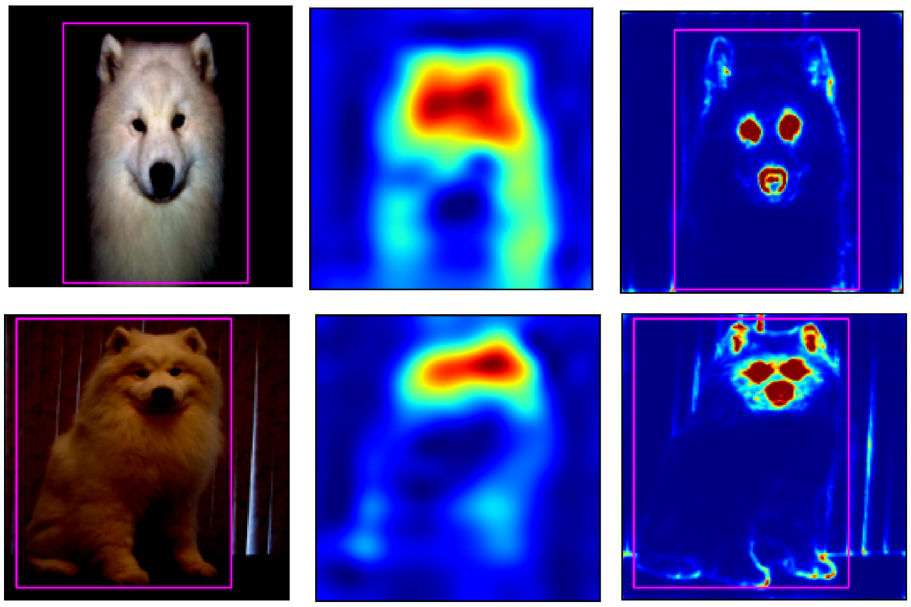


Fig. 35. Sample output from applying proposed method to two Samoyed class representatives. We can see the saliency is skewed into eyes and nose

This revelation suggests that we could assemble a Samoyed specimen by painting 3 black dots on a white background. Unfortunately the manual approach was insufficient -

perhaps one can be generated using GAN , therefore we decided to threshold the Samoyed picture and trim it as much as possible.

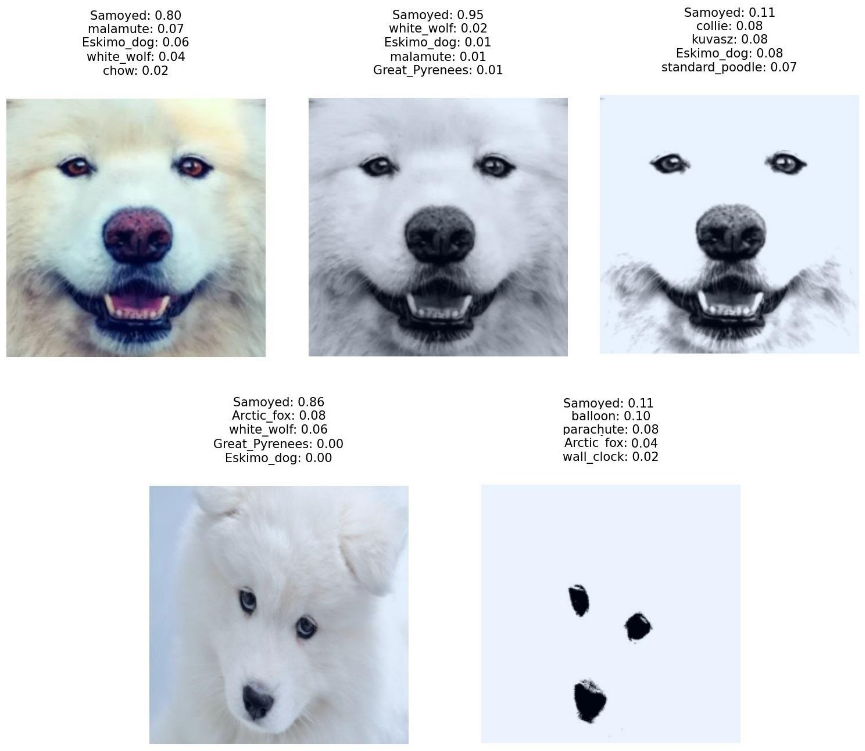


Fig. 36. Successful pruning of colors and features (like ears or fur) to keep Samoyed classification

The confidence drops significantly, however even the 3 dots are still classified as a Samoyed. Another interesting observation is the middle picture in the top row. It depicts the same Samoyed's muzzle as the left one, but after conversion to black and white color palette its confidence score is higher than the original image.

4.1.2.7 Summary

Name	Automatic Detection of Latent Features
Method's application	
Apart from testing accuracy, when evaluating a model, we should also test whether the model is learning the entire object instead of only part of it.	
Procedure	
8. Train the model as usual 9. Perform ROI indication on each image from validating set 10. Obtain saliency map enhanced with GE for each image 11. Measure ratio of saliency inside ROI to ROI's size for each image 12. Calculate average for each class 13. Manually inspect classes with lowest average ratio value 14. If any class appears to learn only part of the object - adjust training	
Benefits	**Limitations**
● Complementary evaluation for accuracy/error rate ● Identifies classes in case which model is learning only part of the object, instead entire object	● ROIs generator might not know all objects known to the evaluated model, e.g. landscapes ● ROIs are usually rectangular. Some objects have irregular shapes (e.g. snake) which results in a false positive ● Requires a lot of additional computation to generate ROIs and saliency maps for each image

4.2 Closer Insight into Models' Decision Process

Thus far we have focused on an entire dataset analysis, but for two reasons we should have tools for more detailed inspection of the certain examples. First, to generate conclusions applied to the entire class. Secondly, the trend is to have much smaller, but significantly better suited datasets to train networks in order to reduce training computation costs. Two methods: Gradual Extrapolation and Principal Image Sections Mapping contribute to this pattern as they allow for deeper insight into particular image classification.

4.2.1 CNN Visualization Performance Evaluation - FIAt

From one of the previous experiments we see that GE can successfully be applied to a practical problem, but here I would like to prove its value by introducing a set of tests that can be used to evaluate virtually any visualization method. The set of tests can be abbreviated into FIAt: faithfulness, interpretability and applicability tests. These tests were described in the paper Enhancing Deep Neural Network Saliency Visualizations with Gradual Extrapolation published in IEEE Access.

4.2.1.1 Faithfulness

First the faithfulness, we would like to assess whether our method is trustworthy. In other words: we check if the outcome of the method gives results as expected. For measuring visualization methods we could utilize pixel-flipping(Samek *et al.*, 2017). The pixel-flipping procedure assesses whether removing the features highlighted by the explanation, as the most relevant, decreases the network prediction abilities.
Pixel flipping goes from the most to the least salient input features, removing them one by one and monitoring the change of the neural network output. The series of recorded decaying prediction scores can be plotted, where the faster the curve decreases, the more faithful the explanation method with respect to the neural network's decision. The pixel-flipping curve can be computed for a single example or averaged over an entire data set to achieve a global estimate of the faithfulness of an explanation procedure under study.
For this assessment, random 1000 images are selected from the ImageNet dataset and classified using VGG-16, yielding a 1.00 confidence score (based on the Softmax formula over the network). In Figure 6, Gradual Extrapolation offers noticeable improvements over conventional Grad-CAM and a minor improvement over similar Excitation Backpropagation.

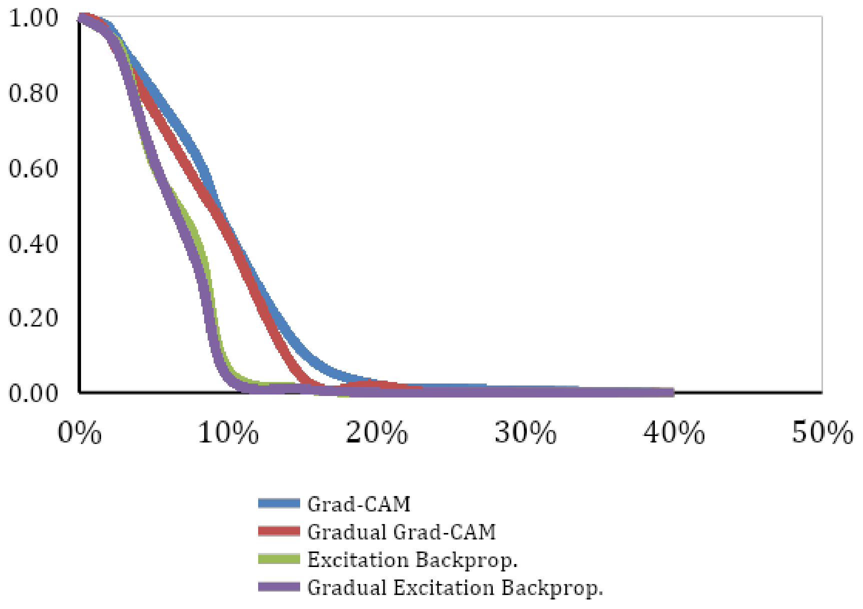
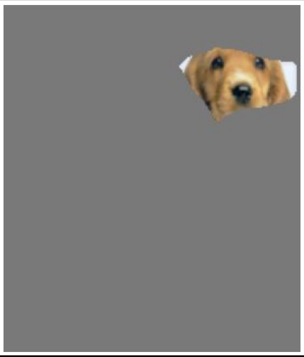
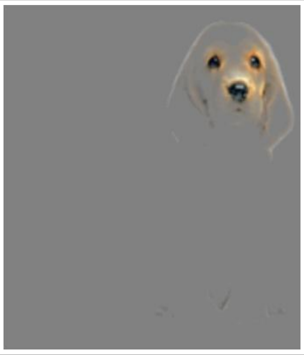
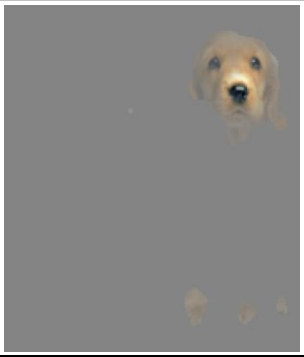
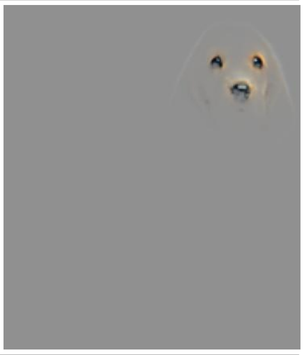


