EVALUATING FEATURE ATTRIBUTION METHODS

Benedict Gattas

June 2020

School of Information Technology and Electrical Engineering,
The University of Queensland

Submitted for the degree of
Bachelor of Engineering
in the field of Software Engineering.

Benedict Gattas
benedict.gattas@uqconnect.edu.au

June 18, 2020

Prof Amin Abbosh
Acting Head of School
School of Information Technology and Electrical Engineering
The University of Queensland
St Lucia, Q 4072

Dear Professor Abbosh,

In accordance with the requirements of the degree of Bachelor of Engineering (Honours) in the division of Software Engineering, I present the following thesis entitled "Evaluating Feature Attribution Methods". This work was performed under the supervision of Dr Alina Bialkowski.

I declare that the work submitted in this thesis is my own, except as acknowledged in the text and footnotes, and has not been previously submitted for a degree at The University of Queensland or any other institution.

Yours sincerely,

Benedict Gattas.

Acknowledgments

I wish to acknowledge the direction and support provided by my supervisor Dr Alina Bialkowski over the course of this project. Feedback after each milestone was invaluable and tips on different approaches helped provide guidance at key junctions. This ranged from helping narrow the scope at the start of the project, suggestions after each deliverable to help improve the next one, and regular appraisal of the direction I was heading throughout the project. It would also be remiss of me not to offer congratulations on baby Emilie's birth! To my family, I'd like to thank my parents and my siblings Joe, Nick and Susannah for their endless care and support.

Abstract

This document is a skeleton thesis for 4th-year students. The printable versions show the structure of a typical thesis with some notes on the content and purpose of each part. The notes are meant to be informative but not necessarily illustrative; for example, this paragraph is not really an abstract, because it contains information not found elsewhere in the document. The LaTeX 2_ε source file (`skel.tex`) contains some non-printing comments giving additional information for students who wish to typeset their theses in LaTeX. You can download the source, edit out the unwanted material, insert your own frontmatter and bibliographic entries, and in-line or `\include{}` your own chapter files. Of course the content of a particular thesis will influence the form to a large extent. Hence this document should not be seen as an attempt to force every thesis into the same mold. If in doubt about the structure of your thesis, seek advice from your supervisor.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

"State of the art" machine learning models now regularly achieve above expert-level performance in fields as diverse as medical imaging and language translation. However, lack of interpretability prevents their adoption in many of the fields that would benefit the most from them. These domains are often ones where the decisions made have a tangible impact on people's lives. Decision-makers in those domains are unlikely to use the predictions made by a highly accurate but opaque model, because of their requirement for trust and accountability.

The requirement for explainability is not in order to replace human experts, but to understand contradictions between expert and algorithm. For example, a radiologist might disagree with a diagnosis made by a model trained to predict pneumonia from chest X-rays, and attributing the error to a factor that one or the other relied upon would be helpful. Was the model relying on some unrelated part of the scan (a spurious feature) or was the radiologist failing to pick up on a subtle pneumonia differential? The former was observed in practice after applying an explanation technique to a convolutional neural network (CNN) trained on such data [1].

In the many domains that require trust, explanations for model predictions are as important as low rates of incorrect predictions. This is the counter-argument to the notion that a highly accurate model need not be interpretable to be effective: the premise of effectiveness requires a level of *trust* that a model relies on unbiased data and non-spurious features, two guarantees that are not at all provided by an objective function that seeks only to minimise prediction error. With poor visibility into the factors that a model relies upon, machine learning researchers tend to use model performance metrics as the basis for arguing a new model architecture is superior. This disconnect between performance in the sense of test set accuracy and performance in the sense of accountability (lack of bias or spurious features) and reliability (sensical behaviour) is not ideal.

Interpretability in a Facebook algorithm recommending product categories, for example, might not be seen as important as interpretability in a cancer diagnosis model, though the possibility of unethical model behaviour from reliance on biased data is as tangible in both domains. One study of 200 sentiment analysis classifiers found several to have significant race and gender bias [2]. The consequences of errors can certainly be higher in some domains however - a poor Netflix recommendation is not as disastrous as a naive algorithm used in government decision-making, such as the "robo-debt" scheme recently employed by the Australian Government [3].

Approaches to Interpretability

There is fortunately an active literature aimed at addressing this 'black-box' critique in machine learning. The top-level distinction among approaches is to either use inherently intepretable models to achieve explainability, or take complex, black-box models and find techniques to isolate and explain a piece of their complexity, such as an individual prediction.

The first approach includes model families with low complexity like linear/logistic regression, decision trees, k-nearest neighbours and Naive Bayes. Within these families are both parametric and non-parametric techniques, which demonstrates that lack of interpretability is more related to model complexity than a particular type of formulation - this empirically observed trade-off between accuracy and interpretability is discussed further in the next section[1]. Since many of these models are often too simplistic for obtaining competitive performance, the motivation to attack the 'black-box' critique from this angle is quite low. Instead, methods to introduce interpretability to modern, high-performance models are a more studied and popular approach to take in the literature, as in the second approach.

The second approach includes "feature attribution" or "feature importance" methods, which compute a weight score for each feature in the input space to measure its contribution to an output class. For example, a CNN classifier predicting "tree" would be expected to rely heavily on green pixels of leaves. This class includes model-specific techniques for neural network architectures, like those based on activations in a hidden layer, and model-agnostic techniques that are compatible with any model family. Within both sub-classes are a variety of techniques, with varying levels of model agnosticity and task compatibility. For example, some methods are

[1]There is a view by some researchers ([4]) that in many domains the accuracy vs interpretability trade-off does not exist, and thus there is a responsibility to use equally effective, inherently interpretable models for high-stakes decisions where those are available.

designed solely for CNNs and are therefore mainly suited for image classification or related tasks.

Importantly there are both global feature attribution methods, which calculate each feature's contribution to a model at large, as well as local methods, which attempt to explain a single prediction. This project has focused on the latter. Local methods are dominant in the literature for modern architectures - when the dimensionality of the data is high, such as in visual data, or when the number of parameters is too high to make conclusions about global model behaviour (as in most modern architectures) this tends to be the only effective approach to interpretability. An author of one local, model-agnostic method notes that understanding these models globally "would require understanding of the underlying prediction function at all locations of the input space" [5].

