Submitted by
Viktor Maximilian Loreth
k12006268

Submitted at
Institute of Computational Perception

Supervisor
Katharina Hoedt, PHD

August 31, 2023

EVALUATION OF IMAGE RECOGNITION NEURAL NETWORK INTERPRETATION METHODS: AN IN-DEPTH LOOK

Bachelor Thesis

to obtain the academic degree of

Bachelor of Science

in the Bachelor's Program

Artificial Intelligence

JOHANNES KEPLER
UNIVERSITY LINZ

Sworn Declaration

I hereby declare under oath that the submitted Bachelor Thesis has been written solely by me without any third-party assistance, information other than provided sources or aids have not been used and those used have been fully documented. Sources for literal, paraphrased and cited quotes have been accurately credited.

The submitted document here present is identical to the electronically submitted text document.

Linz, August 31, 2023

JOHANNES KEPLER
UNIVERSITY LINZ

Abstract

This paper focuses on the accuracy of interpretability methods for machine learning models. The main research problem is the lack of ground truth for evaluation method in interpretability methods. While there are inherent interpretative models, black-box networks perform better and have developed rapidly. Existing interpretability methods for neural networks such as ROAR, KAR, BAM and Real Time Image Saliency for Black Box Classifiers provide a numerical evaluation method but they do still suffer from ambiguity. This paper summarizes the different evaluation methods, compares them and also calculate ROAR on 2 different saliency methods.

Contents

1 Introduction **1**
1.1 Structure of the thesis 2

2 Machine Learning and their Interpretability **3**
2.1 Supervised Machine Learning 4
2.1.1 Linear Models 5
2.1.2 Distance-Based methods 6
2.1.3 Decision Tree-Based Methods 6
2.1.4 Neural Networks 7

3 Interpretation of Neural Networks **8**
3.1 Global Model-Agnostic Methods 8
3.2 Local Model-Agnostic Methods 9
3.3 Neural Network Interpretation: Model specific Interpretations 9
3.3.1 Feature visualization and Network Dissection [25] 10
3.4 Saliency Maps 11
3.4.1 Vanilla Gradient & Deconv Net 12
3.4.2 Grad-CAM, Guided Grad-CAM and Smooth Gradient 13
3.4.3 Top-Down Neural Attention by Excitation Backprop [37] 13
3.4.4 Advantages and Disadvantages of Saliency Maps 14

4 Evaluation of post-hoc interpretability methods **15**
4.1 Evaluation metrics, ground truths [13] 15
4.2 Completeness and Soundness 15
4.3 Perturbation based [30] 17
4.4 A benchmark for interpretability methods in deep neural networks [15] 17
4.5 Benchmarking Attribution Methods (BAM) [34] 18
4.6 Sanity Checks for Saliency Maps [1] 19
4.7 Validating the research 20
4.8 Research findings: Which evaluation method to use, how to control the result? . . . 21

5 Reconstructing ROAR **23**
5.1 Scientific Motivation and Goal 23
5.1.1 Project Setup MNIST 23
5.1.2 Project Setup Food-101 25
5.1.3 Interpretation of the results 26

List of Figures

2.1 Feature Visualization [2] 4
2.2 Decision Tree Example: By Gilgoldm - Own work, CC BY-SA 4.0, https://commons.wikimedia.org

3.1 Feature Visualization [25] 10
3.2 Activation Maximization [25] 11
3.3 Saliency Map - Source: https://captum.ai/tutorials/Resnet_TorchVision_Interpret . 12
3.4 CNN classifier's top-down attention map [37] 13
3.5 Identifying task-relevant neurons in the network. The red shading of a dot indicates its relative likelihood of winning against the other ones in the same layer. [37] 14

4.1 Cascading Randomization on Image Net. The figure shows the original explanations. Progression from left to right indicate complete randomization of network weights up to that block inclusive. The last block corresponds to a network with completely reinitialized weights.[1] 20

5.1 Integrated Gradients and Random Baseline Comparison 24
5.2 Accuracy: Random Baseline vs Integrated Gradient 25

1 Introduction

Artificial Intelligence (AI) has undergone rapid development in the last years. In today's modern era of mobile phones and computers, algorithm's are used on a daily basis to have quick access to information and improve the efficiency of the daily life.

While various Algorithms (e.g.: Decision Trees, Linear Regression, Support Vector Machines, etc.), which are comprehensible by design, have been developed, the spotlight has turned to Deep Neural Networks(DNN). This shift is attributed to the increase in computational power and the exponential increase in accessible data. Despite their remarkable accuracy, DNN remain opaque black-boxes, which we struggle to understand. Nevertheless, the immense improvement in performance and their ability to handle massive datasets have led to widespread adoption in contemporary devices. It is predicted, that algorithms based on Neural Networks will be becoming increasingly popular in the next years.

However, one of the primary difficulties with Neural Networks is the lack of reliable interpretation techniques. Numerous interpretation methods exist, yet a universally reliable method remains missing. Particularly in the domain of image analysis, encompassing critical applications like automated driving and facial recognition, no solution is present. The decision-making rationale of neural networks remain unclear, attributed to factors like background elements, peripheral objects or lighting conditions. Efforts to address this issue have given rise to gradient methods, aiming to assign significance values to pixels and represent their importance on neural network decisions. Another alternative option to mitigate the black-box nature of algorithms involves employing model-agnostic methods. These methods offer an computational linkage between inputs and outputs, irrespective which model is used. Although highly effective for smaller datasets, they begin to struggle as the data size and their complexity increases. Because of this, they do not offer a reliable way to quickly make Neural Networks interpretable.

In light of these prevalent problems, the object of this thesis is to recapitulate interpretation algorithms for neural networks in computer vision. Emphasis is placed on the evaluation of post-hoc interpretability techniques, forecasting potential future developments and focusing on the strengths and weaknesses of distinct techniques. Concluding the theoretical segment, a practical demonstration showcasing the application of ROAR is shown.

1.1 Structure of the thesis

1. The first part presents a overview of contemporary machine learning algorithms, categorizing them into two main groups: algorithms with inherent interpretability and those without. The goal is to make clear how supervised methods can be applied to image recognition tasks.
2. Subsequent sections delve into interpretability, emphasizing global and local model-agnostic techniques. These methods offer insights into overall model behavior, regardless of algorithm specifics.
3. Additionally, in the domain of interpretability techniques for Neural Networks, the paper explores ad-hoc methods for Neural Networks. Features visualization and Gradient-focused methods are explained.
4. After introducing existing interpretation methods, the paper's focus transitions to evaluating post-hoc interpretation methods. Various approaches to assess the effectiveness and dependability of these methods in offering meaningful insights into intricate models are introduced and discussed. Additionally, the advantages and disadvantages of these approaches are carefully examined to provide a comprehensive understanding of their applicability.
5. To exemplify the discussed concepts, the practical application of the ROAR methodology using the Food-101 data set [5] and MNIST dataset [8] is presented. This real-world instance illustrates the current state of art of interpretability techniques in image recognition.

2 Machine Learning and their Interpretability

In the rapidly evolving landscape of machine learning, interpretability has emerged as important concept. Before going in-depth into various algorithms and methods, the fundamental question of: "What is interpretable Machine Learning (IML) and why do we need it?" should be answered. A broad definition of IML given by [2]: "Interpretable machine learning is the use of machine learning techniques to generate human-understandable insights into data, the learned model, or the model output." This definition underscores the important role of understanding complex models and allowing humans to understand the computation.
Interpretability is important for various reasons, ranging from model validation and debugging to fostering validation and trust. Following key objectives have been identified: [23] [27] [22] [12] [19] [10] [6] [9]

Model Validation: Interpretable models are essential for validating (by a human) whether a learned model behaves as expected and consistently aligns with prior expectations and knowledge about the system.