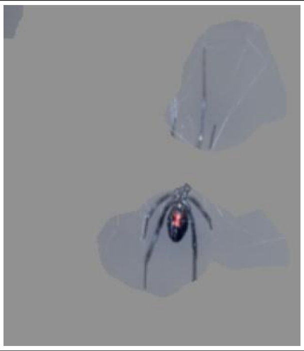
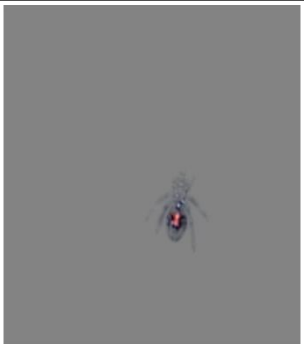
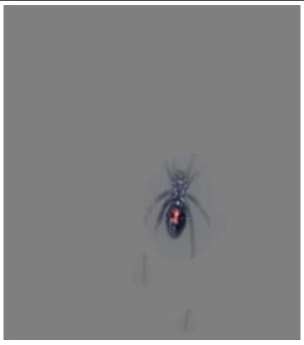
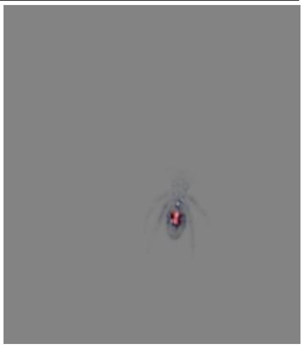
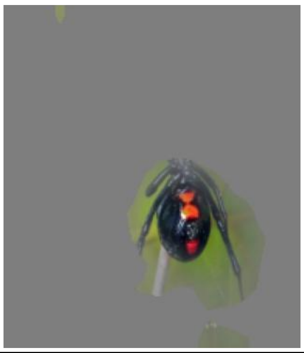
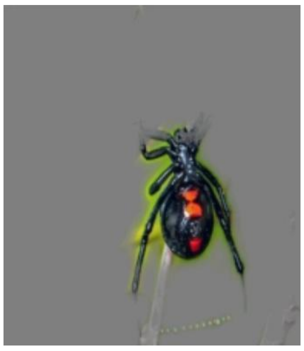
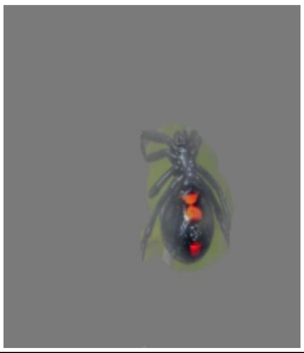
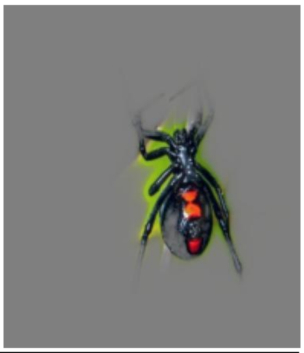
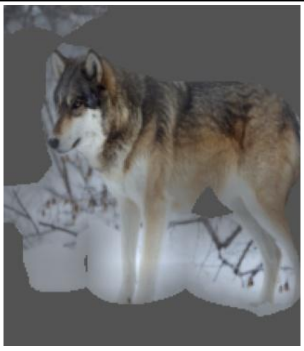
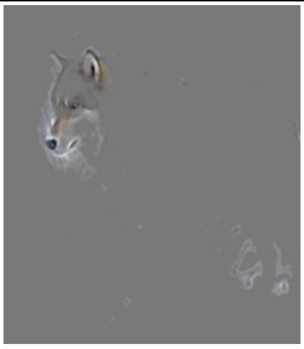
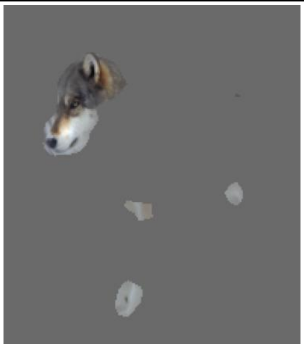
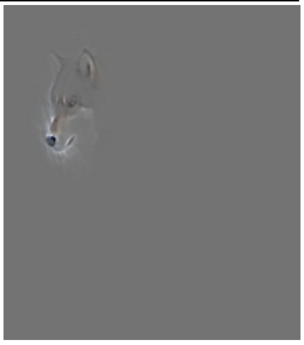
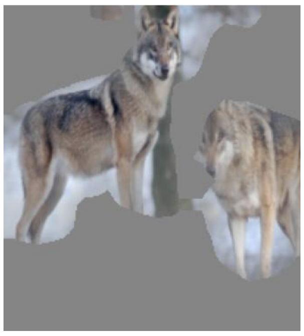
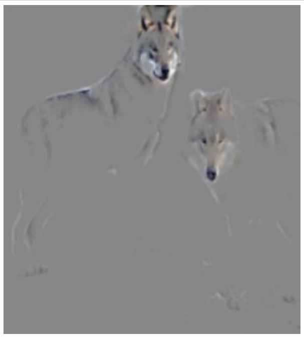
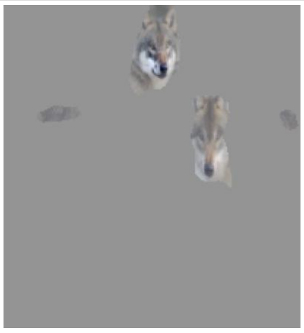
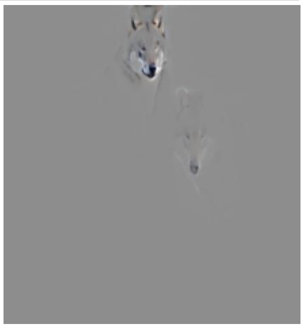
Fig. 37. Results of pixel-flipping impact on model's classification accuracy

4.2.1.2 Interpretability

Next we should evaluate interpretability. In the case of visualization technique interpretability can be defined as how much of original data reduction can be achieved using a given method, but still retaining the significant information. Based on this assumption we designed a test that counts how many pictures are left as very important compared to the original image.
Again we compare GE-enhanced methods with their vanilla counterparts.

Table 5. Significant regions and their sizes (in pixels) based on vanilla Grad-CAM(1st col.), Grad-CAM using Gradual Extrapolated (2nd col.), Contrastive Excitation Backpropagation(3rd col.) and Gradual Extrapolated version(last col.)

			
Significant pixels: 4713	Significant pixels: 3178	Significant pixels: 5557	Significant pixels: 3259

			
Significant pixels: 16576	Significant pixels: 1406	Significant pixels: 8162	Significant pixels: 1341
			
Significant pixels: 9528	Significant pixels: 10039	Significant pixels: 4749	Significant pixels: 8835
			
Significant pixels: 29527	Significant pixels: 4452	Significant pixels: 9994	Significant pixels: 3983
			
Significant pixels: 22673	Significant pixels: 5178	Significant pixels: 11855	Significant pixels: 7883

Generally, we see a firm decline in the size of salient areas between the conventional methods and their enhanced versions. The only exception is one of the black widows images - in the third row. The number of important pixels is higher in the enhanced versions. However, the enhanced version is more focused on an animal unlike vanilla one, which includes a lot of background.

4.2.1.3 Applicability

Last evaluation, applicability, is about utility of the method. Whether it is applicable to more cases than only one, specific, that the method originates from. From the state of the art we know that both GradCAM and Contrastive Excitation Backpropagation are applicable for most or all convolutional neural networks. Question arises whether their GE version applies as well, since it relies on the models architecture.

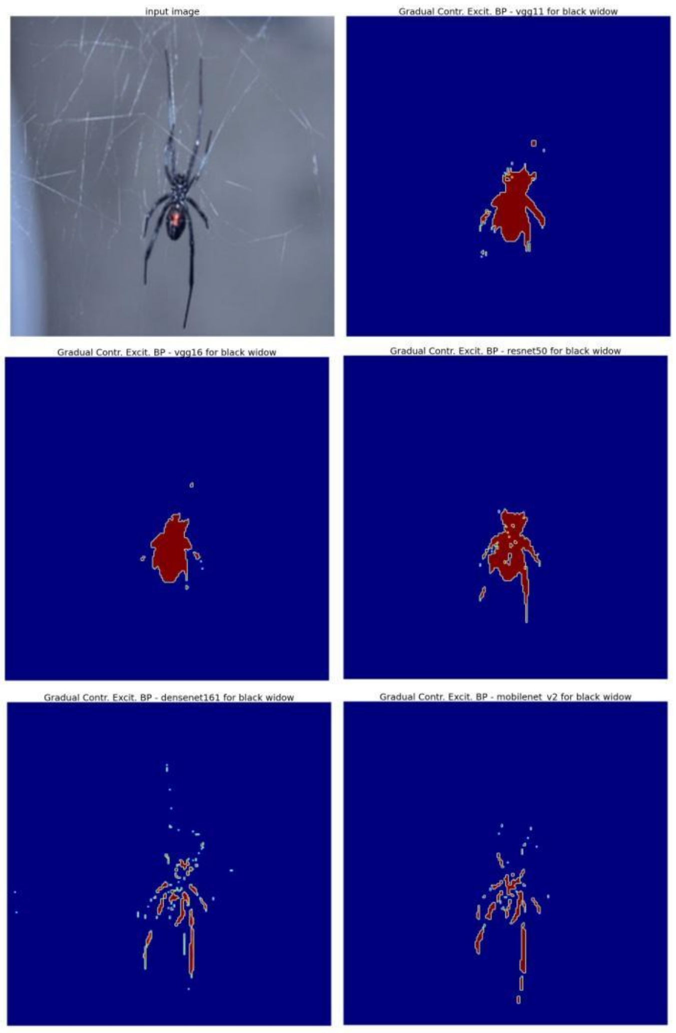
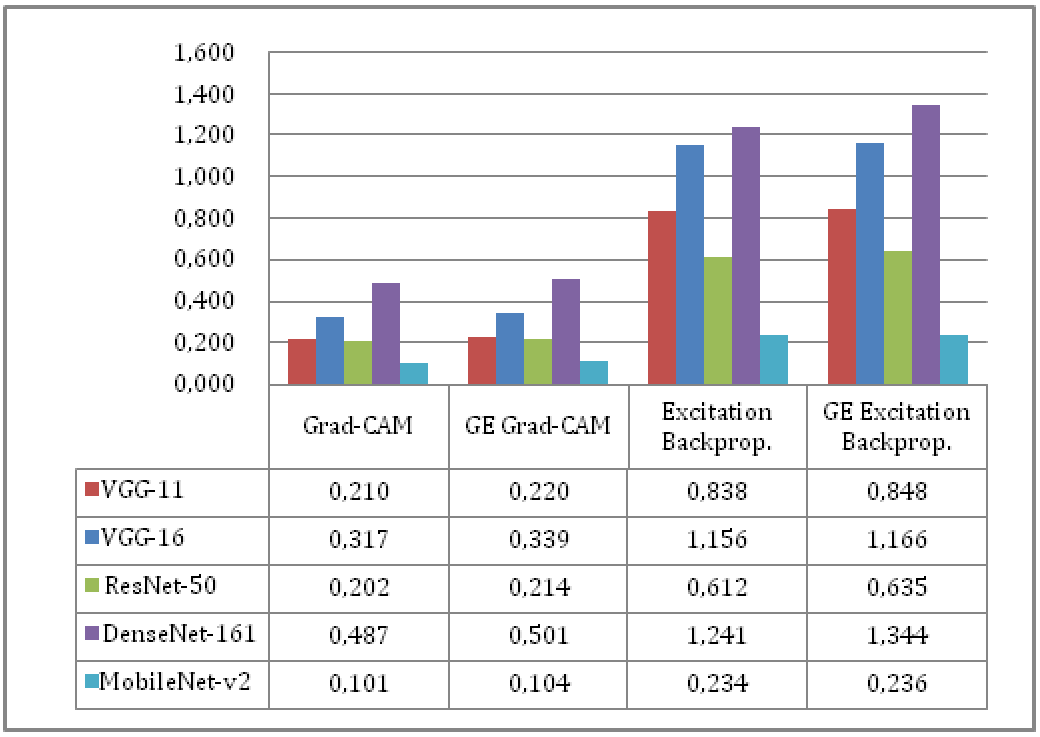


Fig. 38. Gradual contrastive excitation Backpropagation results of several models

This final test shows that Gradual Extrapolation is applicable to a wide variety of models. The following sample networks are used:

- VGG-11 (Simonyan and Zisserman, 2014)
- VGG-16 (Alippi, Disabato and Roveri, 2018)

- ResNet50 (He *et al.*, 2016)
- DenseNet-161 (Huang *et al.*, 2017)
- MobileNet-V2 (Sandler *et al.*, 2018)



	Grad-CAM	GE Grad-CAM	Excitation Backprop.	GE Excitation Backprop.
🟥 VGG-11	0,210	0,220	0,838	0,848
🟦 VGG-16	0,317	0,339	1,156	1,166
🟩 ResNet-50	0,202	0,214	0,612	0,635
🟪 DenseNet-161	0,487	0,501	1,241	1,344
🟦 MobileNet-v2	0,101	0,104	0,234	0,236

Fig. 39. Gradual Extrapolation time obtaining time in second for several models. Average for 30 executions

To measure applicability, the proposed method is applied to several different models and their computational time is measured. The results related to the computational time are shown in figure 39. The difference between the conventional method and the enhanced version is negligible, which is less than 10%. Note that because Gradual Extrapolation depends on an output from the original method its computation period would never be shorter than the original procedure.