Accuracy vs Interpretability

As deep learning and other state-of-the-art model families proliferate in their typical number of parameters, global behaviour has become even less explainable. Researchers maintain some intuitions about the impact of architectural design decisions, though not on predictive behaviour. For example, filters within a CNN model have been shown to act as 'object detectors' of patterns, shapes and other connected regions [6], though these per-layer intuitions don't explain how a network of dozens of layers will determine a husky from a wolf (a recent approach in the literature, however, has looked at abstracting model behaviour into 'concept' vector encodings to capture model behaviour across filters and layers (Net2Vec [7], TCAV [8])).

Feature attribution methods can therefore re-introduce transparency into complex, non-linear models and highlight predictive biases in the context of individual predictions. They also can reveal unexpected features involved in a prediction, such as the spurious features mentioned in the previous X-ray data example, or bugs that could lead to exploitation of adversarial examples [9]. Note these methods do not seek to add causal interpretability to the models they are applied to, only to isolate and highlight a piece of complexity in a way that might make sense to an expert reviewing the explanation. This does not make them shallow - the benefit of the 'post-training' approach is that model designers have more flexibility in their choice of models, and fewer restrictive assumptions about model complexity are made[2]. More model-agnostic methods, with the least restrictive requirements, are not well understood in context with model-specific ones in terms of this accuracy

[2]The counter-argument made by those who argue for the inherently interpretable model approach is that there is no guarantee these explanations are faithful to the model, and that they extend the authority of the black box instead of making it a "glass box" [4]. This is revisited in the **Discussion**.

and interpretability trade-off.

Existing Literature

Some comparisons of feature attribution methods do exist, though typically either in a qualitative context, as a pairwise comparison, **cite** or within a single class of methods. They are not normally compared on speed/performance or adoptability in terms of task/model compatibility. They are however haphazardly compared on explanation quality, which is a difficult criteria to design. Lack of evaluation is partly due to differences in method formulation and therefore representation, but also this difficulty in finding objective proxy metrics for explanation quality.

1.2 Project Overview

This project has sought to evaluate a panel of feature attribution methods representative of different approaches to the interpretability problem. The aim was to highlight their relative strengths and weaknesses and thereby increase the understanding of the benefits of one method's approach over another.

Two other key contributions made over the course of the project have been a quantitative evaluation framework for explanation quality in the image classification context, and the development of a software package to collect image data explanations for multiple underlying methods at scale.

Goals

The aims of this project have been to:

1. **Examine and evaluate** a panel of feature attribution methods for their use cases and performance, using proxy metrics of explanation quality supported by analysis.

2. **Develop** an attribution software package for testing methods at scale with modular support for different models, making it easier for researchers to collect explanations and build more adoptable models.

Project Scope

Section 1.1 introduced a broad motivation for interpretability, though this project has focused specifically on image classification for two main reasons.

Firstly, many interpretability techniques from before the deep learning era have been studied in this domain, and many deep learning specific methods continue to be developed and tested in this domain on modern CNN architectures. The natural 'visual' aspect of image data explanations has also made computer vision a dominant venue for interpretability research, with important applications such as medical imaging.

Secondly, well-annotated datasets and pre-trained, 'off the shelf' models are more easily acquired in this domain. This availability allowed for richer evaluation metrics and the removal of model training as a project requirement.

A more detailed scope is provided at the beginning of the Methodology section, including a description of the specific datasets, models, and feature attribution methods used.

Report Overview

Chapter 2 examines the available feature attribution methods and existing approaches to method evaluation. Chapter 3 breaks down the project's methodology in terms of particular milestones, including the software and evaluation metrics that were designed. Chapter 4 lists evaluation results from quantitative and qualitative standpoints. Chapter 5 provides a discussion on the project's contribution and the limitations encountered, and finally Chapter 6 provides conclusions and recommendations for future work.

Chapter 2

Related Work

2.1 Scope of Research

In Chapter 1 ("Approaches to Interpretability") a brief overview of feature attribution methods was provided with reference to a distinction between model-specific and model-agnostic methods. This is a common distinction in the literature and was also used to guide research in this project. The panel of methods chosen for evaluation ultimately consisted of a balanced selection from both approaches.

A major difficulty of this project was distilling the broad literature on these methods however. For the model-specific (neural network) family, different angles are commonly taken to calculate feature "relevance" or importance. These include generally:

- **Backpropagation-based** methods or 'importance signal' projections (such as activations in a hidden layer)
- **Gradient-based** methods, saliency maps and output sensitivity methods
- **Perturbation-based** methods and input occlusion techniques

This categorisation is based on two recent papers that make similar categorisations of explanation approaches [10] [11]. In this chapter a broad selection of methods based on traction over time, current popularity and representativeness of approach are described, though the reader should note there are more methods under each of those three than have been described here.

For model-agnostic methods, perturbation-based and other miscellaneous approaches are considered. Again the selection was based on traction and literature popularity.

First reviewed are traditional, visual approaches to interpretability to provide context to the task of feature attribution. After the exploration of feature attribution methods, a review of existing evaluation metrics and comparison studies is provided.

Terminology

All methods are variously referred to as *attribution* methods in this project for any projection on the input space that highlights relevant features. Adebayo et al. (2018) instead refer to the broad category of "[...] visualisation and attribution methods aimed at interpreting trained models" as *saliency* methods, particularly in the context of image data [12]. Including by those authors, a *saliency map* is widely used as a catch-all term to refer to input space projections (individual explanations) in the context of interpreting deep neural networks for image data. However they also refer to a specific gradient-based method (Section 2.3.2).

Consensus on terminology is relatively lacking. Some researchers ([11]) describe attribution methods as a subclass of explanation techniques where contribution scores are specifically calculated for each input feature (i.e. excluding higher level 'patterns' which cause neuron activations, as in Zeiler & Fergus (2013) ([13]), (Section **X**) or the back-propagation class generally).