Model Debugging: When unexpected behavior occurs, finding the reasons for fault is impossible without understanding the system. Interpreting and understanding machine learning systems is critical for diagnosing, debugging and fixing systems. [16]

Transparency, Accountability & Trust: IML transforms black-box machine learning systems into understandable systems. The utilization in high-stakes societal applications requires accountability and trust of machine learning systems. [28] [29] [33]

Ethics: Machine learning algorithms can possibly be trained on biased data leading to unfair predictions that are discriminatory. To improve the fairness of machine learning algorithms intepretable methods need to be deployed. [12]

Data Exploration and Discovery: Insights into major patterns, trends, groups, or artifacts of the data are achieved by applying human-interpretable techniques. This data exploration insights influence the data pre-processing and model decisions. [23] [3] [4]

Interpretability methods can be assigned to different categories:

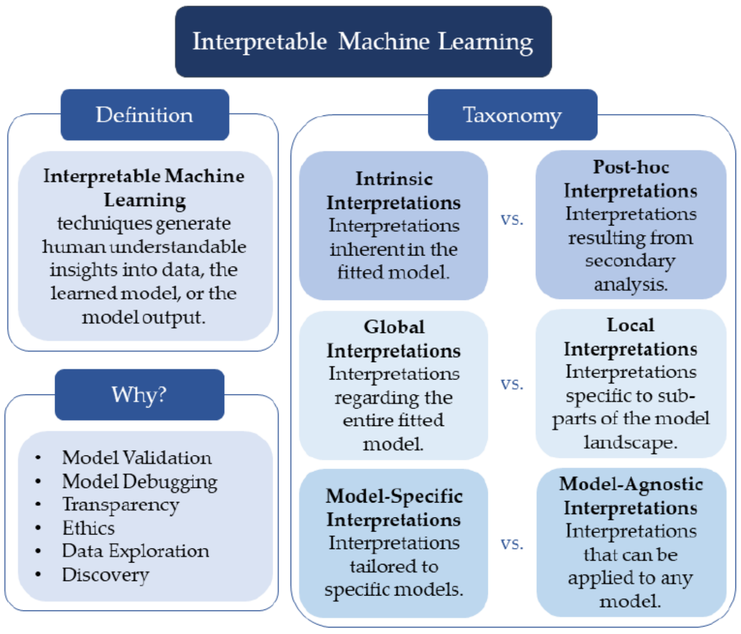
Interpretable Machine Learning
Definition
Interpretable Machine Learning techniques generate human understandable insights into data, the learned model, or the model output.
Why?
Model Validation
Model Debugging
Transparency
Ethics
Data Exploration
Discovery
Taxonomy
Intrinsic Interpretations Interpretations inherent in the fitted model.
vs.
Post-hoc Interpretations Interpretations resulting from secondary analysis.
Global Interpretations Interpretations regarding the entire fitted model.
vs.
Local Interpretations Interpretations specific to sub-parts of the model landscape.
Model-Specific Interpretations Interpretations tailored to specific models.
vs.
Model-Agnostic Interpretations Interpretations that can be applied to any model.

Figure 2.1: Feature Visualization [2]

Intrinsic vs Post-hoc: Interpretability methods can be broadly classified into intrinsic and post-hoc techniques. Intrinsic methods are inherently understandable by design, while post-hoc interpretations involve analyzing the model's behavior after its creation.
Global Interpretations vs Local Interpretations[22]: Global Interpretations encompass the entirety of a fitted model. In contrast, local interpretations zoom in on specific portions of the model landscape, such as class boundaries
Model-Specific Interpretations vs Model-Agnostic Interpretations [22]: Model-specific interpretations are tailored to a particular class of algorithms, like gradient methods for neural networks. In contrast, model-agnostic interpretations, such as LIME or SHAP, can be applied across any classification algorithm.

Before going into detail into the different methods of interpretability an overview of current supervised machine learning is given. Unsupervised machine learning is arguable intrinsically understandable, as the goal is to find a structure in the data. [2] With those examples, the necessity of intepretations for neural networks should be made evident.

2.1 Supervised Machine Learning

In supervised machine learning, there are a range of foundational classification methods. They are briefly introduced and analyzed for their interpretability. This section analyzes the base functionality of each algorithm. Furthermore, an analysis of the interpretability from a human perspective

is made. Due to the rapid development of methods, only the most prevalent algorithms are covered.
The interpretability of algorithms depends on two dimensions: Algorithm Transparency, how does the model learn the underlying structure of the data? Is it possible for a human to understand the implications of the mathematical operations? [22] The second dimension are the model weights: The parameters of a linear regression models are understood easily. Understanding millions of kernel-weights in a neural network is impossible.

2.1.1 Linear Models

When predicting outcomes, one of the simplest methods is to use a linear regression model. This model predicts by adding up features multiplied by an individual weight. The predictive output $\hat{y}$ is calculated:

$$\hat{y} = \alpha_0 + \alpha_1 x_1 + \alpha_2 x_2 + ... + \alpha_n x_n + \epsilon$$

The alphas α_i indicate the significance of each feature. The initial coefficient α_0 is known as the intercept, signifying the baseline. The noise ϵ describes the inevitable errors from inherent non-linearity in real-world dynamics or measurement inaccuracies.
To train the model, the MSE-Loss or the absolute loss can be applied. When using regularization methods, the absolute loss is more resilient to outliers.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

$$\text{ABS} = \frac{1}{n}\sum_{i=1}^{n} y_i - \hat{y}_i$$

The interpretability of the model is simple. The factors are described through the coefficient matrix. Each feature is distinctive to the model and the weighting is visible in the factors α_{i-n} (assuming normalization).

$$\alpha = \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ ... \\ \alpha_n \end{bmatrix}$$

Although linear models possess comprehensibility, provide a straightforward method for prediction and are inherently understandable, their application is limited to linear relationships and small datasets. One can use advanced techniques like L1 and L2 regularization [11] to achieve regularization against outliers and in case of correlation between factors. However, in the domain of image recognition, it is not applicable.

2.1.2 Distance-Based methods

K-Nearest Neighbors server as a classification method by considering the nearest neighbors. Although KNN is not interpretable by default due to the absence of parameters for learning and analysis, its underlying concept is straightforward. While on is able to visualize fewer features or use clustering algorithms to reduce the dimensionality, the relationship between input and output remains unclear. KNN also encounter difficulties when dealing with many features and larger datasets, therefore using it for image recognition is not recommended.
Support Vector Machines (SVM) aim to identify a hyperplane which maximizes the margin between different classes of data points. The optimization problem SVM solves can be summarized as:

$$\text{Minimize } \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i$$

$$\text{Subject to } y_i(w \cdot x_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0 \text{ for } i = 1, \ldots, n$$

Through the use of the kernel trick, SVM can be used for non-linear data. In higher dimensionality, SVM becomes non-interpretable, as displaying the weight matrix is not understandable. SVM also struggles with bigger datasets and growing feature sizes, contributing to its limited suitability for image classification.

2.1.3 Decision Tree-Based Methods

Decision Trees are inherently interpretable due to their transparent structure. However, as the depth increases, the models become less understandable. An illustrative example of the titanic dataset can be seen in 2.2. Their limited generalization method gave rise to ensemble methods.

Random forests are an ensemble of multiple decision trees. Their advantage is a smoother predictive power. However, by combing various algorithms the interpretability shrinks. Techniques such as SHAP values [20] or partial dependence trees can be employed to make them more understandable.
Gradient boosting like XGBoost [**Chen_2016**], LightGBM [**Ke2017**] and CatBoost [**prokhorenkova2019catbc**
share similarities with random forests and decision trees, but they assign differential learned weights to each decision. They suffer from the same interpretability issues as random forests.