4.2.1.4 Summary FIA-t

Name	Faithfulness, Interpretability and Applicability test
Method's application	
Comparison of methods that are generating saliency map for the classified images	
Procedure	
1. Choose methods to comparison - preferably include at least one state-of-the-art technique 2. Identify the most salient pixels according to each method. Start gradually removing them from the image and perform classification 3. Note network's confidence after each series of truncation 4. The more significant drop in accuracy, the truly more important the removed pixels were 5. Agree on threshold of saliency (between 0 and 1) and remove all pictures from entry image that are below this number 6. Calculate amount of pixels left. The less are left the more precise the method is 7. Apply the method to several state-of-the-art models. If the result is useable and obtainable in a finite time the method is considered applicable across multiple models	
Benefits	**Limitations**
● Quantitative comparison of saliency map generating techniques ● Consists of strict rules of comparison	● Focuses only on comparison of saliency map generating techniques ● Does not answer the question whether the method is meaningful for the XAI subject

4.2.1.5 Summary Gradual Extrapolation

Name	Gradual Extrapolation
Method’s application	
Method to sharpen the output generated by saliency map generating techniques	
Procedure	
8. Train the model as usual 9. Set “traps” at the output of each MaxPool layer to collect activations for given layer 10. Obtain saliency map for last layer 11. Expand saliency map to the size of preceding layer 12. Multiply expanded map by the averages output from preceding layer 13. Scale result to [0,1] to avoid extremely low or high values 14. Repeat steps 4-6 until reaching initial layer/size	
Benefits	**Limitations**
● Highlights shapes of the salient object ● Significantly reduces the amount of area marked as salient for given image ● May sometimes improve localisation of salient object ● Applicable to all methods that are generating saliency map or similar result	● Is as good as the base method. If base method points wrong are GE will not fix that ● Output might be blurred when applied to models that have e.g. recurrent or parallel architecture

4.2.2 Feature Scoped Classification Analysis - PRISM method

This subchapter originates from the paper PRISM: Principal Image Sections Mapping published as part proceedings of ICCS 2022 (Szandala and Maciejewski, 2022). In order to prove usability of this method we have performed and analyzed 3 experiments

4.2.2.1 Using PRISM Output to Identify Discriminative features

The first experiment was meant to use PRISM to find and make human-identifiable features that distinguish 2 potentially similar classes. We have started from coyotes and timber-wolves. We have compared these two classes and prepared an example that proves the differentiating features can be identified using PRISM.

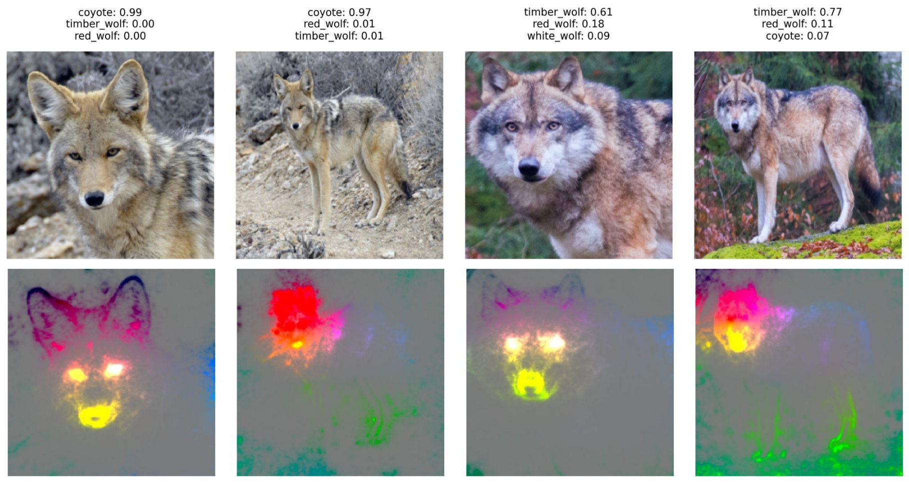


Fig. 40. PRISM output for coyote and its zoomed face as well wolf and its zoomed head

In figure 40 we see PRISM results for 4 pictures. First we started with full bodies of the animals (columns 2nd and 4th). Pawns, tail and back appear to be the similar feature vectors indicated by the same colors, but the difference appears to be in the animal's heads. Therefore we scoped at them by clipping the original images. Again features identified by PRISM seem mostly equal, but the differences now can be found in the eyes and ears areas.

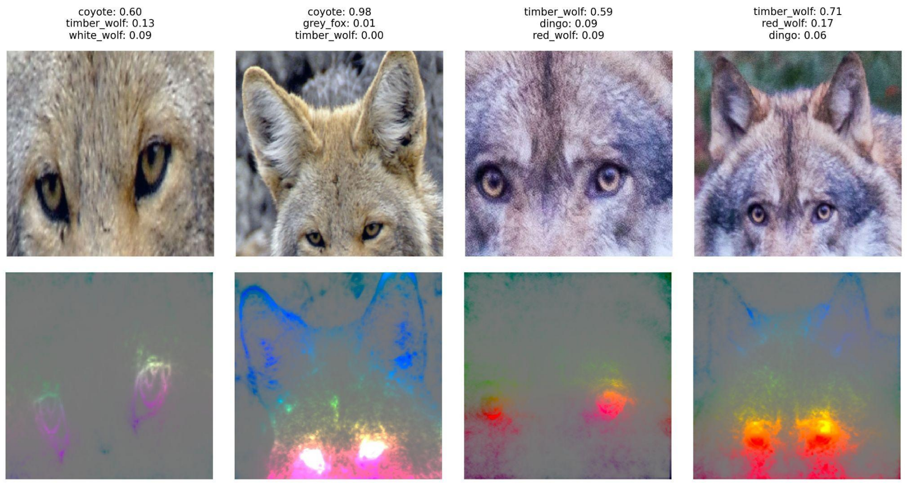


Fig. 41. PRISM output for coyote's and wolf's top parts of the heads and finally eyes

Figure 41. depicts further clipping. First we clip to the top parts of the heads - eyes and ears. This displays similar colors in regards to ears, but different in eyes area, so we perform the final clipping. This leads us to the conclusion that the feature which contributes the most to the distinction between the timber wolf and the coyote lies in their eyes.

In order to confirm our assumption we have merged coyote eyes into the wolf and other way around. As we see on figure 42. the classification changed significantly. Although the most probable class is still a respective animal, the second guess is the one that we tried to induce. Also note that confidence dropped significantly in favor of the second class. Perhaps better results could be achieved with more sophisticated blending instead of simple copy-paste.

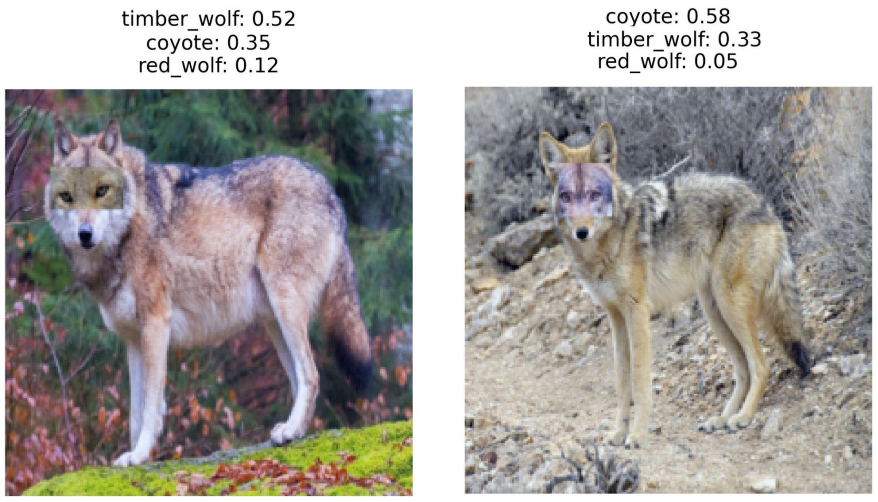


Fig. 42. Classification confidence after swapping eyes between two animals

Of course a question comes, whether basic GradCAM could lead us with a similar deduction path? We have processed the original images and generated GradCAM output for them. As seen in figure 43. the outcome is not easily interpretable, therefore it would be significantly harder to generate a similar so focused adversarial attack.

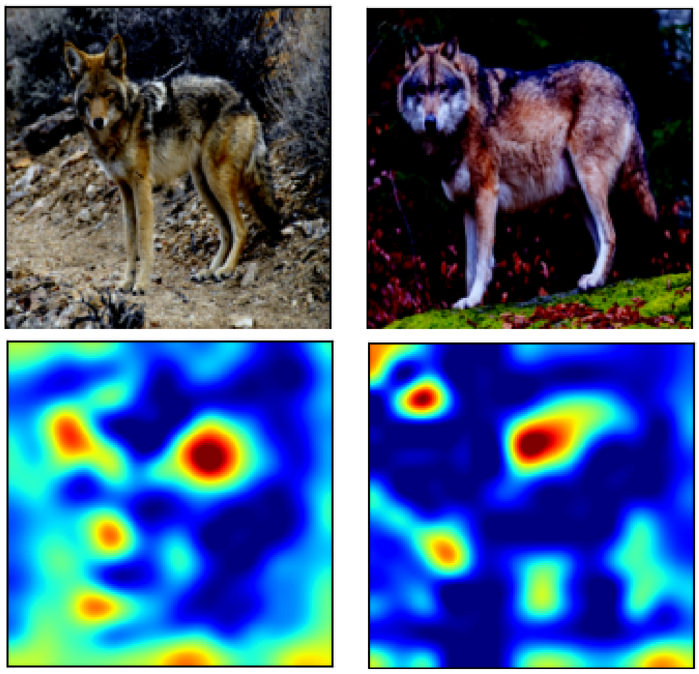


Fig. 43. GradCAM output for original image of coyote and wolf

4.2.2.2 Classes Clustering with PRISM's Output

Despite that PRISM shines while processing several images at once, mostly in conjunction with GE, we believe it may also work as an indicator for ambivalent classes if combined with clustering technique, e.g. Self-Organizing Maps (SOM). In picture 44. we have presented a first draft of PRISM's clustering utility. We have taken 5 canine classes (color from cluster map in bracket):

- coyote (orange)
- gray fox (red)
- timber wolf (green)
- samoyed (purple)
- border collie (blue)

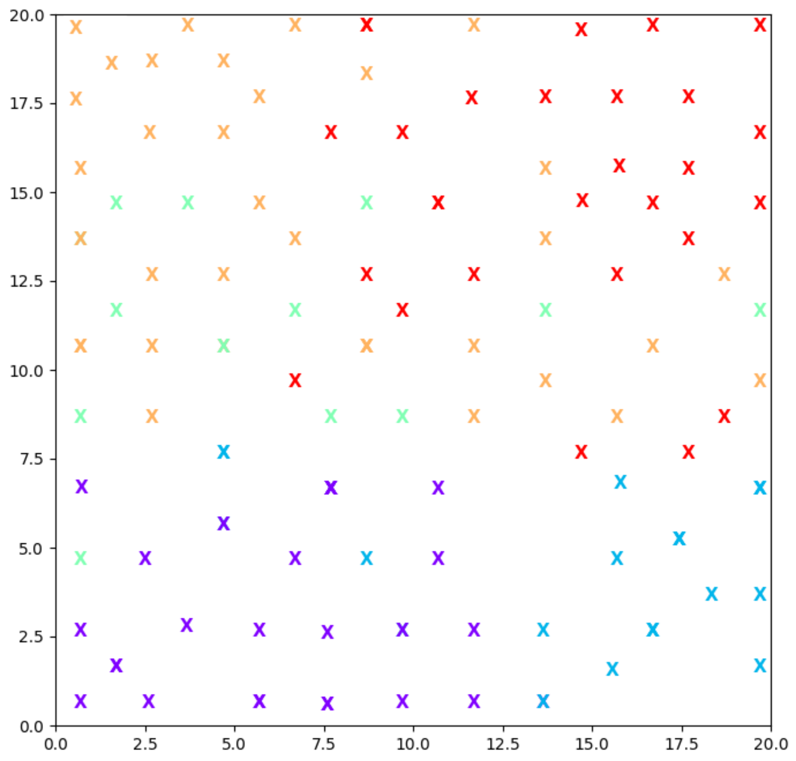


Fig. 44. SOM clustering outcome for 5 arbitrary chosen classes whose representation were generated by the PRISM

Instead of drawing their PRISM-generated representation we have used them as feature vectors for SOM clustering. Note that PRISM is generating real domain values for coloring, therefore we had to quantize the colors to reduce the amount to a finite number of possible tints.