In summary, attribution methods is used here as a general term, but can refer specifically to contribution 'calculators', but saliency methods and saliency maps are also widely used in a general sense for image data.

2.2 Traditional Approaches

2.2.1 Feature Projection

Visualisation tools for high-dimensional data are a popular way to gauge insight into expected model behaviour. These pre-learning, exploratory data analysis techniques include mathematical reductions like PCA and probabilistic techniques like t-SNE, projecting high-dimensional examples that are 'similar' into a visualisable 2D or 3D space [14] (Figure 2.1).

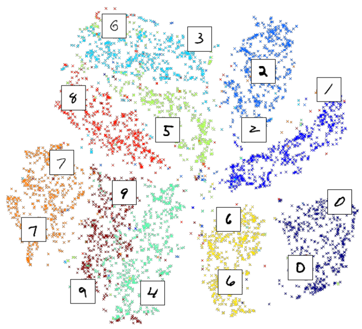

Figure 2.1: 2D embedding of 70,000 handwritten digits (0-9) from MNIST [15].

Other methods of clustering and dimensionality reduction are also widely used for interpreting data, and although useful for gaining an intuition on relationships between features, they are not suited for explaining model behaviour as they examine only the input space itself.

2.2.2 Partial Dependence Plots

A partial dependence plot (PDP) is a tool to demonstrate the marginal effect of one or two features on a prediction outcome. It was proposed by Friedman in 2001 to interpret and visualise the features that the proposed gradient boosting machine relied upon (though it is limited to 1 or 2 input features such that it can be displayed) [16]. A partial dependence function $\widehat{f_{xs}}$ can be calculated for some desired set of features S, by marginalising the model output over the set of 'complement' features C (all other features):

$$\widehat{f_{xs}} = \int \widehat{f}(x_s, x_c) dP(x_c) \tag{2.1}$$

It can be approximated with a Monte Carlo method. Friedman believed in 2001 that these might be used to help interpret "any black box prediction method" such as NN and SVM architectures, and that, "[...] when there are a large number of predictor variables, it is very useful to have a measure of relevance to reduce the potentially large number of variables to be considered" [16]. The mentioned relevance measure was defined only in the context of the decision trees which constituted the paper's gradient boosting machine. Certainly, PDPs are suited for the low-dimensional feature spaces that were imagined in the pre-deep learning era, and are less suitable for high-dimensional input spaces such as in image classification. They are also restricted by an unrealistic assumption of independence among features.

2.3 Model-Specific Methods

Deep learning's reputation for lack of transparency has led to many attempts to explain the predictions of complex NN architectures. This section examines representative attribution methods from the backpropagation-based, gradient-based and perturbation-based approaches overviewed in Section 2.1, with some emphasis on those developed in the context of CNNs (i.e. image data).

2.3.1 Backpropagation-Based

Methods in this class try to isolate an internal model signal, such as neuron activations in a target hidden layer, and map these signals back into the input pixel space. Zeiler & Fergus (2013) introduced the motivation for signal backpropagation as "[...] showing what input pattern originally caused a given activation in the feature maps" [13].

Visualising Activations - DeconvNet, Guided Backpropagation

Visualisation of per-layer activations is one approach to explain inner model behaviour. It is different from other feature attribution approaches in that it seeks to visualise 'learned features' instead of the contributions of input space features. DeconvNet and Guided Backpropagation are two popular examples of the approach and are briefly described as two forerunners of the backpropagation (and gradient) approach.

Deconvolutional networks (DeconvNet) generate backwards projections of neuron activations, by reversing individual activations during a 'deconvolution' backwards-pass [13]. The procedure can be summarised as passing feature map output activations as input into 'deconv' layers, iteratively reversing activations and reconstructing input signals until the input pixel space is reached.

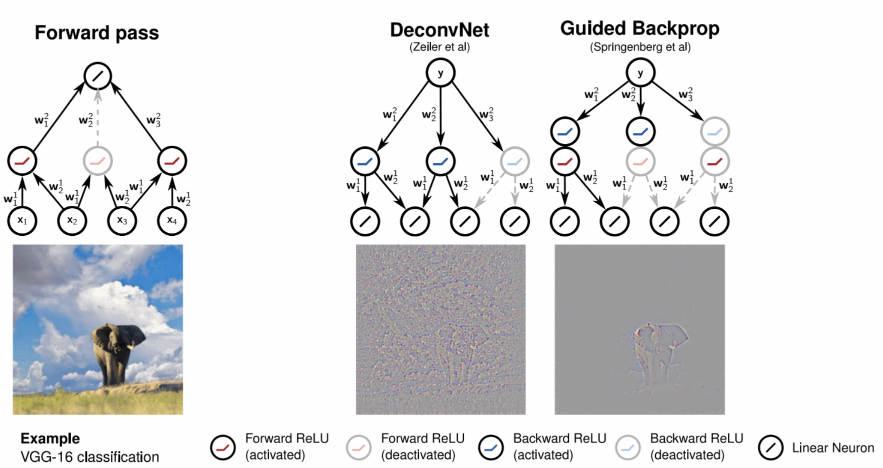

Figure 2.2: (Adapted from [11]) Illustration of DeconvNet and Guided Backprop.

Since it can target one layer's activation at a time, the method is useful for understanding the hierarchical learned features that CNNs generate over multiple layers. Its limitation is that during the backwards pass, it ignores any negative inputs to ReLu activations that were zero'd out in the forward pass (deactivated activations in Figure 2.2).

Guided Backpropagation was an enhancement by Sprigenberg et al. (2014) that added an additional signal at each step by zero'ing the importance signal if it was a negative activation in the forward pass phase *or* negative in the backwards pass (the two intermediate signals in Figure 2.2 right) [17]. This stopped negative gradients in lower layers from decreasing the activation of the higher layer units which were the target, and this leads to sharper explanations than those created by DeconvNet [17].