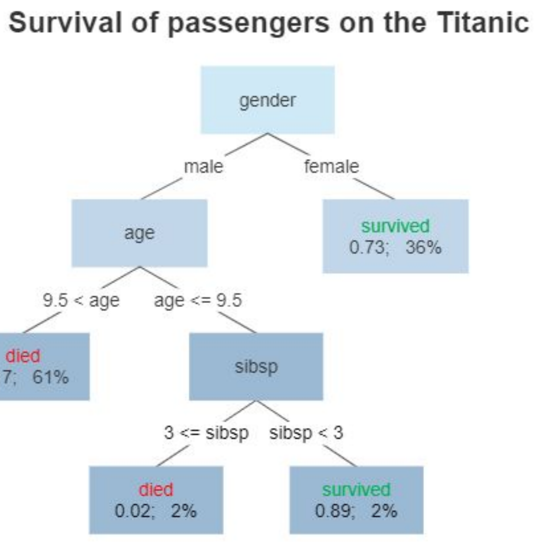

Figure 2.2: Decision Tree Example: By Gilgoldm - Own work, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=90405437

While decision tree-based methods can be applied for image classification, their accuracy tends to be worse than in neural networks.

2.1.4 Neural Networks

The rise of neural networks and their strong predictive power makes them a common choice for classification task. But as they grow in complexity, the traditional approach of comprehending through weight examination becomes impossible. With the rise of CNN in 2012 [17] Neural Networks have become state-of art-for image prediction. Special interpretability methods which consider the learned weights are explored in detail in section 3.3.

3 Interpretation of Neural Networks

In this chapter, evaluation method are described to analyze the behavior of neural networks.

3.1 Global Model-Agnostic Methods

Global model-agnostic methods describe expected outcomes based on the distribution of the data. They can show a correlation between singular or multiple features and an outcome.

1. **Partial Dependency Plots:** Partial Dependency Plots (PDPs) are a straightforward method to display the relationship between individual or multiple features and an outcome. However, when dealing with a bigger feature size, there are too many plots to make sense out of the data. Therefore they are not feasible for Neural Networks and image recognition tasks.
2. **Accumulated Local Effects:** Accumulated Local Effects (ALE) are an an advancement of PDP. While it overcomes certain difficulties from PDPs, they struggle from the same problem when it comes to a bigger feature size.
3. **Feature Interaction:** Feature Interaction analyzes the interaction between features inside the model. The underlying Friedman's H-statistic offers a method to evaluate the correlation and variance. Because of the complexity of image tasks and the high computational costs, this method is not applicable to Neural Networks in a meaningful matter.
4. **Functional Decomposition:** Functional Decomposition is commonly used in Neural Networks. In Chapter 3.3.1 a method to disassemble networks is presented.
5. **Permutation Feature Importance:** Permutation Feature Importance is regularly used in visual machine learning tasks. In section 4.3 an example for a perturbation method is shown.
6. **Prototype and Criticism:** Prototype and Criticism can be used as adversarial attacks in Neural Networks. [**xu2019adversarial**] Evaluation methods use the underlying idea of prototype and criticism to evaluate saliency maps.

3.2 Local Model-Agnostic Methods

To explain individual predictions, Local model-agnostic methods are used. A single outcome is correlated to some features and explain the model.

1. **LIME: Local Interpretablle Model-agnostic Explanations:** Lime generates locally faithful explanations by training interpretable models on perterurbed instances of the original data.
2. **Local surrogate models:** Local surrogate models can be programmed to select a singular instance to explain.
3. **Scoped Rules (Anchors):** Find so called anchors to explain the predictions.
4. **Individual conditional expectation curves:** Practically not useful in image recognition, as the computation is too high.
5. **Counterfactual explanations:** Try to find a change while still resembling the original image.
6. **SHAPly:** Calculate SHAP values for an image prediction to determine how each pixel contributes to the prediction's deviation from the average prediction across all images. Positive SHAP values indicate pixels that push the prediction up, while negative values indicate pixels that pull it down.

3.3 Neural Network Interpretation: Model specific Interpretations

In the domain of Natural Language Processing (NLP) and Computer Vision, Deep Learning has proven very successful. By passing the input data through a sequence of layers, characterized by matrix-multiplications with kernel weights and nonlinear transformations functions, a prediction is computed. Depending on the specific task, additional elements like Long Short-Time Memory(LSTM) layers and Convolutional layers (CNN) are utilized. Given the immense amount of mathematical operations underlying a single prediction, humans are not fit to apprehend the mapping. To interpret predictions, we would have to decipher the intricate learned knowledge of numerous different kernels and weights. Recognizing that it's impossible for humans to grasp millions of weights, the demand for evaluation methods is high. To assess the behavior and predictions of Deep Neural networks, specific interpretation methods were developed. These methods calculate the likelihood of an input entry being responsible for the result.

While model-agnostic methods offer an approach to understand Neural Networks, the sheer size of the data used to train and test Neural Networks make this task extremely hard. For instance, an image with the dimensions of 3x224x224, as commonly encountered in Food-101,the data entries

exceed 150.000. In NLP tasks, where vocabularies often encompass around 20.000 words, the computational complexity renders most model-agnostic techniques as too expensive.
In the pursuit of comprehending the complexity of Deep Neural Networks it makes sense to utilize the weights in the model. The information saved in the hidden layers as learned weights can be used to evaluate the network. Moreover, the gradients can be taken into consideration as well. In the following subsections several concepts for understanding Deep Neural Networks are introduced.

3.3.1 Feature visualization and Network Dissection [25]

Modern Neural Networks like ResNet50 or Bard consist of several million layers. Network dissection attempts to overcome this challenge by breaking down separate layers and connecting them with ideas.

The higher-level features in these networks relate to clear concepts, shown in Figure 3.1. As the input image moves through layers, it changes at each layer. In each convolutional layer, the network gains new and more complex features. The smooth joining of fully connected layers then changes image-based data into predictions.

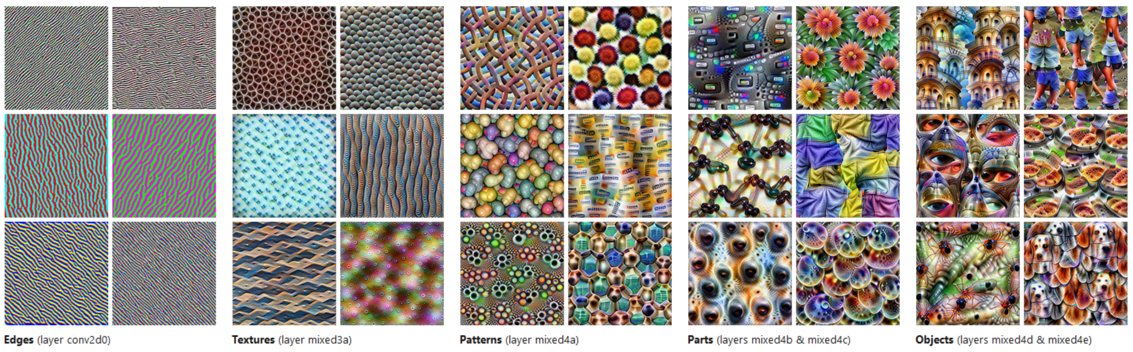

Figure 3.1: Feature Visualization [25]

The image explains this process. The first convolutional layers find simple features like edges and basic textures. Later, they recognize more detailed patterns. The deepest layers learn about parts and objects. This object information passes to the other hidden layers, which then finally make a prediction.