From figure 44 we can conclude that coyotes(orange) could be easily confused with timber wolves(green) and gray foxes(red). On the other hand the Samoyed and Border collie specimens (purple and blue respectively) are clearly distinguishable from the rest.

Table 6. Confusion matrix for images clustered using PRISM.

	Predicted class				
	coyote	gray fox	timber wolf	samoyed	border collie
coyote	25	2	3		
gray fox	1	22	3		
timber wolf	1	1	10		
samoyed				19	1
border collie			1	1	11

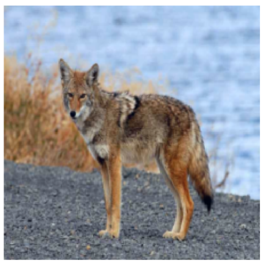
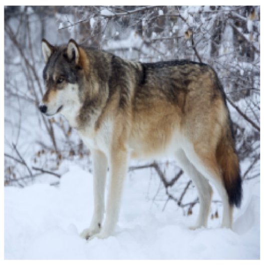
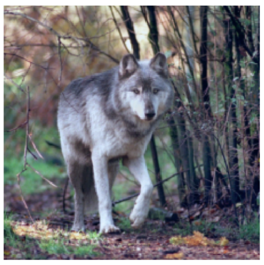
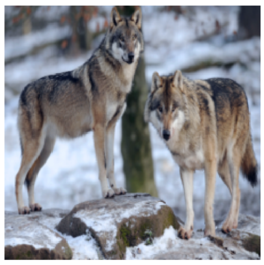
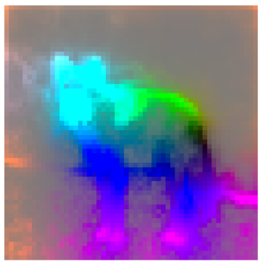
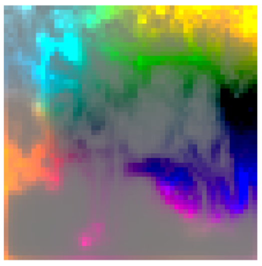
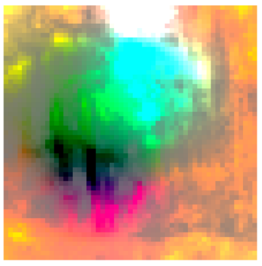
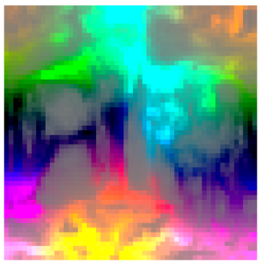
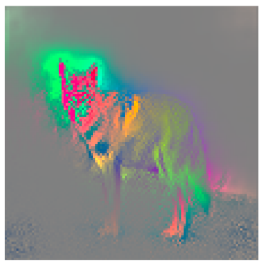
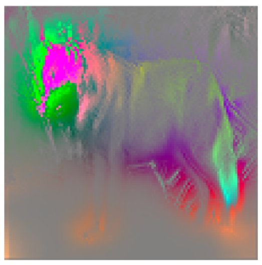
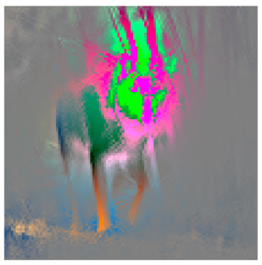
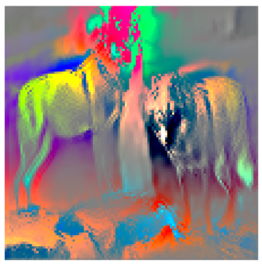
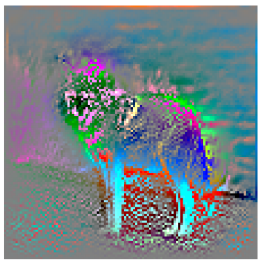
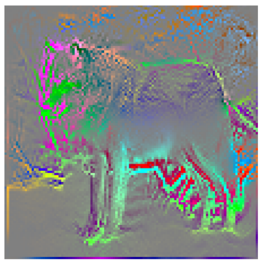
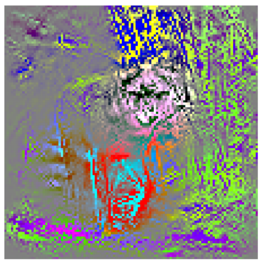
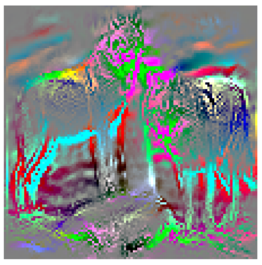
Having the clustered output we have also prepared the confusion matrix. In the matrix we can clearly see that 2 superclusters have been reproduced: one for coyote, gray fox and timber wolf; second for samoyeds and border collies. It proves that PRISM has successfully predicted possible confusions between classes. Timber wolves can be easily confused with coyotes and foxes and Samoyeds and Border collies are clearly separated from wild specimens.

4.2.2.3 Applicability to Different Models

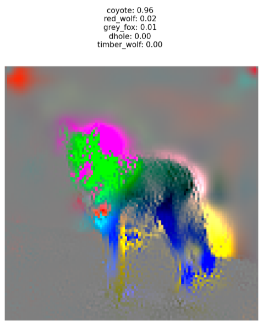
PRISM, by its design, is applicable to any model that generates representations. However it shines when combined with GE, therefore we have validated this combination for 9 models found in the Model ZOO. We have tested GE PRISM with:

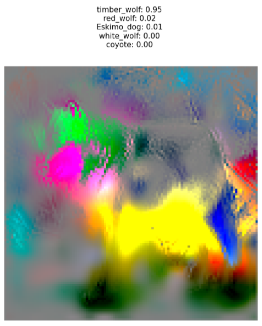
- AlexNet (Simonyan and Zisserman, 2014)
- GoogleNet (Szegedy *et al.*, 2015)
- ResNet-18 (He *et al.*, 2016)
- ResNet-50 (He *et al.*, 2016)
- ResNet-101 (He *et al.*, 2016)
- SqueezeNet (Iandola *et al.*, 2016)
- VGG-11 (Simonyan and Zisserman, 2014)
- VGG-16 (Alippi, Disabato and Roveri, 2018)
- VGG-19 (Simonyan and Zisserman, 2014)

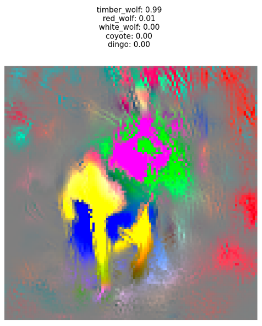
Table 7. PRISM output for multiple state-of-the-art models

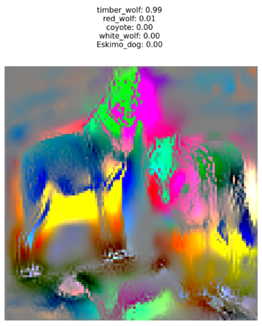
Original Input			
			
AlexNet			
coyote: 0.98 red_wolf: 0.01 grey_fox: 0.00 dingo: 0.00 timber_wolf: 0.00	red_wolf: 0.60 timber_wolf: 0.36 coyote: 0.04 Eskimo_dog: 0.00 malamute: 0.00	timber_wolf: 0.92 red_wolf: 0.04 coyote: 0.02 lynx: 0.01 malamute: 0.01	timber_wolf: 0.88 red_wolf: 0.11 coyote: 0.00 Eskimo_dog: 0.00 dhole: 0.00
			
GoogleNet			
coyote: 0.88 red_wolf: 0.03 timber_wolf: 0.03 grey_fox: 0.02 dhole: 0.01	timber_wolf: 0.81 coyote: 0.06 red_wolf: 0.03 Eskimo_dog: 0.02 Siberian_husky: 0.01	timber_wolf: 0.94 coyote: 0.03 red_wolf: 0.01 white_wolf: 0.01 dingo: 0.00	timber_wolf: 0.68 red_wolf: 0.13 coyote: 0.08 white_wolf: 0.02 dingo: 0.01
			
ResNet-18			
coyote: 0.95 grey_fox: 0.04 timber_wolf: 0.01 red_wolf: 0.00 red_fox: 0.00	timber_wolf: 0.84 red_wolf: 0.11 coyote: 0.02 Eskimo_dog: 0.01 Siberian_husky: 0.00	timber_wolf: 0.99 white_wolf: 0.01 coyote: 0.00 dingo: 0.00 red_wolf: 0.00	timber_wolf: 0.96 coyote: 0.02 red_wolf: 0.02 Eskimo_dog: 0.00 dingo: 0.00
			