DeepLIFT

DeepLIFT (Deep Learning of Important FeaTures) [10] was created out of the motivation that the zero'ing of negative gradients by DeconvNet and Guided Backpropagation meant that neither are able to highlight inputs that contribute negatively to an output. In some sense they are therefore missing half the story of feature contribution / neuron activation. DeepLIFT's authors (Shrikumar et al. (2017)) also wanted to overcome the unaddressed saturation problem, which is that relevant neurons that contribute to a saturated output activation would not individually change the output if they were turned off (as might be tried in perturbation approaches).

DeepLIFT's innovation over previous backpropagation and gradient-based methods was to realise that where this problem existed, a single gradient value of an output with respect to an input value did not adequately or necessarily capture input contributions to an output. The authors' proposal was to find input contributions by instead calculating the absolute change in output with respect to a neuron's 'reference' activation.

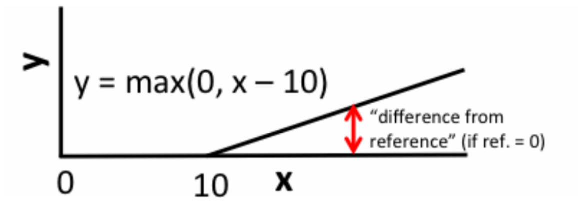

Figure 2.3: (From [10]) Difference-from-reference attributions can avoid bias terms.

Finding reference activations is an implementation difficulty. Generally, the authors propose all-zero input as a baseline (a black square for image data), and a better reference to be the average input over a background data sample [10].

Like other backpropagation methods, DeepLIFT is extremely fast to calculate as it requires only a single backwards pass to propagate an importance signal back into the input space. The authors also provide different formulations for practical implementation, and a relatively high-level public implementation [18]. The concept itself is also compatible with any NN architecture or application, unlike for example DeconvNets and Guided Backpropagation, designed for ReLu activation functions, and GradCAM (Section **X**) which was designed for CNNs.

Other Backpropagation Methods

DeepLIFT is one of several modern methods to decompose a network's prediction onto input features using activation backpropagation. For brevity its main competitors, Layerwise Relevance Propagation [19] and DeepTaylorDecomposition [20], are omitted here but mentioned for the reader's reference.

2.3.2 Gradient-Based

These methods aim to explain a class output in terms of sensitivity in the input space by relying on a gradient function of the output[1]. The goal is to find the input features that make that prediction more or less confident: for example, for an output class of 'tree' they seek to answer "what makes a tree more/less a tree?".

Saliency Maps

An early formulation of a local explanation method was provided by Baehrens et al. (2010) for any nonlinear classification algorithm (though developed in the context of Bayesian classification) [21]. The local explanation gradient vectors that this paper devised were based on class probability gradients, characterising how much a data point has to be moved for a predicted label to change.

Simonyan et al. (2014) later applied a similar idea to CNNs to create 'class saliency maps' specific to a given image and class [22].

Figure 2.4: (From [22]) Example of an image-specific class saliency map.

The formulation is based on finding the derivative of an output class with respect to an input image via back-propagation. The authors also formulate a method to generate an image that maximises the output class score for a particular class, to visualise the model's 'interpretation' of a class.

This paper sparked great interest in CNN explanations and further interest in creating explanations from network gradients generally. Along with DeconvNet and Guided Backpropagation (developed relatively simultaneously with similar ideas) these three methods are the most historically popular and influential saliency techniques[2]. A drawback of saliency maps is that noisy images can be produced when a model does not distinguish between objects that are being predicted and nearby objects that are associated (i.e. a tree with leaves, in an image of a bird).

[1] There is a strong overlap between gradient and backprop. methods: gradients as derivatives are approximated via backprop, and the weights updated by these gradients in training are the contributions to the neuron activations measured via backprop. techniques.

[2] Saliency maps are synonymous with gradient methods to the point where it is sometimes referred to as simply the 'Gradients' technique, as in Adebayo et al. (2018) [12], who may have done so to disambiguate the technique from saliency maps generally - see Section 1.1 ("Terminology").

Grad-CAM

Class activation maps (CAMs) are another approach aimed at understanding the behaviour of CNNs introduced by Zhou et al. (2015) [23], based on the motivation that deeper convolutional layers capture higher-level visual constructs while retaining spatial information [24]. While examining global average pooling (GAP) layers, a technique previously suggested for regularisation during training [25], the authors realised a final convolutional layer's separate RGB channels (or 'feature maps') could be input into a GAP layer with outputs used as features in a fully-connected layer, just before the final softmax layer. The class-associated weights in that fully-connected layer can be combined with the original final convolutional layer feature maps to capture deep representations as object localisations/class 'activation' maps:

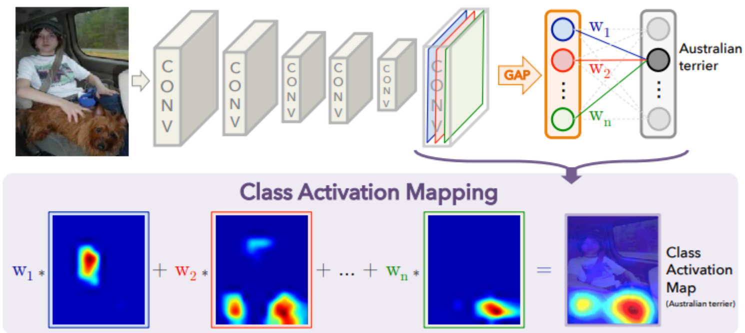

Figure 2.5: (From [23]) Summary of the CAM formulation. RGB channels are emphasised as inputs into the weighted sum that creates a CAM.

A major limitation of the CAM approach is that it requires specific CNN architectures without previous fully-connected layers, and a GAP layer to be added before the output softmax layer to generate the deep representations it visualises. As well as this being a hurdle to adoption, the representations are highly coarse and only roughly approximate class-associated regions (a reason for heatmap presentation).