Feature visualization is based on activating one kernel in the network. This involves maximizing the activation of a specific neuron (Visible in 3.2). There are two methods for achieving this. First, we can make use of the training image that triggers the highest activation. Yet, this approach faces a significant problem. When an image contains multiple objects, it's hard to pinpoint which object causes the activation. Because of this an alternative route is adopted: generating new images from random noise. This is accomplished through methods like Generative Adversarial Networks (GANs) or other diffusion-based techniques.

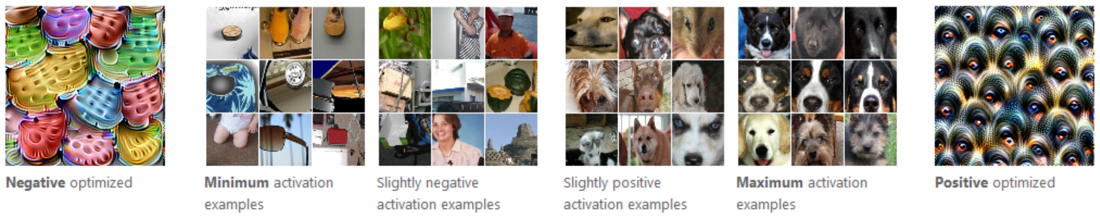
Negative optimized
Minimum activation examples
Slightly negative activation examples
Slightly positive activation examples
Maximum activation examples
Positive optimized

Figure 3.2: Activation Maximization [25]

Advantages of Feature Visualization:

1. **Initial Model Insights:** Feature visualization offer an initial view into a model's behavior, improving the understanding of its inner layers.
2. **Enhanced Domain Understanding:** It has the potential to enrich domain understanding by aligning learned features with domain-specific knowledge. An example can be seen in the medical industry
3. **Debugging and Improvement:** Feature visualization assists in debugging and refining models, contributing to their overall performance enhancement.

Disadvantages of Feature Visualization:

1. **Unclear Decision-Making:** While activations are evident, understanding the meaning behind them and how they contribute to decision-making remains challenging.
2. **Subjective Interpretation:** The interpretation of visualized features can be subjective, potentially leading to differing conclusions among observers.
3. **Limited Applicability to Visual Data:** Feature visualization's applicability is limited to visual data types.

3.4 Saliency Maps

Saliency maps are visualizations that highlight the regions of an input image that have the most significant impact on a model's output. By revealing the areas that strongly influence a prediction, saliency maps bridge the gap between the model's "black-box" nature and human understanding.

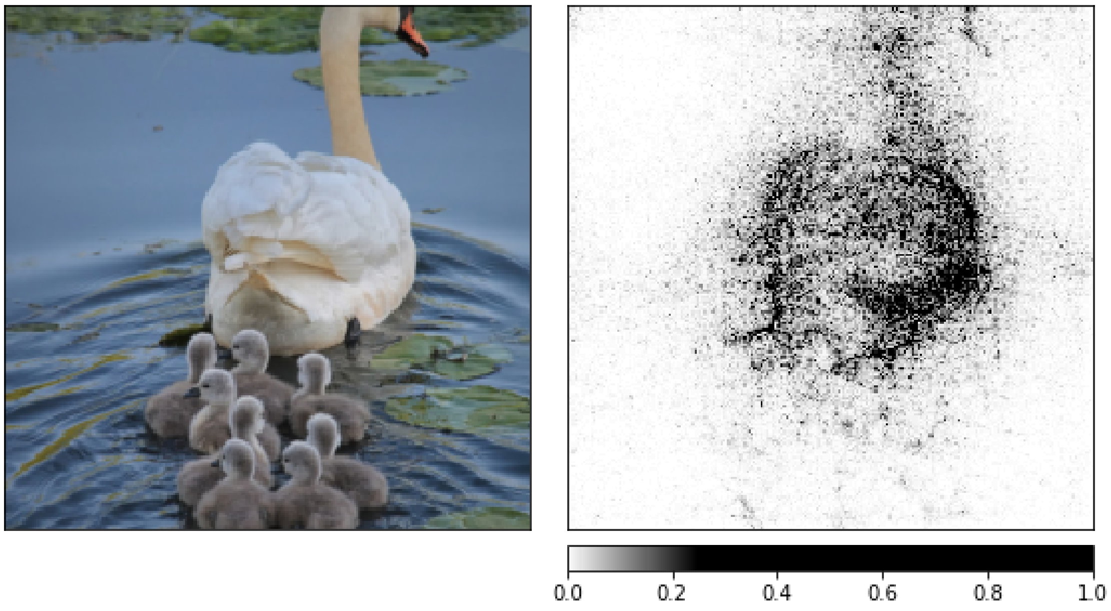

Figure 3.3: Saliency Map - Source: https://captum.ai/tutorials/Resnet_TorchVision_Interpret

Saliency maps are commonly calculated using SHAP [20] or gradient methods. Saliency maps provide a direct and intuitive way to understand which parts of an input data are influencing a model's decision. The main difficulty is the generation of reliable saliency maps.

3.4.1 Vanilla Gradient & Deconv Net

Vanilla Gradient[31] focuses on computing the gradients of the network. A forward pass of an image is generated. The gradients of the class score is computed. Then the gradients are visualized. It is the same process as neural network backpropogation. This method has 2 problems: When ReLU is used and the gradients are negative, then information is lost. Furthermore, in pooling layers, there are no gradients and the information is lost.
Deconv Net [35] takes care of the gradient problem of Pooling and Convolutional Layers.

A standard convolutional network with convolutional layers and fully connected layers and a activation funtion is chosen. An image input image x_i maps to a probability layer y_i. A deconvnet [36] is attached to each of it's layers proving a path back to the image pixels. Unpooling, Rectifying and filtering is done to reconstruct the activity in the layers.

Unpooling: The pooling operation is non-invertible. By recording the original locations of the maxima using a set of switch variables, the reconstructions can be created.

Rectification: Each layer passes through a relu non-linearity. Each layer backwards also passes through a relu non-linearity to keep all signals positive.

Filtering: Filtering is done by using inverted original filters. The devoncnet uses transposed versions of the same filters. This means flipping each filter vertically and horizontally.

3.4.2 Grad-CAM, Guided Grad-CAM and Smooth Gradient

Gradient-weighted Class Activation Map (Grad-CAM) provides visual explanations for CNN decisions. The goal is to understand at which parts of an image a convolutional layer looks for a certain classification.

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

c = class of interest A^k = feature map i,j = width and height dimension Z = average term

GRAD-CAM sets all classes which are not interesting to 0. Then it backpropogates the gradients of the main class. Then the gradients are scaled between [0,1] and displayed. GRAD-CAM does take all convolutional feature maps but the last. This is done, because the last features are responsible for all classes and are therefore not informative. A main problem of GRAD-Cam is that it is very coarse.
Guided Grad-CAM computes Grad-CAM with another method to have a better localization. It's basically multiplied with another method to achieve stable results. GRAD-Cam works as a lenses to focus on specific parts of the pixel-wise attribution map.
SmoothGrad [32] is a smoothing method for other gradient calculation methods. By adding noise to the input image and calculating several input images and taking the gradient, noise should cancel out and be removed.

3.4.3 Top-Down Neural Attention by Excitation Backprop [37]

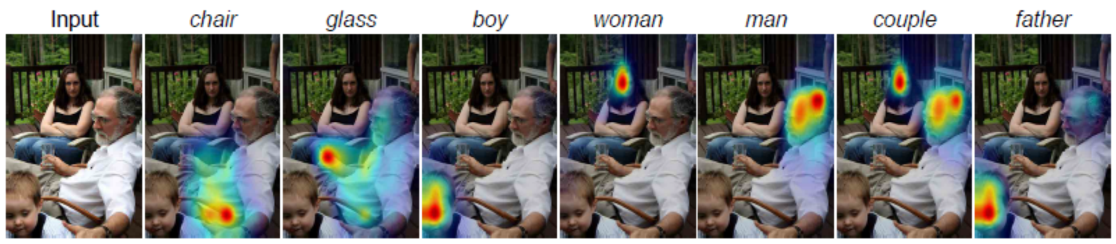

Figure 3.4: CNN classifier's top-down attention map [37]

By calculating the importance of a neuron set and saving it during the computational step, the gradient step can be combined with the importance and localize exact objects. The probabilistic WTA formulation produce well-normalized attention maps that enable direct subtraction.