ResNet-50

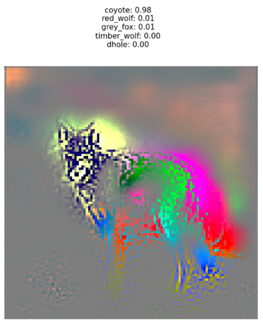


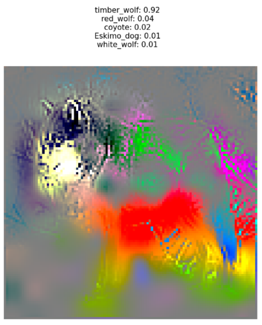


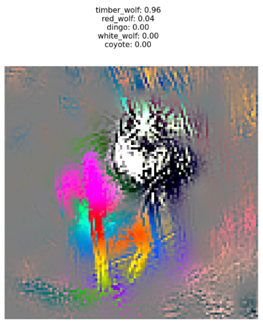


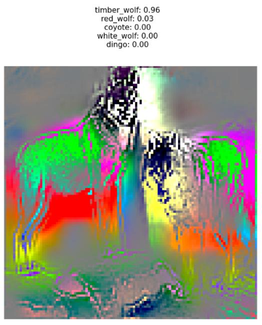


ResNet-101

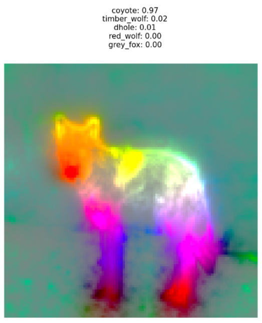


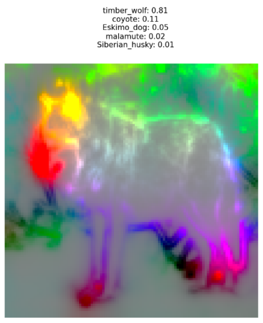


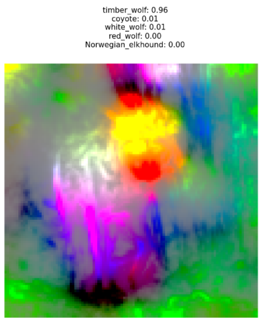


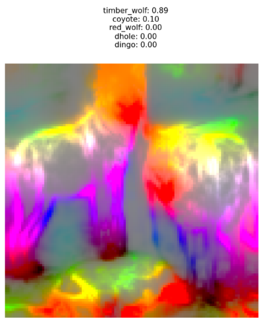


SqueezeNet

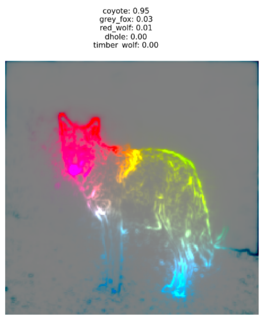


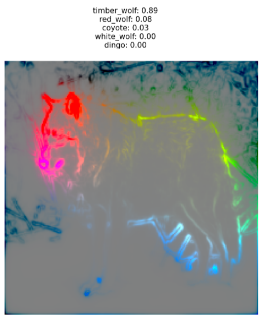


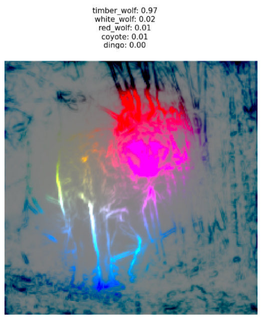


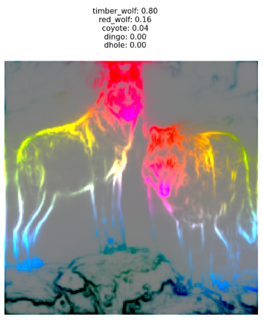


VGG-11

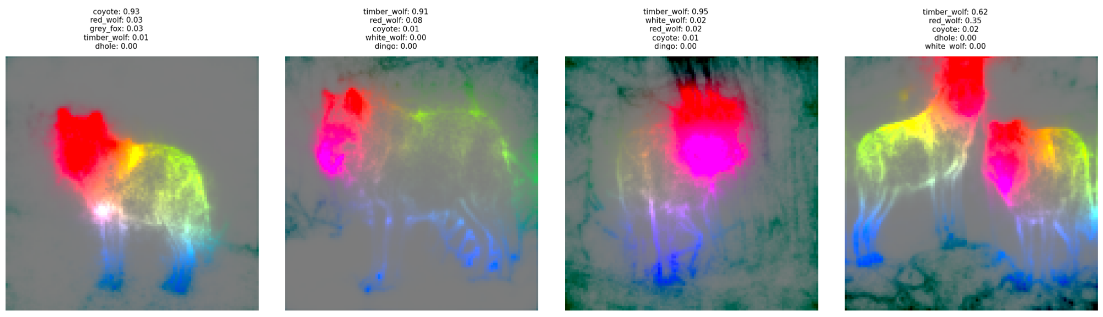




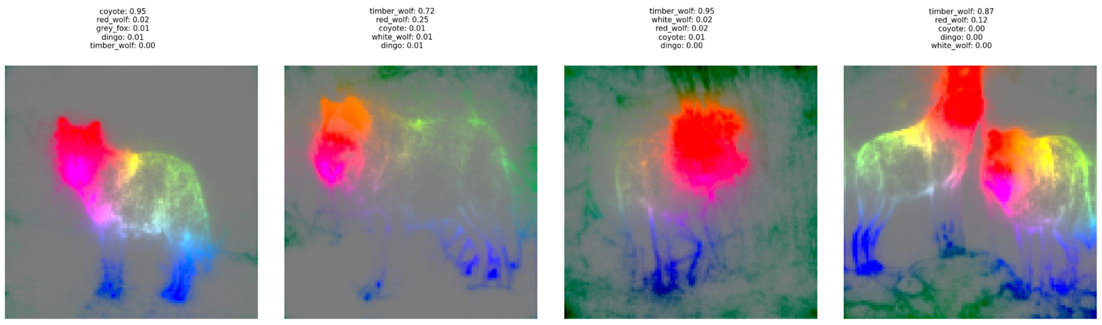




VGG-16



VGG-19



Results show that GE PRISM is applicable to all state-of-the-art models. Only minor concerns appear when considering the output from ResNet-18 and 50, which looks highly messy compared to others. Surprisingly ResNet-101's output seems more organized and more readable than its "simpler siblings". Probably a tuning in the procedure is required for this particular model family.

4.2.2.4 Immunity to Image Rotations

PRISM is also immune to the picture modifications as long as the model still generates correct representation. In the figure we see that wolves, which are recognizable after rotations, are maintaining the same set of features according to the model. While the identified features are the same - so behaves as expected, the combination of PRISM and GE for the first picture highlights the lower confidence of the model. The more uncertain the network is, the less explicit the output is.

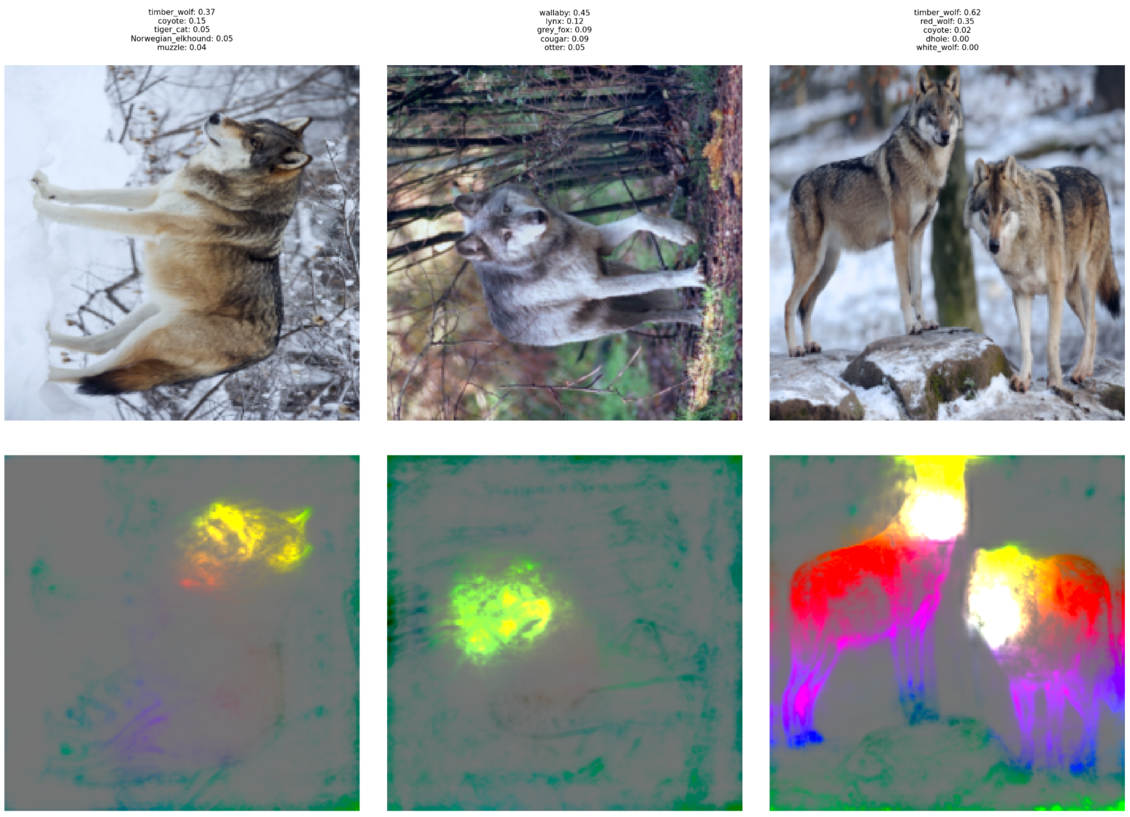


Fig. 45. PRISM immunity for rotation of pictures. As long as model still recognizes correct features the PRISM will highlight them accordingly

4.2.2.5 As a method to Monitor Training Process

With PRISM we can have an insight into untrained models. We have performed classification of 4 images with untrained VGG-16. The result (fig. 46) seems fuzzy but we can see that the untrained network is providing a readable output for each image. This means that we might be able to use PRIM to track and better see the training process of the model. Tho this approach requires further research.

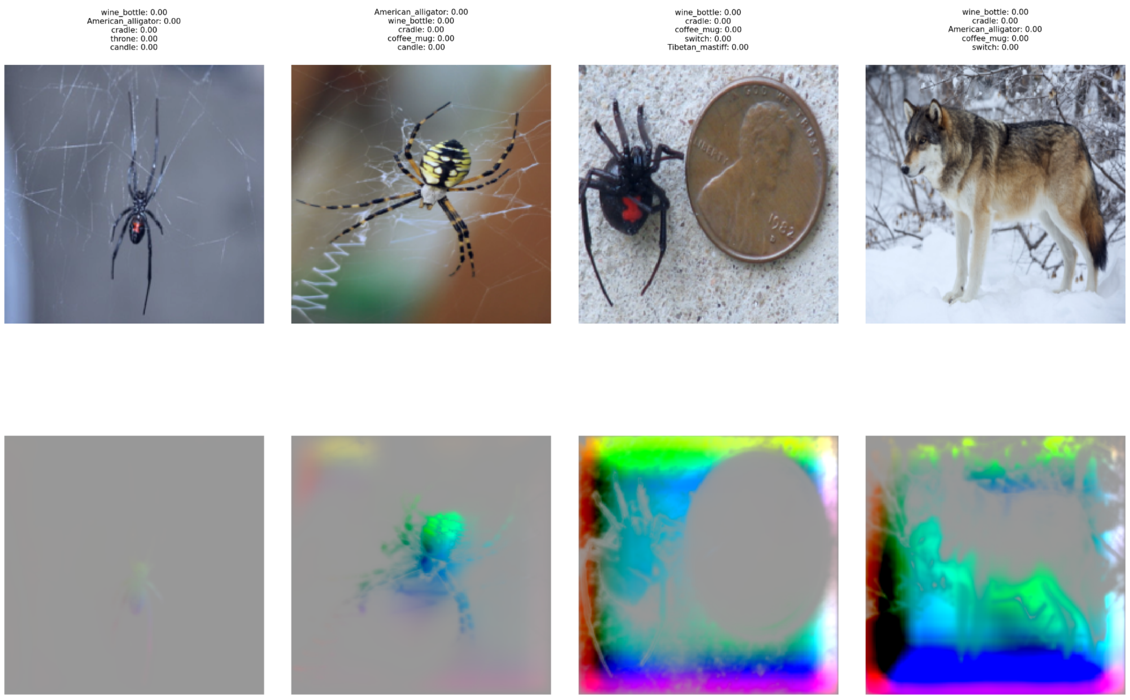


Fig. 46. PRISM output for an untrained model. The color palette seems to be evenly distributed and resembles a full spectrum of colors

4.2.2.6 Example that PRISM Lacks Built-in Saliency Indicators

It is mandatory to not consider PRISM as a replacement to commonly known visualization techniques like GradCAM or Excitation Backpropagation. Please see the image below - figure 47.