Grad-CAM was proposed by Selvaraju et al. (2016) aimed at making CAMs applicable to a wider range of CNN models, and for visual tasks other than image classification [24]. It requires no alteration to model architecture. It still targets the final convolutional layer's channels, as in CAM, but uses the *gradient* of an output class score with respect to these channels' output activations to then globally-average-pool the gradients over the layer's width and height dimensions. The importance weights produced can be visualised as a localised heatmap over the input space, though the authors also multiply a Guided Backpropagation output (Section 2.3.1) with the heatmaps to generate a class-discriminative, "high-resolution" version called Guided Grad-CAM (Figure 2.6).

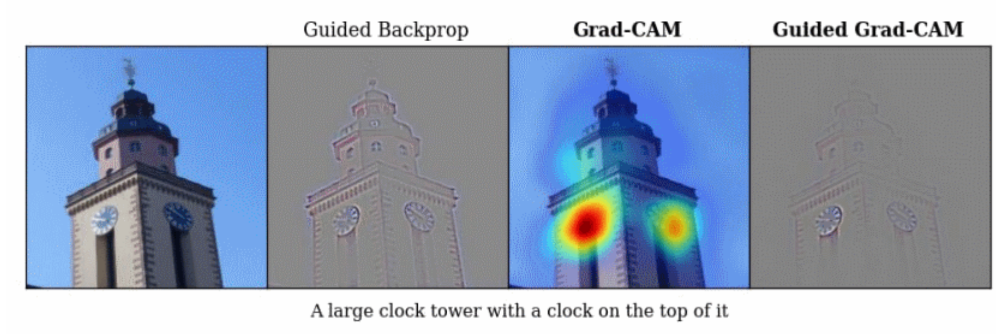

Figure 2.6: (Adapted from [24]) *Guided Backpropagation*, *Grad-CAM* and *Guided Grad-CAM* on an image captioning example from the Neuraltalk2 model.

One of Grad-CAM's strengths is that its authors prove its effectiveness for a variety of use cases. These include highlighting causes of incorrect predictions ('failure modes'), the effect of adversarial noise and causes of model confusion, and identifying training dataset bias[3]. Its application to a variety of models by other researchers ([26], [1]) is one testament to its authors' claims on cross-model applicability and explanation quality, as well as its popularity in the literature (>2000 citations).

Other Gradient Methods

Three other gradient-based methods are widely cited and relevant to discussion - Integrated Gradients [27], Gradient * Input [28] and SmoothGrad [29].

Integrated Gradients is similar to DeepLIFT - instead of taking difference-from-reference activation as a signal of contribution, the approach is to integrate the possible gradients over all inputs (from 0 up to the activation caused by the input image) and then use this area under the activation function as an information measure of input relevance. It also addresses the saturation problem (Section 2.3.1) though is naturally expensive to compute. Approximating the integral requires a baseline example that can provide zero input score as a comparison, and the method is applicable to a variety of architectures (two other similarities with DeepLIFT).

Gradient * Input was a simple proposal by DeepLIFT's authors to sharpen the saliency maps of Simonyan et al. (2014) [22] by multiplying the gradient with the original input signal. SmoothGrad was another attempt to sharpen gradient-based saliency maps, that noted gradients as derivatives can fluctuate sharply at local scales. To compensate for the noise in the output explanation, they propose a local average of gradient values that can be computed based on random samples of the original input image with Gaussian noise added.

[3]For example, they showed that a VGG model trained to classify nurses from doctors had learned to look at long hair to incorrectly label a female doctor a nurse, and the bias was because of gender-skewed training data.

2.3.3 Other Model-Specific Techniques

Backpropagation and gradient methods can be viewed as reflections of one approach based on a similar assumption: that propagating a relevant signal back through a neural network model is a means to explain how the signal was originally encoded. A third and more unrelated category of model-specific methods are perturbation techniques. These treat the underlying model as a true black-box by iteratively occluding patches of the input feature space in order to measure the change in output class score. Where the occlusion caused a noticeable change, the inference is that the region masked was important. Zeiler & Fergus (2013) ([13]) pioneered this approach in the same paper that proposed DeconvNets described in Section 2.3.1[4].

An influential work by Fong & Vedaldi (2017) [30] formalised a meta-predictor framework that can *learn* masks via what they describe as a deletion minimisation problem. Summarised, the informative region is found by minimising the size of a small deletion mask that causes the output score to drop significantly. This mask generation framework leads to similar results as other methods mentioned so far (Figure 2.7). The authors rely similarly on gradients to extract information to solve their optimisation problem, but a key performance drawback is that *many* gradient calculations for successively increasing mask sizes are required.

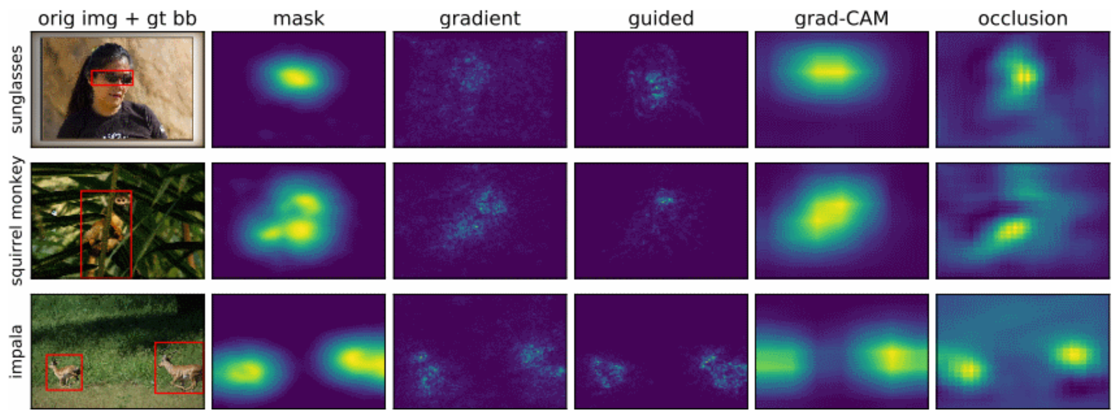

Figure 2.7: (Adapted from [30]) Comparison of methods discussed so far: Mask generation [30], Gradient (saliency maps) [22], Guided Backpropagation [17], Grad-CAM [24] and Occlusion [13]. Ground truth bounding boxes are labelled “gt bb”.