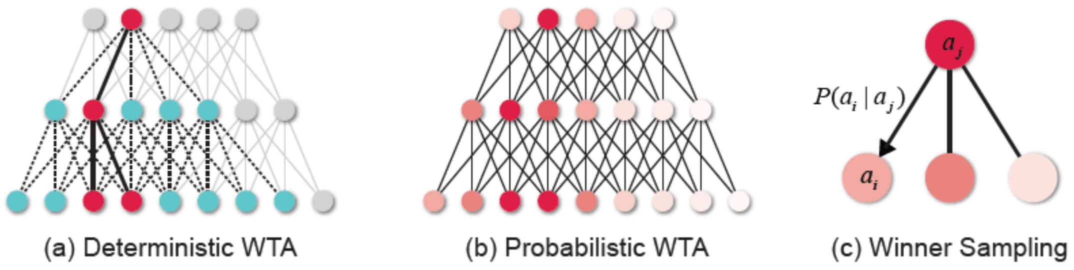

Figure 3.5: Identifying task-relevant neurons in the network. The red shading of a dot indicates its relative likelihood of winning against the other ones in the same layer. [37]

3.4.4 Advantages and Disadvantages of Saliency Maps

The advantage of saliency maps is that they are fast to compute and easy to interpret. The explanations are visual and can immediately be recognized. Furthermore, there are many methods to choose from.
The main disadvantage is that it is unclear whether an explanation is correct or not. In the next chapter possible evaluation methods are explained.

4 Evaluation of post-hoc interpretability methods

Gradient-methods which generate saliency maps are hard to measure. While humans can evaluate the maps by giving a general statement, this can not be applied to thousands of images. Despite many significant recent contributions to saliency maps, the valuable effort of explaning machine learning models face this methodological challenge: the difficulty of assessing the scope and quality of model explanations [1].

4.1 Evaluation metrics, ground truths [13]

Different evaluation methods exist to evaluate saliency methods. Evaluations can be extrinsic [13], involving human evaluations and comparing the results to certain ground truth explanations [37]. Intrinsic methods use computations involving the net itself and the saliency map, without human evaluation or retraining a net. These methods are based on creating a new composite input using the heat map and the original input. Then they are evaluated using the original trained model. E.g.[7] These methods suffer from violating one key assumption in machine learning: the training and evaluation data must come from the same distribution. [15] Without re-training it is not clear if the degrade in performance stems from the distribution shirt or because informative features were removed.

[24] states that comparing the rank correlation between attention weights and feature-additive methods is not appropriate. This means, using agreement as evaluation (using a method as standard to compare to) should not be used. In the paper a low empirical evidence was found, that models are similar. This means that just evaluating similar methods and taking the common results is not applicable. More rigorous defined metrics need be considered.

4.2 Completeness and Soundness

[13] proposes Soundness and Completeness, two concepts which are required for evaluation metrics which involve using a composition of a heat map and the original input.
Soundness is needed: The masked input method proves that a certain part of the image "caused" the networks output.

Completeness means, for any composition of an input image of label a with a well functioning saliency mask the model must still be able to identify the correct label.
$\forall x : f(x \rightarrow a)$
$x = input(a) \odot mask$
Soundness means, for any composition of an input image of label b with a well functioning saliency mask them model must not return a wrong label.
$\nexists x : f(x \rightarrow a)$
$x = input(b) \odot mask$

The AUC metric [26] of the insertion game:
For s = [1,dim(x)] take the top s pixels as per saliency map m and plot the probability f(x,a) given by the model. The top s pixels of x are retained and the remaining pixels are assigned a default value). Return the area under the curve.

The method is α complete on f,x,a if:

$$g_{AUC}(x, a, m) >= \alpha f(x, a)$$

The method is β sound on f,x,a if:

$$g_{AUC}(x, a, m) <= \frac{1}{\beta} \alpha f(x, a)$$

$$\alpha(x, a) = min(\frac{max(g_{AUC}(x, a, n), \epsilon_1)}{f(x, a)}, 1)$$

$$\alpha(x, a) = min(\frac{max(f(x, a), \epsilon_2)}{g_{AUC}(x, a, n)}, 1)$$

Then $\alpha(m) = E_x[min_a(\alpha(x, a))]$
Then $\beta(m) = E_x[min_a(\beta(x, a))]$

If alpha is close to 1: The blocked input predicts correctly.
If beta is close to 1: The blocked input mimics the behavior and is not overconfident.

x: input to the model
a: true label
f: model
m : saliency method.

ϵ = term if x is very small

Older evaluation methods only try to maximize the g_{AUC} curve. This completely ignores if the method is overconfident.

Critics to the method: No human interpretability is included. Furthermore, the masked images do suffer from not having the same distribution as in the original trained maps. This could cause a unwanted drop in performance.

4.3 Perturbation based [30]

A heatmap is an array of pixel-wise scores that indicate which pixels are relevant.

This paper describes a method, where the most important pixels are replaced by randomly sampled values. IT then is measured how the values f(x) the label value goes down. Depending on the method used, the method works superior or worse. If the edges are the most important in a saliency map, then deleting them results in a steep decrease.

This method has high computational costs.

4.4 A benchmark for interpretability methods in deep neural networks [15]

In this paper two methods to estimate the effectiveness of saliency maps are presented. "RemOve And Retrain (ROAR)" and "Keep And Retrain (KAR)".

ROAR proposes a numerical solution to evaluate attribution maps. It is a algorithm which estimates the effectiveness of saliency maps. This is done, by removing supposedly informative features from the input and observing the reaction of the neural network. ROAR is applicable to any visual domain. In Section 5, an example for MNIST and Food-101 is done.

RoaR works as follows:

1. **Selection of the most important pixels from an attribution map**
   An attribution map provides a mapping to the most important pixels. Those pixels are ranked based on their importance. Depending on the algorithm, different values differ on the importance. In the case of Integrated Gradient, the pixels with the highest absolute values are considered the most important.

2. **Replacement of x% of the pixels with the mean value**
   In the Roar algorithm, 0.1, 0.3, 0.5, 0.7 or 0.9 % of the most important pixels are identified and replaced by the mean value. Both the training and the testing data undergo this process.

3. **Retraining the model using the modified dataset**
   To ensure the consistency of the model, retraining is necessary. Afterwards, the model is evaluated using several seed runs on the test set. The same split should be used for all runs. This is crucial, because the training data and the test data must be drawn from the same distribution. Without retraining the model this property is ignored.

4. **Comparison of the attribution map with a random baseline**
   For each training setup, respectively with 0.1, 0.3, 0.5, 0.7 or 0.9 % pixels replaced, a random baseline is considered. The random baseline also replaces the same percentage of pixels. The random baseline is expected to perform worse than a sophisticated method.

5. **Evaluation**
   An attribution method is deemed effective, if it consistently outperforms the random baseline across various setups.

KAR works similar: Instead of removing the most important pixels, the least important pixels are identified. Because KAR performs worse than ROAR, it is not described further.

The conclusion of [15] is that commonly used based estimator, Gradients, Integrated Gradient and Guided BackProp are worse than a random assignment. The effectiveness of Smooth-Grad-Squared and VarGrad was proofed.