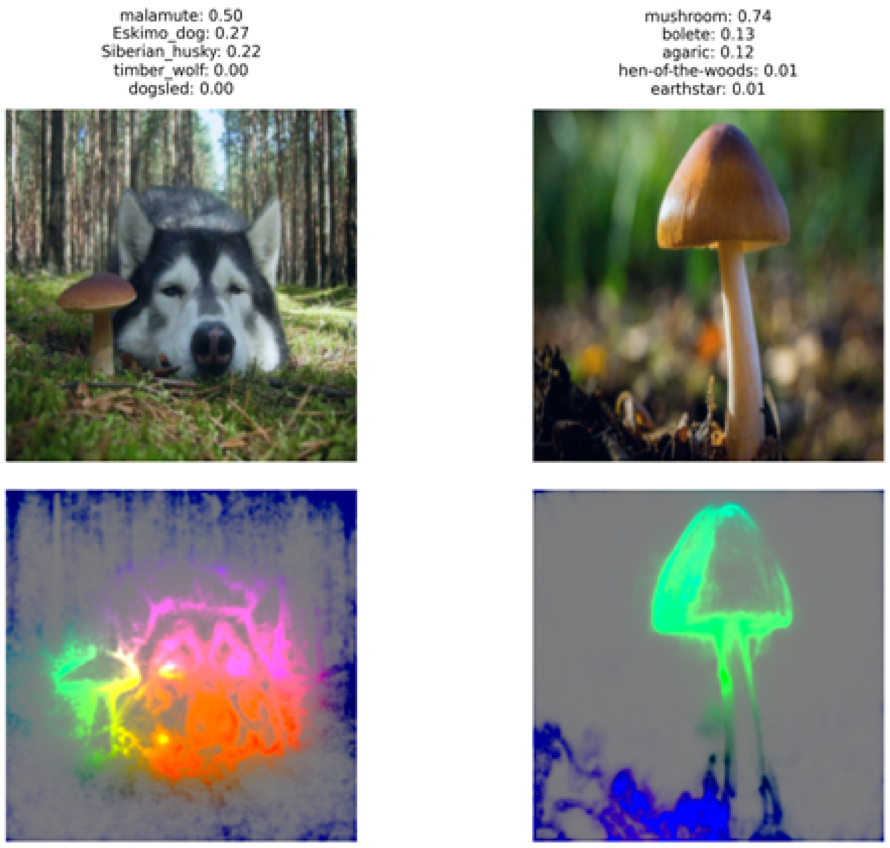


Fig. 47. PRISM output displaying detecting fungus features – a minor issue concerning the proposed method

PRISM is not a saliency indicator. In case of an image displaying 2 or more entirely different classes it will still highlight all detected features as seen in figure 47. Left image depicts a dog alongside a mushroom. Lack of saliency factor may suggest that both dog and mushroom are equally important for the network, still if we look at the classification scores we see that model mentions only canine classes to be identified on this image. The countermeasure for this could be blending a saliency method into PRISM's output which makes PRISM a complementary method to aforementioned saliency maps generators.

4.2.2.7 Summary

Name	Principal Image Sections Mapping
Method's application	
Identification of inclusive and exclusive images in a single batch of images.	
Procedure	
1. Perform classification as usual, but save output from the final layer (all layers if You aim to combine with GE) 2. Transform final layer's output into 2 dimensional matrix 3. Calculate PCA for obtained matrix 4. Recreate original shape using Principal Components matrix 5. **If** You wish to get human readable output: truncate chosen channels (usually 3 first) and assign them colors: red, green, blue 6. You can also apply Gradual Extrapolation **7.** **Else** use data from point 4. as input feature vectors to clustering method 8. Depict clustered images using e.g. Self-Organizing Maps	
Benefits	**Limitations**
● Supplementary method for saliency maps techniques - highlights different features with different colors ● Supplementary method for method Explanation by Example to highlight inclusive and exclusive features ● Method for comparing large scale datasets and their analysis with clustering - shows potentially ambiguous classes	● Does not highlight saliency of found features, only their presence ● When using for human interpretation virtually limited to only Principal Components ● Proposed clustering, using SOMs may always slightly return different results due to SOM's randomness

5 Conclusions

The goal of this dissertation was to study state-of-the-art CNN representation generating techniques, their visualization, as well identify the gaps and propose improvements/alternatives where possible.
The first identified problem was the coarseness of the output heatmaps. Saliency maps generated via GradCAM, Excitation Backpropagation etc. were mainly stains that only indicated the area where discriminative features are present. There was no well grounded method for deeper insight into the importance of specific parts of the object. **Solution here was the Gradual Extrapolation which uses sequential and weighted resizing of originally obtained heatmaps to narrow the approximative area into a shaped object.**

Even having the new method that appears to be an improvement over state-of-the-art there is demand to prove its effectiveness and usability. Therefore a group of tests has been devised. They test:

- **faithfulness**, defined as reliably and ability to comprehensively represent the local decision structure of the analyzed model. To assess such a property of the model, a proposed technique is “pixel flipping”. The pixel-flipping procedure assesses whether removing the features highlighted by the explanation, as the most relevant, decreases the network prediction abilities.
- **interpretability**, which is an ability to reduce the information from the original object and only retain the elements that play the highest role in classification. Based on this consideration, a test is established that computes the number of pixels of the original image remaining after its truncation only in salient areas.
- **applicability,** which is the check to determine whether the explanation method can be implemented in various models, at least to most state-of-the-art models, and whether the explanation can be obtained quickly enough with finite computational resources. To measure applicability, the proposed method is applied to several different models and their computational time is measured.

Proposed method can work as a quantitative, strict comparison and approval procedure for both old and new saliency map generating techniques.

Two next accomplishments were tied to demand for procedures for evaluation of large datasets trained models. First one, detecting spurious correlations, was addressed with evaluation of Network Attention Focus. We have indicated a Region-of-Interest area

where the classified object has been present and computed a fit metric, defined as ratio of saliency inside the ROI to saliency in the entire image. This procedure, applied to all validating images, indicated classes where researchers can expect significant impact of the context to the classification, instead of object-focuses reasoning.
This method allowed us to perform adversarial attacks by extracting one object from its natural context and pasting it into another context, thus cheating the network.
Proposed method allows researchers to identify classes prone to focusing on the context during classification instead of the actual object. Due to the semi-automation form it is applicable to even the extremely large training datasets.

Second automation related success is the automatic detection of latent features using ROI and Gradual Extrapolation enhanced GradCAM. Same as previously we employ Detectron2 tool to draw the Regions-of-Interest around potential objects. Next, we analyze the pictures using GE GradCAM and compute the metric: salient pixels inside ROI, divided by all pixels in ROI. This formula answers: how big part of the object actually plays a significant role in the classification. The low average for all images with given class in a validating set should trigger a deeper insight into given class. We have confirmed that some of the objects are classified only with a latent feature of the object instead of it as a whole - even for models from an official model ZOO repository.
Proposed method is another complementary (to accuracy/error rate metric) technique to evaluate obtained AI models. Despite two identified limitations it correctly indicates classes where trained models might be focused only on a subset of an object, instead of the entire object.

The final project answers the problem of features differentiation. While taking a look onto saliency maps for coyote and the wolf we see that in both cases the head was mainly highlighted. Therefore to answer the question whether both heads are literally the same for the network a Principal Image Sections Mapping (PRISM) has been introduced.
It takes the output of given layer, like saliency methods:preferable the last one, performs the Principal Components Analysis and thus obtained a ranked list of vectors that might be used for features discrimination and comparison. Moreover, all but the three most principal components can be removed giving us three unrelated vectors which we can assign a color: red, green and blue. That procedure results in a colorful map of the most discriminative features.

This method can now go two ways: automatic clustering or human interpretable features. First the clustering. Since we have representations converted to a subset of vectors that represent the RGB colors, we can use Self-Organizing maps to place each image in a 2 dimensional map. Thus, with a sufficient amount of images, we obtain a map of clusters and see potentially ambiguous classes. Equipped with this knowledge, the researcher should have a closer look into given classes to ensure their differentiation by the trained model.

On the other hand we can take a closer look into specific instances of the datasets. While applying Gradual Extrapolation to the checkered PRISM result we obtain an illustration of distinguishable features seen by the network. With that approach we can see inclusive and exclusive features according to our model and enhance the training procedure accordingly.

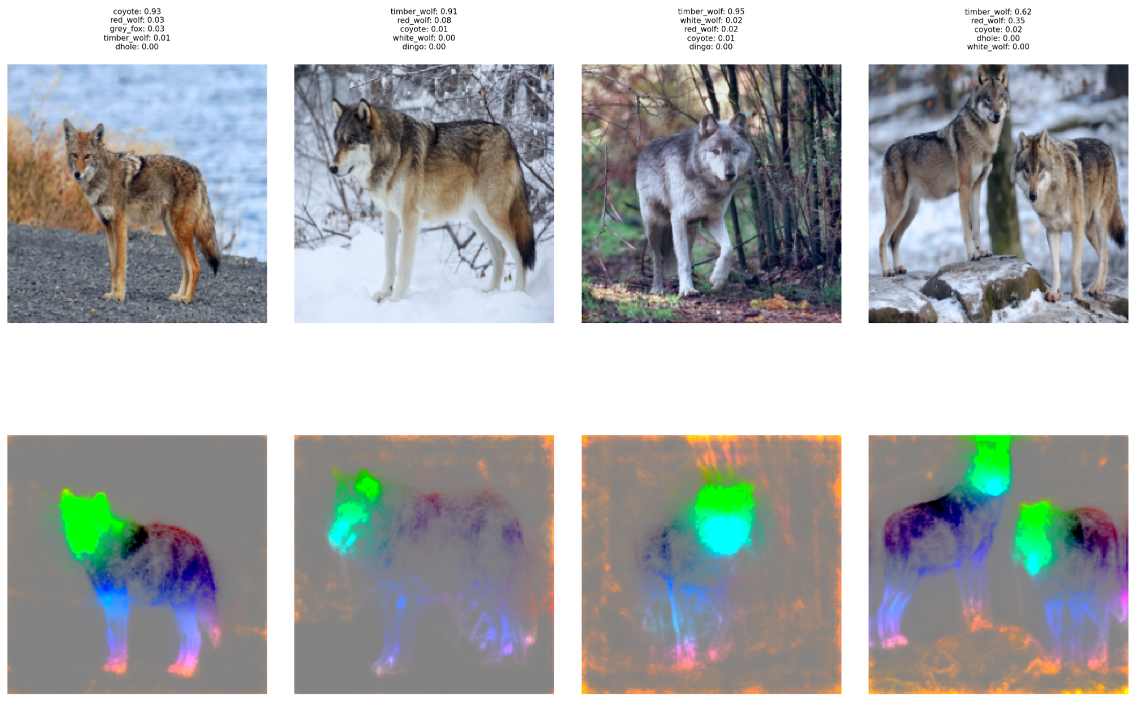


Fig. 48. Result of combining PRISM with GE. We can clearly see the coyote (first picture) differs from others by a part of its head. All wolves have a specific feature marked with light-blue stain on the muzzles

PRISM method enhances model's evaluation in two ways. First (automation) it can indicate classes that often come with similar features during classification and therefore are prone to misclassification between them. On the other hand PRISM allows a closer insight into inclusive and exclusive features identified by the evaluated model thus increasing the reasoning behind certain DCNN's classification.
Nonetheless, few aspects require more investigation: like taking into consideration more than 3 Principal Components or combining PRISM with saliency map methods to be able to rank identified features.

In the future I believe that several limitations of proposed methods should be addressed. Starting from the automatic evaluation techniques: we rely on rectangular ROI which might be modified into a more fitting shape. Moreover, instances where ROI does not know objects - currently we are virtually ignoring them. Perhaps a more sophisticated procedure could be devised.

On the other hand, the FIA-test has significant limitations about its portability. First it is designated only to picture-classification problems. There are other problems like text-classification, genomics, etc. While the idea of evaluating faithfulness, interpretability and applicability is applicable to them as well, the adjusted exact tests should be defined.

The most unknown is around PRISM. First one is the sticking to only 3 first Principal Components, which might not always be enough - it is possible that a more sophisticated procedure to choose PCs is required. Moreover, even left with 3 channels we are receiving different colors for each analyzed batch of images. Perhaps there is a possibility to fixate eigenvectors so that findings from one batch can be extrapolated to another set. Finally the clustering part is currently in the state of Proof of Concept. While it gave satisfying results thus far, it may be further enhanced to provide more detailed answers on overlapping classes. Perhaps the fuzzy sets could be used in this place? Nevertheless it requires more research which I am willing to perform in the future.

Since the naming Principal Image Sections Mapping with abbreviation PRISM I have identified several other entities with that name like:

- Prism - data analysis tool
- PRISM (Proteomic Investigation Strategy for Mammals)
- PRISM (Projects Integrating Sustainable Methods)

This may enforce the change of the name but as for today it serves its purpose well.