For completeness, some NN methods which do not attribute the contribution of input features specifically (or project internal model signals onto the input space) are not within project scope as a result, but are mentioned here. These are ‘concept vector’ methods that extract model behaviour across layers: Net2Vec [7] and Testing with Concept Activation Vectors (TCAV) [8].

[4] The grey-square masking method from the DeconvNet paper is usually referred to as ‘Occlusion’ in the literature.

2.4 Model-Agnostic Methods

In parallel to the literature aimed at explaining neural network model behaviour, a standalone approach in the interpretability literature is to find model-agnostic explanations. Similar to the motivation for perturbation-based methods (and with overlap), these are 'black-box' methods that make no restrictive assumptions about model architecture or classification task at all, and therefore can theoretically be applied to any model or task. Stark differences in approach and computational complexity still exist however. The most prominent methods in this sub-field are listed below.

2.4.1 Perturbation-Based

Described in Section 2.3.3, perturbing the input space (by toggling feature patches) is one way to isolate model behaviour. Changes in output score reveal importance in the occluded inputs. Several methods implement this intuition by training "surrogate", interpretable models around the isolated features that are locally relevant to a single prediction. These include Local Interpretable Model-Agnostic Explanations (LIME) [31], Anchors [32] and similar variants [5].

LIME

LIME's authors propose that non-linear, complex decision boundaries can be approximated locally around a single prediction via a simpler model that only relies only on the "neighbourhood" of the relevant input space [31].

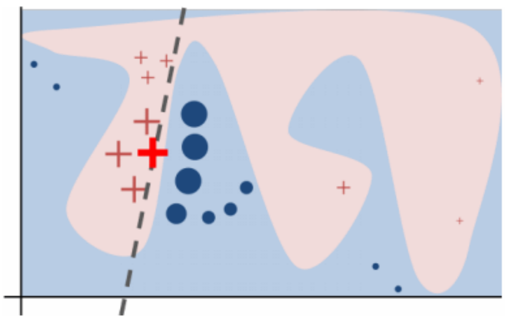

Figure 2.8: (From [31]) LIME's intuition about complex decision boundaries.

To compute explanations, LIME trains an interpretable model (i.e. decision tree or Lasso regression) on permuted samples from a given instance, weighted by their proximity to the instance being explained. The training simply minimises mean-squared error based on the samples' black-box predictions. Finally, the interpretable model's weights on these proximal features (superpixels in the case of image data) represent the 'explanation' for the underlying instance.

The formulation in the paper is relatively vague however, particularly with regards to the size of the region of influence / neighbourhood that should be used for finding relevant features. The vagueness was potentially motivated by their intention to be agnostic about the choice of explanation model and the black-box model, though this leaves questions about the accuracy-interpretability compromise largely unaddressed. Generating samples and training models for each sample is also extremely performance costly. Even for a confident prediction and using many samples, instability from random sampling makes an explanation undesirably non-deterministic. This was noted by the authors [31]. The authors of the mask generation technique described in Section 1.3.3 note that LIME bears similarity to their perturbation approach, but takes significantly longer to converge and produces a coarser heatmap via super-pixels instead of their pixel-level attribution [30].

Despite these shortcomings, LIME's simplicity and complete compatibility across all model families and classification tasks explains its high popularity among model-agnostic methods.

Anchors

LIME's problems motivated a more recent successor called "Anchors: High-Precision Model-Agnostic Explanations" by its original authors [32]. They replaced the use of simple local models with high fidelity if-then rules around a prediction, and give a framework for finding those rules efficiently. A rule (termed an 'anchor' explanation) is one that, "[...] sufficiently anchors an explanation locally, such that changes to the rest of the features of the instance do not matter" [32].

Formal ways to define the optimal region of influence ('coverage'), build rule accuracy ('precision'), and calculate rules efficiently were devised - three key improvements over their LIME proposal.

Though these anchor rules are highly interpretable, code available from the authors is limited and (possibly as a result) it has less popularity both in practice and in the literature. The GitHub repository provides for example a rough implementation for only text and tabular data and not image data [33].

2.4.2 Other Model-Agnostic Methods

SHAP Framework

Many methods have been listed so far and a reader may be fatigued of the variety. The SHAP framework (SHapley Additive exPlanations) by Lundberg & Lee (2017) was a push-back on this issue of method proliferation, showing links between many existing methods with a theoretical approach from game theory [34]. It has become extremely popular among researchers from all sub-fields in interpretability due to its attractive formal properties and 'outsider' formulation, as well as its implementation approximations for many model families and a well-maintained GitHub repository to host those implementations[5].

The game theoretic concept of Shapley values are a unique solution to the problem of calculating fair, marginal per-player rewards for the reward earned collectively in a cooperative game [36]. Since each player's contribution may produce interaction effects with others, they are calculated by averaging contributions in all possible sub-coalitions of players. In the context of machine learning, if features are taken as players, the method can be applied to a prediction (the "payoff") and it retains its per-player theoretical properties. These qualities include *efficiency*, which is that the sum of p constituent input feature contributions (Shapley value ϕ_j for a feature j) must equal the difference between a prediction of an input x ($\widehat{f}_x$) and an average for all inputs (an 'expectation' of model output $E_X(\widehat{f}(X))$):

$$\sum_{j=1}^{p} \phi_j = \widehat{f}_x - E_X(\widehat{f}(X)) \tag{2.2}$$

A key problem is that finding ϕ_j requires iteratively computing outputs for all feature subsets including j and is therefore computationally infeasible for large input spaces[6].

SHAP values are an implementable version of Shapley values. They connect Shapley value theory with local explanation techniques including LIME and DeepLIFT [34]. They are formulated as the difference between expected model output (approximated by an average background data sample's prediction) and the instance at hand being predicted. The authors show that the game theoretical properties including efficiency apply to a whole class of 'additive feature attribution methods', both model-specific (e.g. DeepLIFT) and model-agnostic. They provide several model-specific approximations for implementing their attributions, one based on Integrated Gradients and another based on DeepLIFT.