4.5 Benchmarking Attribution Methods (BAM) [34]

BAM with relative feature importance explores wether features are important or not. In reality, we do not know how important a feature is. However, we can calculate how important a feature is to model relative to another model. Given the relative feature importance, the metrics compare attributions between pairs of models and pairs of input.

The main idea of BAM is: If you paste a grey square to every training image for all classes, it is expected that this square matters less than the original image region being covered. The same expectation should hold if instead an object (which is not represented by the dataset) is pasted in every image. Any explanations that assign higher attribution to the inserted object than to the original pixels are false positives.

BAM solves the problem of an distribution shift by inserting objects by only inserting objects which don't strongly change the distribution of the image. E.g. an object with mean similar to the mean of the dataset is chosen.

The BAM algorithm works as follows:

1. **BAM dataset construction**
   The BAM dataset is constructed by pasting object pixels from MSCOCO [18] into scene images from MiniPlaces [38]. An object is re-scaled to between 1/3 to 1/2 of a scene image at a randomly chosen location. Resulting images have an object label and a scene label. Either can be used to train a classifier. Every object class appears in every scene class and vice versa. Scenes which contain original BAM objects are not used.

2. **Common features and commonality**
   Common features are defined as a set of pixels with semantic meaning (e.g. looks like a dog) which commonly appear in all examples of one or more classes. For example, a dog which appears in all bamboo forests is less common than a dog which appears in all images of bamboo forests, bedroom and corn field.

3. **2 Classifiers are trained and attribution maps are created**
   An object detection classifier and a scene detection classifier are trained. Objects should be significantly more important than the scene to the object detector than to the scene detector. To verify this intuition, the objects are removed. With this knowledge, attribution maps can be tested by checking if the pixels are assigned noticeably higher attributions.

4. **Relative importance is calculated.**
   With this knowledge, the object pixels should be more important than any other pixels for the detection classifier. Additionally for the scene attribution, the attribution maps should be higher if the object pixels are replaced with the original pixels.

The advantage of BAM over ROAR [15] and other methods is the lower computational cost. No retraining or perturbation is required.

4.6 Sanity Checks for Saliency Maps [1]

In this paper an actionable methodology is proposed, which kinds of explanation a given method can and cannot provide. It shows that visual inspection by humans does not provide good information if the explanation is sensitive to the underlying model and data.

Two instances of the framework are tested:

1. **Model Parameter randomization test**
   The model parameter randomization test compares the output of a salinecy method on a trained model with the output of a randomly initalized untrrained network of the same architecture. The output should differ, otherwise the saliency map is deemed as not helpful.

2. **Data randomization test**
   The data randomization test randomly shuffles the labels of the data and the model is trained on this method. If the saliency maps does not differ to a normally trained model, then the method does not depend on the relationship of the images and labels.

   If any of the 2 hypothesis is failed, then the method can safely be rejected. This is a so called sanity check.

   Extensive experiments on several explanation methods are done across data sets and model architectures.

   On the tested methods, Gradients & GradCam pass the Sanity checks, while Guided Back-Prop & Guided GradCAM fail.

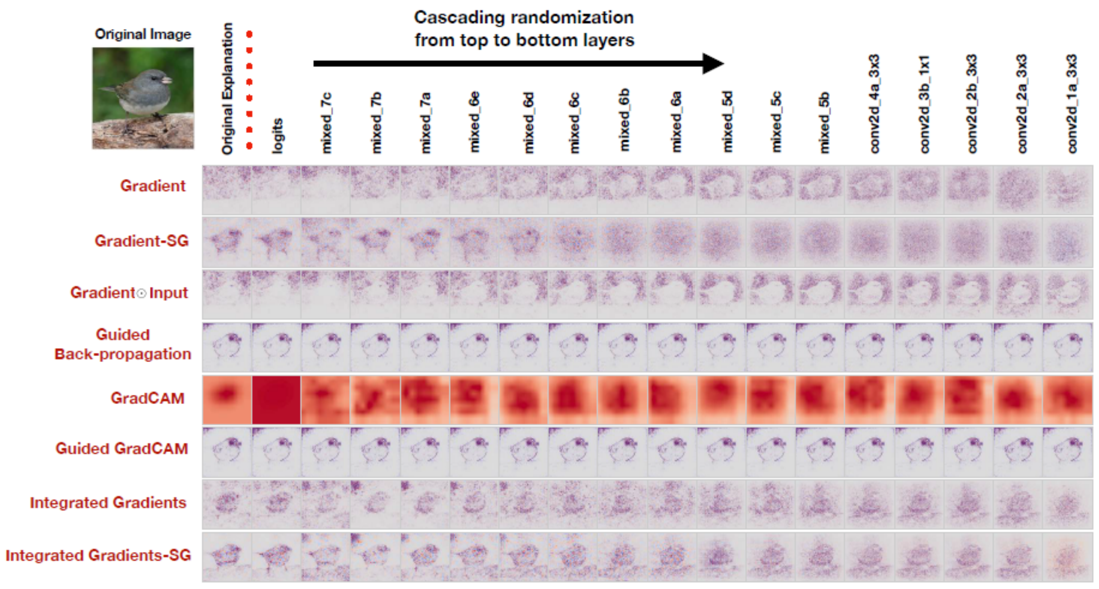

Figure 4.1: Cascading Randomization on Image Net. The figure shows the original explanations. Progression from left to right indicate complete randomization of network weights up to that block inclusive. The last block corresponds to a network with completely reinitialized weights.[1]

4.7 Validating the research

After understanding common machine learning algorithms, one common problem is identified: validating the research [2].

1. Replicability, Reliability and Trust. Evaluation methods need to be replicable to some degree. Predictions and finding should be robust to sensitivity tests like small changes in the data or the model, out-of-sample prediction tests and consistency with domain knowledge [21].

2. Why do machine learning interpretations fail?

a. The machine learning model could be a poor fit for the model and any resulting interpretation approach would poorly reflect the signal in the data.

b. The interpretation approach could be a poor fit for the model. Especially when fitting a second model to generate the interpretation, like LIME.

c. There can be a mismatch to the employed interpretation technique and the desired discovery task. For example trying to find out why a model predicts wrongly can't be understood using a poor saliency map.

For prediction tasks well established techniques for validation were developed: Using a split for train, test and validation data. Using this the predictive model should generalize well to new, unseen data. Knowing this, can't one simply employ a training and test set to validate interpretations? [2] This prospect is way more complicated for interpretations. There exists no fair metric that determines which interpretation is the best. Using human evaluation or domain experts could also be an option, but this also struggles from having no good evaluation method. And even experts do not have a common consensus for each dataset.

Validating discoveries from interpretable machine learning is still a challenge and more research is needed to promote trustworthy and reliable data science.

4.8 Research findings: Which evaluation method to use, how to control the result?

Several saliency maps and possible evaluation methods were introduced. Now which one should a data scientist use?
The unsatisfactory answer is: There is no clear right and wrong. Depending on the model and dataset used, different evaluation methods give different results. In [1] and [15] various methods were seen as unsuitable.

Using a SG - GRAD or a VAR-GRAD approach results in a good results for ROAR and SG-Grad also has good results for Sanity Check for saliency maps. Excitation Backprop has not been analyzed yet with this methods, but could also be trustworthy. After all, there is not enough reliable research to say one method is clearly better than another. This can seen as recommendation, but not as clear instruction.

It remains difficult to evaluate visual explanations and there is no clear truth yet, on which evaluation method is the best. More research is needed to be able to give clear guidelines.

5 Reconstructing ROAR

5.1 Scientific Motivation and Goal

In RoaR[15] the paper does not list the standard deviation of the trained nets. We expect to validate the results by achieving similar results.