Overall contribution to the subject can be divided between reviewing state of the art, identification of gaps and proposing solutions to them. Finally a set of tests has been provided to quantitatively compare and evaluate interpretation techniques applicable to convolutional neural networks.

The main goal of this work is to convince AI practitioners to include more tools in their pursuit for better models. Not only accuracy/error rate matters. The newly introduced semi-automatic tests are recommended to detect if the model focuses on the object's context or an element of the object. On the other hand: Gradual Extrapolation and PRISM aim to provide deeper insight into the model's representation reasoning. Finally PRISM's clustering peculiarity is another semi-automation tool to examine a model's robustness. I sincerely recommend including at least some of my contribution into the standard AI researcher's toolkit.

Literature

Aas, K., Jullum, M. and Løland, A. (2021) ‘Explaining individual predictions when features are dependent: More accurate approximations to Shapley values’, *Artificial intelligence*, 298, p. 103502.

Abbas, Q., Ibrahim, M.E.A. and Jaffar, M.A. (2019) ‘A comprehensive review of recent advances on deep vision systems’, *Artificial Intelligence Review*, 52(1), pp. 39–76.

Alippi, C., Disabato, S. and Roveri, M. (2018) ‘Moving Convolutional Neural Networks to Embedded Systems: The AlexNet and VGG-16 Case’, in *2018 17th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. ieeexplore.ieee.org, pp. 212–223.

Barredo Arrieta, A. *et al.* (2020) ‘Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI’, *An international journal on information fusion*, 58, pp. 82–115.

Bertero, M., De Mol, C. and Viano, G.A. (1980) ‘The Stability of Inverse Problems’, in Baltes, H.P. (ed.) *Inverse Scattering Problems in Optics*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 161–214.

Biran, O. and Cotton, C. (2017) ‘Explanation and justification in machine learning: A survey’, in *IJCAI-17 workshop on explainable AI (XAI)*. cs.columbia.edu, pp. 8–13.

Böhle, M. *et al.* (2019) ‘Layer-Wise Relevance Propagation for Explaining Deep Neural Network Decisions in MRI-Based Alzheimer’s Disease Classification’, *Frontiers in aging neuroscience*, 11, p. 194.

Bromley, J. *et al.* (1993) ‘SIGNATURE VERIFICATION USING A “SIAMESE” TIME DELAY NEURAL NETWORK’, *International Journal of Pattern Recognition and Artificial Intelligence*, 07(04), pp. 669–688.

Calude, C.S. and Longo, G. (2017) ‘The Deluge of Spurious Correlations in Big Data’, *Foundations of science*, 22(3), pp. 595–612.

Campbell, M., Hoane, A.J. and Hsu, F.-H. (2002) ‘Deep Blue’, *Artificial intelligence*, 134(1), pp. 57–83.

Chattopadhay, A. *et al.* (2018) ‘Grad-CAM++: Generalized Gradient-Based Visual Explanations for Deep Convolutional Networks’, in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. ieeexplore.ieee.org, pp. 839–847.

Chen, C. *et al.* (2018) ‘This Looks Like That: Deep Learning for Interpretable Image Recognition’, *arXiv [cs.LG]*. Available at: http://arxiv.org/abs/1806.10574.

Chopra, S., Hadsell, R. and LeCun, Y. (2005) ‘Learning a similarity metric discriminatively, with application to face verification’, in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*. ieeexplore.ieee.org, pp. 539–546 vol. 1.

Das, A. *et al.* (2017) 'Human Attention in Visual Question Answering: Do Humans and Deep Networks Look at the Same Regions?', *Computer vision and image understanding: CVIU*, 163, pp. 90–100.

Deng, J. *et al.* (2009) 'ImageNet: A large-scale hierarchical image database', in *2009 IEEE Conference on Computer Vision and Pattern Recognition*. ieeexplore.ieee.org, pp. 248–255.

Deramgozin, M. *et al.* (2021) 'A Hybrid Explainable AI Framework Applied to Global and Local Facial Expression Recognition', in *2021 IEEE International Conference on Imaging Systems and Techniques (IST)*. ieeexplore.ieee.org, pp. 1–5.

Dong, C. *et al.* (2016) 'Image Super-Resolution Using Deep Convolutional Networks', *IEEE transactions on pattern analysis and machine intelligence*, 38(2), pp. 295–307.

Ferrando, P.J. and Lorenzo-Seva, U. (2017) 'Program FACTOR at 10: Origins, development and future directions', *Psicothema*, 29(2), pp. 236–240.

Ferrucci, D. *et al.* (2013) 'Watson: Beyond Jeopardy!', *Artificial intelligence*, 199-200, pp. 93–105.

Fidel, G., Bitton, R. and Shabtai, A. (2020) 'When Explainability Meets Adversarial Learning: Detecting Adversarial Examples using SHAP Signatures', in *2020 International Joint Conference on Neural Networks (IJCNN)*. ieeexplore.ieee.org, pp. 1–8.

Fong, R.C. and Vedaldi, A. (2017) 'Interpretable explanations of black boxes by meaningful perturbation', in *Proceedings of the IEEE international conference on computer vision*. openaccess.thecvf.com, pp. 3429–3437.

Fukushima, K. and Miyake, S. (1982) 'Neocognitron: A Self-Organizing Neural Network Model for a Mechanism of Visual Pattern Recognition', in *Competition and Cooperation in Neural Nets*. Springer Berlin Heidelberg, pp. 267–285.

Gao, H. *et al.* (2015) 'Are You Talking to a Machine? Dataset and Methods for Multilingual Image Question Answering', *arXiv [cs.CV]*. Available at: http://arxiv.org/abs/1505.05612.

Geirhos, R. *et al.* (2018) 'ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness', *arXiv [cs.CV]*. Available at: http://arxiv.org/abs/1811.12231.

Geirhos, R. *et al.* (2020) 'Shortcut learning in deep neural networks', *Nature Machine Intelligence*, 2(11), pp. 665–673.

Girshick, R. *et al.* (2014) 'Rich feature hierarchies for accurate object detection and semantic segmentation', in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580–587.

Goodfellow, I., Bengio, Y. and Courville, A. (2017) 'Deep learning. Adaptive computation and machine learning', *An MIT Press Book* [Preprint].

Granter, S.R., Beck, A.H. and Papke, D.J., Jr (2017) 'AlphaGo, Deep Learning, and the Future of the Human Microscopist', *Archives of pathology & laboratory medicine*, 141(5), pp. 619–621.

Gu, J. *et al.* (2018) ‘Recent advances in convolutional neural networks’, *Pattern recognition*, 77, pp. 354–377.

Gunning, D. *et al.* (2019) ‘XAI—Explainable artificial intelligence’, *Science Robotics*, 4(37), p. eaay7120.

Guo, Y. *et al.* (2016) ‘Deep learning for visual understanding: A review’, *Neurocomputing*, 187, pp. 27–48.

He, K. *et al.* (2016) ‘Deep residual learning for image recognition’, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.

Hinton, G.E. *et al.* (2012) ‘Improving neural networks by preventing co-adaptation of feature detectors’, *arXiv [cs.NE]*. Available at: http://arxiv.org/abs/1207.0580.

Hou, L. *et al.* (2021) ‘Deep Learning-Based Applications for Safety Management in the AEC Industry: A Review’, *NATO Advanced Science Institutes series E: Applied sciences*, 11(2), p. 821.

Huang, G. *et al.* (2017) ‘Densely connected convolutional networks’, in *Proceedings of the IEEE conference on computer vision and pattern recognition*. openaccess.thecvf.com, pp. 4700–4708.

Hubel, D.H. and Wiesel, T.N. (1968) ‘Receptive fields and functional architecture of monkey striate cortex’, *The Journal of physiology*, 195(1), pp. 215–243.

Iandola, F.N. *et al.* (2016) ‘SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size’, *arXiv [cs.CV]*. Available at: http://arxiv.org/abs/1602.07360.

Jacobsen, J.-H. *et al.* (2018) ‘Excessive Invariance Causes Adversarial Vulnerability’, *arXiv [cs.LG]*. Available at: http://arxiv.org/abs/1811.00401.

Jetley, S., Lord, N. and Torr, P. (2018) ‘With friends like these, who needs adversaries?’, *Advances in neural information processing systems*, 31. Available at: https://proceedings.neurips.cc/paper/2018/hash/803a82dee7e3fbb3438a149508484250-Abstract.html.

Jeyakumar, J.V. *et al.* (2020) ‘How can i explain this to you? an empirical study of deep neural network explanation methods’, *Advances in neural information processing systems* [Preprint]. Available at: https://drive.google.com/file/d/1sXKXtcljwJvVsCEDraHPUz4qboXEE526/view.

Johnson, J. and Karpathy, A. (2016) ‘Densecap: Fully convolutional localization networks for dense captioning’, *Proceedings of the IEEE* [Preprint]. Available at: http://openaccess.thecvf.com/content_cvpr_2016/html/Johnson_DenseCap_Fully_Convolutional_CVPR_2016_paper.html.

Karim, M. *et al.* (2020) ‘Deepcovidexplainer: Explainable covid-19 predictions based on chest x-ray images’, *arXiv preprint arXiv:2004. 04582* [Preprint]. Available at: https://saira.eco/open-access-covid-19/uploads/2004-04582-compressed-5ec6d5d41339f2.86983306.pdf.

Khan, A. *et al.* (2020) ‘A survey of the recent architectures of deep convolutional neural networks’, *Artificial Intelligence Review*, 53(8), pp. 5455–5516.

Kindermans, P.-J. *et al.* (2019) ‘The (Un)reliability of Saliency Methods’, in Samek, W. et al. (eds) *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. Cham: Springer International Publishing, pp. 267–280.

Kingma, D.P. and Welling, M. (2013) ‘Auto-Encoding Variational Bayes’, *arXiv [stat.ML]*. Available at: http://arxiv.org/abs/1312.6114v10.

Lapuschkin, S. *et al.* (2019) ‘Unmasking Clever Hans predictors and assessing what machines really learn’, *Nature communications*, 10(1), p. 1096.

LeCun, Y. *et al.* (1990) ‘Handwritten Digit Recognition with a Back-Propagation Network’, in Touretzky, D. (ed.) *Advances in Neural Information Processing Systems*. Morgan-Kaufmann. Available at: https://proceedings.neurips.cc/paper/1989/file/53c3bce66e43be4f209556518c2fcb54-Paper.pdf .

LeCun, Y., Kavukcuoglu, K. and Farabet, C. (2010) ‘Convolutional networks and applications in vision’, in *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*. ieeexplore.ieee.org, pp. 253–256.

Lee, H. *et al.* (2007) ‘Efficient sparse coding algorithms’, in *Advances in neural information processing systems*. researchgate.net, pp. 801–808.

Liu, T. *et al.* (2015) ‘Implementation of Training Convolutional Neural Networks’, *arXiv [cs.CV]*. Available at: http://arxiv.org/abs/1506.01195.

Liu, W. *et al.* (2017) ‘A survey of deep neural network architectures and their applications’, *Neurocomputing*, 234, pp. 11–26.

Lundberg, S.M. *et al.* (2020) ‘From Local Explanations to Global Understanding with Explainable AI for Trees’, *Nature machine intelligence*, 2(1), pp. 56–67.

Lundberg, S.M. and Lee, S.-I. (2017) ‘A unified approach to interpreting model predictions’, *Advances in neural information processing systems*, 30. Available at: https://proceedings.neurips.cc/paper/2017/hash/8a20a8621978632d76c43dfd28b67767-Abstract.html (Accessed: 29 April 2022).

Matcha, W. *et al.* (2020) ‘A Systematic Review of Empirical Studies on Learning Analytics Dashboards: A Self-Regulated Learning Perspective’, *IEEE Transactions on Learning Technologies*, 13(2), pp. 226–245.