2.5 Evaluation of Attribution Methods

Creators of feature attribution methods tend to provide visual comparisons with existing methods, as for example in Figure 2.7. An issue is that qualitative evaluation is often left to the reader, who is encouraged to infer the author's saliency maps are either equal in standing or more aesthetically pleasing than another method's.

[5] The SHAP paper has over 1000 citations and its GitHub has over 9000 'stars' at the time of writing [35], despite only being published in late 2017.

[6] They are however commonly used in low-dimensional linear regression settings, like in economics for example, for calculating global feature importances when features are dependent [36].

Lack of evaluation *independent* of method creators is another issue. Introduced methods do often suggest their own interpretability criteria, though with little reference to criteria in the prior art, and so the interpretability field's agreement on desirable standards for interpretability is low. To some extent, the proliferation in attribution methods (mostly variants) can be attributed to these unfortunately subjective and still actively researched desiderata.

There have however been efforts to unify existing methods. SHAP's momentum in the literature can somewhat be explained by its ability to show how existing methods are related and that it demonstrates where one can be preferable. This research section therefore aims to:

1. Overview attempts like SHAP that have independently evaluated methods and performed 'unification' or comparison studies;
2. Examine existing evaluation criteria that method authors have independently proposed, and;
3. Describe existing software for generating explanations, that package several methods for visual comparison for use by researchers and practitioners.

2.5.1 Existing Studies

Towards a Rigorous Science of Interpretable Machine Learning [37]

Doshi-Velez & Kim (2017) provided an influential set of guidelines or "evaluation paradigms" for evaluating interpretability techniques, arranged in three levels of increasing abstraction [37]. This motivation was to provide a scientific basis for the different objectives in evaluation that method authors implicitly target (both qualitative and quantitative):

1. **Functionally-grounded**: How well the attribution method performs on quantitative criteria: 'proxy' metrics for explainability like weakly supervised object localisation (Section X).
2. **Human-grounded**: How much everyday people agree that an explanation is visually superior to another: user studies as an example.
3. **Application-grounded**: How well the explanation helps domain experts solve real tasks; such as a radiologist agreeing with fractures pointed out by a model trained on X-ray data.

Research on evaluation criteria below (Section 2.5.2) has focused on functionally-grounded criteria, since these are more common in the literature and since higher-level paradigms are difficult to compare objectively.

Towards Better Understanding of Gradient-based Attribution Methods [38]

Similar to SHAP in motivation, an independent attempt at providing a unified framework for the gradient-based class[7] of attribution methods was provided by Ancona et al. (2017) [38]. In their work they prove conditions of equivalence between Layerwise Relevance Propagation and Gradient * Input, and DeepLIFT and Integrated Gradients (Section 2.3). Another major contribution was to propose a generalisation for the ideal 'additivity' property that SHAP's authors noted defined a class of methods (Section 2.4.2)[8]. This criteria was termed *Sensitivity-n*: "[...] when the sum of the attributions for any subset of features of cardinality n is equal to the variation of the output S_C caused by removing the features in the subset" [38].

Some of their key insights were that existing methods like DeepLIFT and Integrated Gradients show very high correlation, that the former therefore acts as an approximation of the latter in practice, and that on complex model architectures like InceptionV3, all gradient-based methods produce noisier saliency maps and less appealing explanations.

Sanity Checks for Saliency Maps [12]

This paper by Adebayo et al. (2018), along with Lundberg & Lee (2017) (SHAP's authors) and Ancona et al. (2017) above, have been the three main independent contributions towards understanding and unifying the model-specific set of feature attribution methods described in earlier sections of this chapter. This motivation was similar though more practical than theoretical: find an actionable methodology to help a practitioner / researcher decide between competing attribution methods.

The methodology proposed was a set of two statistical randomisation tests ('sanity checks'):

1. **Model Parameter Randomisation Test:** A misleading attribution method could be insensitive to model parameters, so an untrained version of the same model with random weights should not produce a similar saliency map.

2. **Data Randomisation Test:** A misleading method could be dependent on training labels: a test with randomly permuted labels should therefore show significantly different saliency maps.

[7] The authors do not distinguish between backpropagation and gradient-based methods.

[8] The authors actually reference the properties of *Completeness* (proposed by the authors of Integrated Gradients [27]) and *Summation to Delta* (proposed by the authors of DeepLIFT [10]), two variants of a similar idea to additivity.

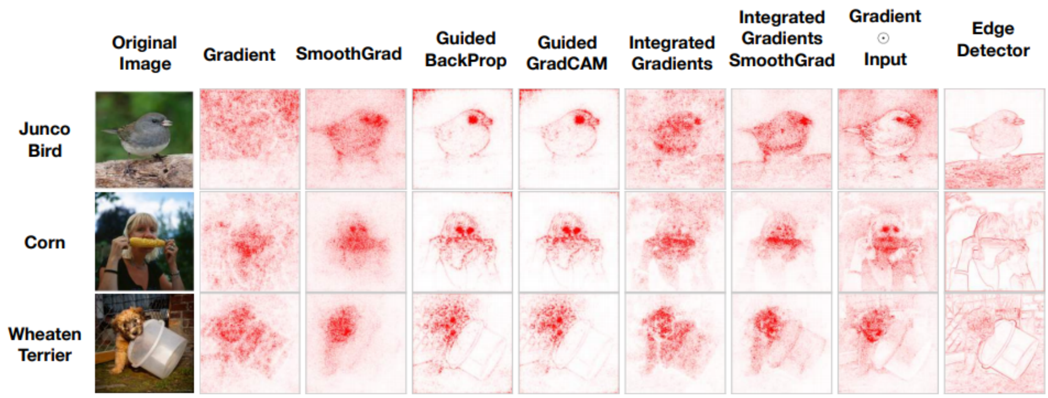

Figure 2.9: (From [12]) The panel of methods examined by Adebayo et al. (2018), all previously discussed in Section 2.3. They highlight that an edge detector alone can produce a mask similar to the output of some methods.