As training 25 image nets requires high computational power we do not have right now, we limited our research to evaluating food-101 [5] using only 2 interpretation methods and comparing it to the baseline.

Additionally, I also add a proof-of-concept evaluation using the MNIST dataset.

5.1.1 Project Setup MNIST

The MNIST dataset [8] was chosen as proof-of-concept. Because calculating the gradients and judging if a mask is meaningful is easier here, it should give a reliable baseline for the results. Only integrated gradients is tested, as is it only an example.

A. Training the model

The model used in this report is a simple convolution neural network architecture consisting of two convolution layers, two pooling layers, and three linear layers. The model was trained for ten epochs with a learning rate of 1e-3 and a SGD optimizer with a momentum term of 0.9. The batch size was 64. Across five separate runs, the model achieved an average accuracy of 97.4% with a standard deviation of 0.4% on the test data.

B. Dataset preparation and Splitting

The MNIST data set consists of 10 classes and 70.000 different images. It was chosen, because it is easy to evaluate the effectiveness of integrated gradients and ROAR: a) The importance of the pixels is clear from a human perspective. b) The model required to achieve a solid performance is very modest and speeds up the retraining. c) The information loss in retraining is clearly visible in the pictures. Furthermore, it is a small, usable data set. The images are 28x28 pixels big and one pixel has a value ranging from [0,1] in gray-scale. The data set used in this report was normalized using the mean and standard deviation, and was split using the PyTorch library's default splitting method (60.000 images for the training data, 10.000 images for the test data). The training data was shuffled randomly using the inbuilt functions. The Cross entropy loss was selected for training runs. The same data set split was used for all training runs to ensure consistency in the results.

C. Example of the new dataset

In 5.1 the difference between Integrated Gradients and the Random Baseline is visible. In the first row the random baseline is visible, in the second row Integrated Gradients is shown. [0.1,0.3,0.5,0.7,0.9%] of the pixels are blacked out (left to right).

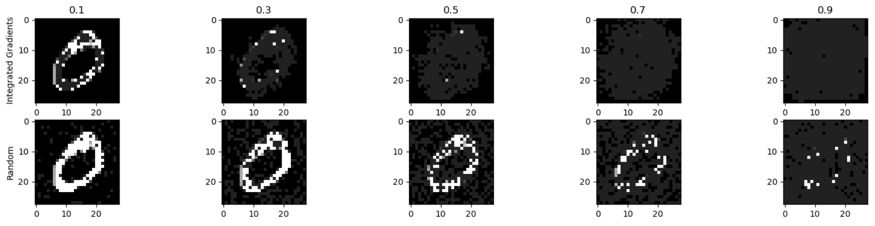

Figure 5.1: Integrated Gradients and Random Baseline Comparison

Judging by human intuition, Integrated Gradients is appearing to perform better over all thresholds. The model is expected to perform worse following retraining.

D. Retraining upon the new datasets

To ensure that the model learns consistently, the same split and learning parameters were utilized as in the standard case to retrain the model on the new data set. An untrained network was created and then retrained for each training run. In order to validate the results and minimize the impact of random seed initialization, the training was performed with 5 different seeds and 5 models were created. In the Appendix, the exact results can be found.

E. Advantages and Disadvantages of the setup

Because of the small size of the data set and the limited amount of classes, training a model is fast and effective. One disadvantage is the limited generalization of this setup for other datasets and models. The majority of use cases primarily involve working with images in the RGB color space, utilizing deep learning models that are often more complex, and dealing with a larger number of distinct classes. Furthermore, the pixels itself are blurred on their importance. In RGB images 3 channels are combined, which make attribution maps more difficult to compute and understand.

F. Results

Integrated Gradient is performing better than the random baseline.

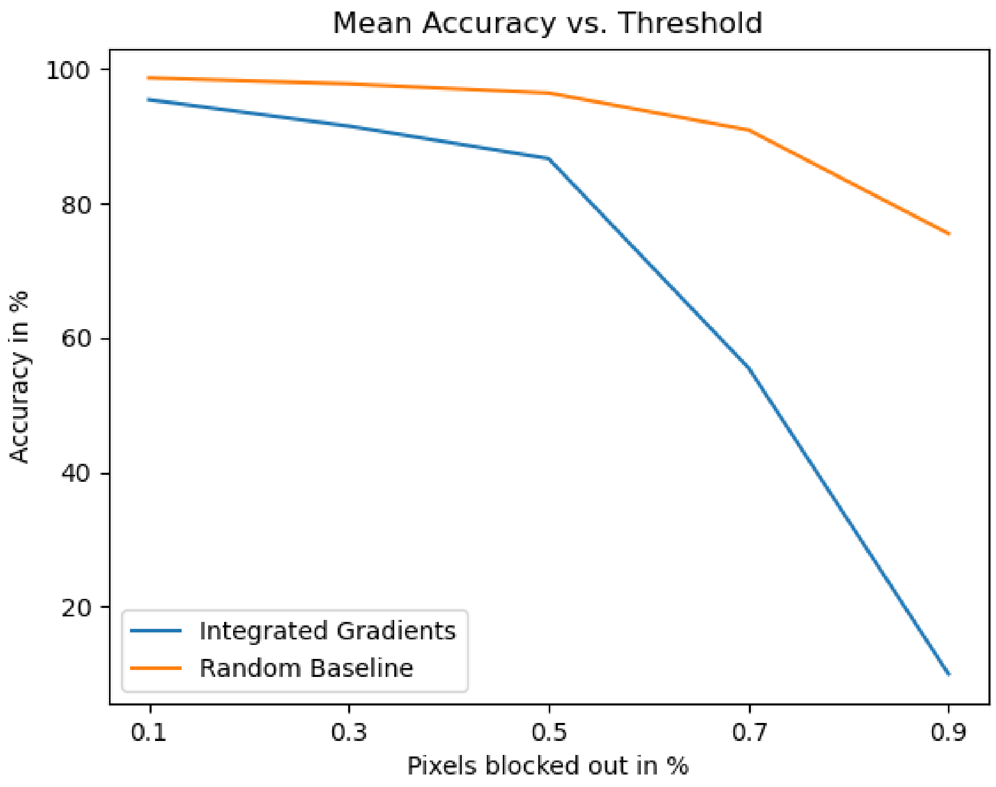

Figure 5.2: Accuracy: Random Baseline vs Integrated Gradient

5.1.2 Project Setup Food-101

The MNIST dataset [8] was chosen as proof-of-concept. Because calculating the gradients and judging if a mask is meaningful is easier here, it should give a reliable baseline for the results. Only integrated gradients is tested, as is it only an example.

A. Training the model

The model used in this report is ResNet50 [14]. The Food-101 [5] dataset was chosen to replicate. Learning rate was adjusted to 0.175, batchsize of 64. Using an SGD optimizer with momentum =0.9 and weight decay = 0.0001 and a scheduler at epoch 30 with gamma=0.1, an accuracy of averagely 70.5% with standard deviation of 0.01 was achieved over 5 separate training runs on the test score. In total 31 epochs were done.

In the original paper an accuracy of 84.54% was achieved on the Food-101 dataset. A smaller accuracy was achieved, as the training was noticeably shorter because only a NVIDIA GTX 1080 was available.

The data was resized with center-crop to make all images 3x224x224. Afterwards they were normalized. No data augmentation like flips and mirroring was applied. The standard split 75:25 was applied. The altered datasets were calculated with all using the same model.

B. Results

5.1.3 Interpretation of the results

The results from the original paper were confirmed. Integrated Gradients preforms as good as a random baseline for the Food-101 setup. While the MNIST result shows that Integrated Gradient indeed seems to find important pixels, in the Food-101 setup Integrated Gradient fails. Therefore we can say that Integrated Gradient seems to fail for bigger networks.