Mehta, J. and Majumdar, A. (2017) ‘RODEO: Robust DE-aliasing autoencOder for real-time medical image reconstruction’, *Pattern recognition*, 63, pp. 499–510.

Montavon, G. *et al.* (2022) ‘Explaining the Predictions of Unsupervised Learning Models’, in Holzinger, A. et al. (eds) *xxAI - Beyond Explainable AI: International Workshop, Held in Conjunction with ICML 2020, July 18, 2020, Vienna, Austria, Revised and Extended Papers*. Cham: Springer International Publishing, pp. 117–138.

Nagendran, M. *et al.* (2020) ‘Artificial intelligence versus clinicians: systematic review of design, reporting standards, and claims of deep learning studies’, *BMJ* , 368, p. m689.

Nair, V. and Hinton, G.E. (2010) ‘Rectified Linear Units Improve Restricted Boltzmann Machines’, *Icml*. Available at: https://openreview.net/pdf?id=rkb15iZdZB (Accessed: 6 October 2021).

Najafabadi, M.M. *et al.* (2015) ‘Deep learning applications and challenges in big data analytics’, *Journal of Big Data*, 2(1), pp. 1–21.

Namatēvs, I. (2017) ‘Deep Convolutional Neural Networks: Structure, Feature Extraction and Training’, *Information Technology & Management Science (Sciendo)*, 20(1). Available at: http://search.ebscohost.com/login.aspx?direct=true&profile=ehost&scope=site&authtype=crawler&jrnl=22559086&AN=127283332&h=rYmhzVOHMZ0d%2FmWMHV2b7vOjjyf7%2B7rWofryXTSBgZwTzp7NBejC4MBWoaa7fwKj3D2G7mjrxHlTfa%2BeaNorUQ%3D%3D&crl=c.

Noh, H., Hong, S. and Han, B. (2015) ‘Learning deconvolution network for semantic segmentation’, in *Proceedings of the IEEE international conference on computer vision*. openaccess.thecvf.com, pp. 1520–1528.

Pan, M. *et al.* (2020) ‘Visual Recognition Based on Deep Learning for Navigation Mark Classification’, *IEEE Access*, 8, pp. 32767–32775.

Parsa, A.B. *et al.* (2020) ‘Toward safer highways, application of XGBoost and SHAP for real-time accident detection and feature analysis’, *Accident; analysis and prevention*, 136, p. 105405.

Pham, V., Pham, C. and Dang, T. (2020) ‘Road Damage Detection and Classification with Detectron2 and Faster R-CNN’, in *2020 IEEE International Conference on Big Data (Big Data)*, pp. 5592–5601.

Pope, P.E. *et al.* (2019) ‘Explainability methods for graph convolutional neural networks’, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. openaccess.thecvf.com, pp. 10772–10781.

Rawat, W. and Wang, Z. (2017) ‘Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review’, *Neural computation*, 29(9), pp. 2352–2449.

Ribeiro, M.T., Singh, S. and Guestrin, C. (2016) ‘“why should I trust you?”: Explaining the predictions of any classifier’, *arXiv [cs.LG]*. Available at: http://arxiv.org/abs/1602.04938.

Rosato, A. *et al.* (2021) ‘2-D Convolutional Deep Neural Network for the Multivariate Prediction of Photovoltaic Time Series’, *Basel*. search.proquest.com, p. 2021. doi:10.3390/en14092392.

Samek, W. *et al.* (2017) ‘Evaluating the Visualization of What a Deep Neural Network Has Learned’, *IEEE transactions on neural networks and learning systems*, 28(11), pp. 2660–2673.

Sandler, M. *et al.* (2018) ‘Mobilenetv2: Inverted residuals and linear bottlenecks’, in *Proceedings of the IEEE conference on computer vision and pattern recognition*. openaccess.thecvf.com, pp. 4510–4520.

Schott, L. *et al.* (2018) ‘Towards the first adversarially robust neural network model on MNIST’, *arXiv [cs.CV]*. Available at: http://arxiv.org/abs/1805.09190.

Schroff, F., Kalenichenko, D. and Philbin, J. (2015) ‘Facenet: A unified embedding for face recognition and clustering’, in *Proceedings of the IEEE conference on computer vision and pattern recognition*. cv-foundation.org, pp. 815–823.

Scott, A.C. *et al.* (1977) *Explanation capabilities of production-based consultation systems*. STANFORD UNIV CA DEPT OF COMPUTER SCIENCE. Available at: https://apps.dtic.mil/sti/citations/ADA042719.

Seibold, C., Hilsmann, A. and Eisert, P. (2021) ‘Focused lrp: Explainable ai for face morphing attack detection’, in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. openaccess.thecvf.com, pp. 88–96.

Selvaraju, R.R. *et al.* (2017) ‘Grad-cam: Visual explanations from deep networks via gradient-based localization’, in *Proceedings of the IEEE international conference on computer vision*. openaccess.thecvf.com, pp. 618–626.

Shahid, M. and Channappayya, S.S. (2021) ‘Surveying for man-made objects in photographic images’, in *Target and Background Signatures VII*. *Target and Background Signatures VII*, SPIE, pp. 118–127.

Simonyan, K. and Zisserman, A. (2014) ‘Very Deep Convolutional Networks for Large-Scale Image Recognition’, *arXiv [cs.CV]*. Available at: http://arxiv.org/abs/1409.1556.

Slack, D. *et al.* (2020) ‘Fooling LIME and SHAP: Adversarial Attacks on Post hoc Explanation Methods’, in *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*. New York, NY, USA: Association for Computing Machinery, pp. 180–186.

Smilkov, D. *et al.* (2017) ‘SmoothGrad: removing noise by adding noise’, *arXiv [cs.LG]*. Available at: http://arxiv.org/abs/1706.03825.

Soleimani, A. *et al.* (2018) ‘Convolutional Neural Networks for Aerial Vehicle Detection and Recognition’, in *NAECON 2018 - IEEE National Aerospace and Electronics Conference*. ieeexplore.ieee.org, pp. 186–191.

Sparkes, M. (2015) ‘Top scientists call for caution over artificial intelligence’, *The Times (UK)*, 24.

Springenberg, J.T. *et al.* (2014) ‘Striving for Simplicity: The All Convolutional Net’, *arXiv [cs.LG]*. Available at: http://arxiv.org/abs/1412.6806.

Swartout, W.R. (1985) ‘Explaining and Justifying Expert Consulting Programs’, in Reggia, J.A. and Tuhrim, S. (eds) *Computer-Assisted Medical Decision Making*. New York, NY: Springer New York, pp. 254–271.

Szandała, T. (2020a) ‘Benchmarking Comparison of Swish vs. Other Activation Functions on CIFAR-10 Imageset’, in *Engineering in Dependability of Computer Systems and Networks*. Springer International Publishing, pp. 498–505.

Szandała, T. (2020b) ‘Using Convolutional Network Visualisation to Determine the Most Significant Pixels’, in *Theory and Applications of Dependable Computer Systems*. Springer International Publishing, pp. 626–632.

Szandala, T. (2021a) ‘Enhancing deep neural network saliency visualizations with gradual extrapolation’, *IEEE Access*, 9, pp. 95155–95161.

Szandała, T. (2021) ‘Review and Comparison of Commonly Used Activation Functions for Deep Neural Networks’, *Bio-inspired Neurocomputing* [Preprint]. Available at: https://link.springer.com/chapter/10.1007/978-981-15-5495-7_11.

Szandala, T. (2021b) ‘TorchPRISM: Principal Image Sections Mapping, a novel method for Convolutional Neural Network features visualization’, *arXiv [cs.CV]*. Available at: http://arxiv.org/abs/2101.11266.

Szandała, T. and Maciejewski, H. (2021) ‘Automated Method for Evaluating Neural Network’s Attention Focus’, in *Lecture Notes in Computer Science*. Springer. Available at: https://link.springer.com/chapter/10.1007/978-3-030-77964-1_33.

Szandala, T. and Maciejewski, H. (2022) ‘PRISM: Principal Image Sections Mapping’, in *International Conference on Computational Science 2022*. *International Conference on Computational Science 2022*, Springer.

Szegedy, C. *et al.* (2015) ‘Going deeper with convolutions’, in *Proceedings of the IEEE conference on computer vision and pattern recognition*. cv-foundation.org, pp. 1–9.

Szyc, K. (2020) ‘An Impact of Different Images Color Spaces on the Efficiency of Convolutional Neural Networks’, in *Engineering in Dependability of Computer Systems and Networks*. Springer International Publishing, pp. 506–514.

Szyc, K., Walkowiak, T. and Maciejewski, H. (2021) ‘Checking Robustness of Representations Learned by Deep Neural Networks’, in *Machine Learning and Knowledge Discovery in Databases. Applied Data Science Track*. Springer International Publishing, pp. 399–414.

Tang, Y. (2013) ‘Deep Learning using Linear Support Vector Machines’, *arXiv [cs.LG]*. Available at: http://arxiv.org/abs/1306.0239.

Tomsett, R. *et al.* (2020) ‘Sanity Checks for Saliency Metrics’, *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04), pp. 6021–6029.

Vincent, P. *et al.* (2008) ‘Extracting and composing robust features with denoising autoencoders’, in *Proceedings of the 25th international conference on Machine learning*. New York, NY, USA: Association for Computing Machinery (ICML ’08), pp. 1096–1103.

Visani, G. *et al.* (2022) ‘Statistical stability indices for LIME: Obtaining reliable explanations for machine learning models’, *The Journal of the Operational Research Society*, 73(1), pp. 91–101.

Visani, G., Bagli, E. and Chesani, F. (2020) ‘OptiLIME: Optimized LIME Explanations for Diagnostic Computer Algorithms’, *arXiv [cs.LG]*. Available at: http://arxiv.org/abs/2006.05714.

Wu, Y. *et al.* (2019) ‘Detectron2’.

Xiao, K. *et al.* (2020) ‘Noise or Signal: The Role of Image Backgrounds in Object Recognition’, *arXiv [cs.CV]*. Available at: http://arxiv.org/abs/2006.09994.

Yamashita, R. *et al.* (2018) ‘Convolutional neural networks: an overview and application in radiology’, *Insights into imaging*, 9(4), pp. 611–629.

Yu, W. *et al.* (2016) ‘Visualizing and comparing AlexNet and VGG using deconvolutional layers’, in *Proceedings of the 33 rd International Conference on Machine Learning*. icmlviz.github.io. Available at: https://icmlviz.github.io/icmlviz2016/assets/papers/4.pdf.

Zeiler, M.D. and Fergus, R. (2013) ‘Stochastic Pooling for Regularization of Deep Convolutional Neural Networks’, *arXiv [cs.LG]*. Available at: http://arxiv.org/abs/1301.3557.

Zeiler, M.D. and Fergus, R. (2014) ‘Visualizing and Understanding Convolutional Networks’, in *Computer Vision – ECCV 2014*. Springer International Publishing, pp. 818–833.

Zhang, J. *et al.* (2018) ‘Top-Down Neural Attention by Excitation Backprop’, *International journal of computer vision*, 126(10), pp. 1084–1102.

Zhang, Y., Lee, K. and Lee, H. (2016) ‘Augmenting Supervised Neural Networks with Unsupervised Objectives for Large-scale Image Classification’, in Balcan, M.F. and Weinberger, K.Q. (eds) *Proceedings of The 33rd International Conference on Machine Learning*. New York, New York, USA: PMLR (Proceedings of Machine Learning Research), pp. 612–621.

Zhang, Z. *et al.* (2019) ‘Contextual Local Explanation for Black Box Classifiers’, *arXiv [cs.LG]*. Available at: http://arxiv.org/abs/1910.00768.

Zhou, B. *et al.* (2016) ‘Learning deep features for discriminative localization’, in *Proceedings of the IEEE conference on computer vision and pattern recognition*. openaccess.thecvf.com, pp. 2921–2929.