Bibliography

[1] Julius Adebayo et al. *Sanity Checks for Saliency Maps*. 2020. arXiv: 1810.03292 [cs.CV].

[2] Genevera I. Allen, Luqin Gan, and Lili Zheng. *Interpretable Machine Learning for Discovery: Statistical Challenges & Opportunities*. 2023. arXiv: 2308.01475 [stat.ML].

[3] P. Berkhin. "A Survey of Clustering Data Mining Techniques". In: *Grouping Multidimensional Data* (2006), pp. 25–71. URL: http://dx.doi.org/10.1007/3-540-28349-8_2.

[4] H. Beyer. "Tukey, John W.: Exploratory Data Analysis. Addison-Wesley Publishing Company Reading, Mass. — Menlo Park, Cal., London, Amsterdam, Don Mills, Ontario, Sydney 1977, XVI, 688 S." In: *Biometrical Journal* 23.4 (1981), pp. 413–414. DOI: https://doi.org/10.1002/bimj.4710230408. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/bimj.4710230408. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/bimj.4710230408.

[5] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. "Food-101 – Mining Discriminative Components with Random Forests". In: *European Conference on Computer Vision*. 2014.

[6] Diogo V. Carvalho, Eduardo M. Pereira, and Jaime S. Cardoso. "Machine Learning Interpretability: A Survey on Methods and Metrics". In: *Electronics* 8.8 (2019). ISSN: 2079-9292. URL: https://www.mdpi.com/2079-9292/8/8/832.

[7] Piotr Dabkowski and Yarin Gal. *Real Time Image Saliency for Black Box Classifiers*. 2017. arXiv: 1705.07857 [stat.ML].

[8] Li Deng. "The mnist database of handwritten digit images for machine learning research". In: *IEEE Signal Processing Magazine* 29.6 (2012), pp. 141–142.

[9] Finale Doshi-Velez and Been Kim. *Towards A Rigorous Science of Interpretable Machine Learning*. 2017. arXiv: 1702.08608 [stat.ML].

[10] Mengnan Du, Ninghao Liu, and Xia Hu. *Techniques for Interpretable Machine Learning*. 2019. arXiv: 1808.00033 [cs.LG].

[11] Jerome H. Friedman, Trevor Hastie, and Rob Tibshirani. "Regularization Paths for Generalized Linear Models via Coordinate Descent". In: *Journal of Statistical Software* 33.1 (2010), pp. 1–22. DOI: 10.18637/jss.v033.i01. URL: https://www.jstatsoft.org/index.php/jss/article/view/v033i01.

[12] Riccardo Guidotti et al. *A Survey Of Methods For Explaining Black Box Models*. 2018. arXiv: 1802.01933 [cs.CY].

[13] Arushi Gupta et al. *New Definitions and Evaluations for Saliency Methods: Staying Intrinsic, Complete and Sound*. 2022. arXiv: 2211.02912 [stat.ML].

[14] Kaiming He et al. *Deep Residual Learning for Image Recognition*. 2015. arXiv: 1512.03385 [cs.CV].

[15] Sara Hooker et al. *A Benchmark for Interpretability Methods in Deep Neural Networks*. 2019. arXiv: 1806.10758 [cs.LG].

[16] Pang Wei Koh and Percy Liang. *Understanding Black-box Predictions via Influence Functions*. 2020. arXiv: 1703.04730 [stat.ML].

[17] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Commun. ACM* 60.6 (2017), pp. 84–90.

[18] Tsung-Yi Lin et al. *Microsoft COCO: Common Objects in Context*. 2015. arXiv: 1405.0312 [cs.CV].

[19] Zachary C. Lipton. *The Mythos of Model Interpretability*. 2017. arXiv: 1606.03490 [cs.LG].

[20] Scott Lundberg and Su-In Lee. *A Unified Approach to Interpreting Model Predictions*. 2017. arXiv: 1705.07874 [cs.AI].

[21] Xiao-Li Meng. "Reproducibility, Replicability, and Reliability". In: *Harvard Data Science Review* 2.4 (2020). https://hdsr.mitpress.mit.edu/pub/hn51kn68.

[22] Christoph Molnar. *Interpretable Machine Learning. A Guide for Making Black Box Models Explainable*. 2nd ed. 2022. URL: https://christophm.github.io/interpretable-ml-book.

[23] W. James Murdoch et al. "Definitions, methods, and applications in interpretable machine learning". In: *Proceedings of the National Academy of Sciences* 116.44 (2019), pp. 22071–22080. DOI: 10.1073/pnas.1900654116. eprint: https://www.pnas.org/doi/pdf/10.1073/pnas.1900654116. URL: https://www.pnas.org/doi/abs/10.1073/pnas.1900654116.

[24] Michael Neely et al. *Order in the Court: Explainable AI Methods Prone to Disagreement*. 2021. arXiv: 2105.03287 [cs.LG].

[25] Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. "Feature Visualization". In: *Distill* (2017). https://distill.pub/2017/feature-visualization. DOI: 10.23915/distill.00007.

[26] Vitali Petsiuk, Abir Das, and Kate Saenko. *RISE: Randomized Input Sampling for Explanation of Black-box Models*. 2018. arXiv: 1806.07421 [cs.CV].

[27] Ribana Roscher et al. "Explainable Machine Learning for Scientific Insights and Discoveries". In: *IEEE Access* 8 (2020), pp. 42200–42216. DOI: 10.1109/ACCESS.2020.2976199.

[28] Cynthia Rudin. *Stop Explaining Black Box Machine Learning Models for High Stakes Decisions and Use Interpretable Models Instead*. 2019. arXiv: 1811.10154 [stat.ML].

[29] Wojciech Samek and Klaus-Robert Müller. "Towards Explainable Artificial Intelligence". In: *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. Springer International Publishing, 2019, pp. 5–22. DOI: 10.1007/978-3-030-28954-6_1. URL: https://doi.org/10.1007%2F978-3-030-28954-6_1.

[30] Wojciech Samek et al. "Evaluating the Visualization of What a Deep Neural Network Has Learned". In: *IEEE Transactions on Neural Networks and Learning Systems* 28.11 (2017), pp. 2660–2673. DOI: 10.1109/TNNLS.2016.2599820.

[31] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. *Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps*. 2014. arXiv: 1312.6034 [cs.CV].

[32] Daniel Smilkov et al. *SmoothGrad: removing noise by adding noise*. 2017. arXiv: 1706.03825 [cs.LG].

[33] Feiyu Xu et al. "Explainable AI: A Brief Survey on History, Research Areas, Approaches and Challenges". In: Sept. 2019, pp. 563–574. ISBN: 978-3-030-32235-9. DOI: 10.1007/978-3-030-32236-6_51.

[34] Mengjiao Yang and Been Kim. *Benchmarking Attribution Methods with Relative Feature Importance*. 2019. arXiv: 1907.09701 [cs.LG].

[35] Matthew D Zeiler and Rob Fergus. *Visualizing and Understanding Convolutional Networks*. 2013. arXiv: 1311.2901 [cs.CV].

[36] Matthew D. Zeiler, Graham W. Taylor, and Rob Fergus. "Adaptive deconvolutional networks for mid and high level feature learning". In: *2011 International Conference on Computer Vision* (2011), pp. 2018–2025. URL: https://api.semanticscholar.org/CorpusID:975170.

[37] Jianming Zhang et al. "Top-Down Neural Attention by Excitation Backprop". In: *International Journal of Computer Vision* 126 (Oct. 2018). DOI: 10.1007/s11263-017-1059-x.

[38] Bolei Zhou et al. *Places: An Image Database for Deep Scene Understanding*. 2016. arXiv: 1610.02055 [cs.CV].