For comparisons between a normal attribution output and a modified one according to one of those two experiment conditions, the authors use similarity metrics including Spearman rank correlation and a structural similarity (SSIM) index. Note these metrics do not evaluate one method's similarity with another: only the similarity with its modified version. Some insights are prepared from these metrics, like that Guided Backpropagation and Gradient * Input show unexpected insensitivity (perform badly on the sanity checks).

One problematic point they emphasise though is that non-performing saliency methods may only be visually salient because they act like an edge detector (Figure 2.10). Although the work valuably points out that observers might have confirmation bias when viewing highlighted edges in a saliency map, this qualitative conclusion ignores the role of connected regions and pixel-wise attribution intensities, which some methods may have as a strength over others. These latter two criteria are arguably as important or more important for visual quality than edges alone, and the authors' similarity metric results also do not relate to the standalone observation about edges.

2.5.2 Existing Criteria

Saliency Metrics (Quantitative Approach)

Some method authors however have made an effort to account for regions in method evaluation. In particular, 'weakly supervised object localisation' (WSOL) has been the technique used for evaluating saliency maps via bounding boxes derived from a segmentation algorithm (in natural image classification settings). It was first applied by Simonyan et al. (2014) while introducing the original saliency maps method (Section 2.3.2) [22] . Those authors used a colour segmentation algorithm on thresholded saliency maps, found bounding boxes on the segmented regions, and then submitted their annotations to an ILSVRC-2103 localisation challenge where they out-performed many fully supervised algorithms.

The paper that introduced CAM saliency maps (Section 2.3.2, [23]) used a similar technique and has been considered the seminal work for use of WSOL [39]. They chose a 20% quantile threshold for CAM values before finding the largest connected component in each map. Unlike Simonyan et al. who only evaluated via the competition, CAM's authors independently evaluate their derived bounding boxes using an intersection-over-union metric (thresholded at 0.5 to exclude instances where the method was significantly off). Their results on an ILSVRC validation set outperformed the benchmarks of localisation-trained models.

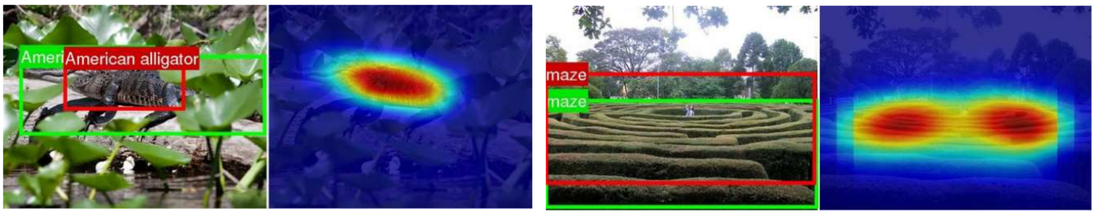

Figure 2.10: (Adapted from [23]) CAM's predicted bounding box in green and the ground truth annotation in red (left in each sub-pane), and the original CAM (right in each sub-pane). Their IOU metric is calculated over the annotations.

Notably this IOU metric does not account for the weight of the attribution in any one pixel, which is relevant to most methods. The methodology also unfairly penalises more perforated-looking saliency maps due to relying on segmentation algorithms to find connected components and therefore ignoring pixel-level detail. On the one hand, connected components are more observer-friendly explanations of single objects (with an implicit assumption that the underlying model also cares about connected regions), though on the other hand some methods may be noisier and so may be punished harshly for having small, disconnected sub-regions yet still maintaining saliency in the 'bigger picture'.

Dabkowski & Gal (2017) compensate for the latter problem by proposing an augmented localisation metric that takes a bounding box of the *entire* salient region, rather than just a thresholded and single-component subset of the saliency map [40]. A strong aspect of their work was to provide a "Max box" baseline in reporting localisation error: this reports the ground truth box's overlap with the whole image. Where the localisation error is similar, this allows them to suggest the interpretability of their generated localisation boxes is similar to the ground truth boxes themselves.

Independently of the explainability literature, pixel-wise mask-based metrics have been developed for generic WSOL applications by Choe et al. (2020) [39], though it seems these have not been applied in the context of evaluating explainability techniques.

Higher Level Criteria (Qualitative Approach)

Some authors sidestep the functionally-grounded approach to method evaluation and evaluate on higher level criteria. These proposed desiderata are summarised in Table 1.X, from lower to higher levels of abstraction:

consistency between models input invariance transferability trust fair and ethical decision making

Across in the model-agnostic camp Ribeiro et al. (2016) have made the case for model flexibility as a strong motivation for explanation technique usability [41]. When comparing a tree-based model with a deep learning model, for example, neither a practitioner nor a researcher can get a fair idea of interpretability if two explanations are different in representation on account of different attribution methods. The observation by Ancona et al. (2017) that more complex architectures are not explained as well by current model-specific techniques also supports this argument (Section 2.5.1) [38].

2.5.3 Existing Explanation Frameworks

Ancona et al. (2017) developed a software package called *DeepExplainer* to support their unified framework [42]. It implements a suite of the methods they showed relationships between: Saliency Maps, Gradient * Input, Integrated Gradients, DeepLIFT and Occlusion (Section 2.3). It provides abstraction over these methods with some parameter specification allowed: .

Aimed at practitioners for auditing their models for bias, *FairML* was developed by Adebayo (2017) [43]. This package

There are no explanation frameworks for collecting both model-agnostic and model-specific techniques for saliency maps. Additionally, no automated method evaluation frameworks for WSOL-based evaluation are known to be available.

Chapter 3

Methodology

3.1 Overview

A panel of four local attribution methods were chosen for evaluation in this project, picked for representativeness of approach and their relative prominence in the literature. The panel was also a balanced selection from both 'classes' of method approach: model-specific (Section 2.3) and model-agnostic (Section 2.4):

1. **DeepLIFT:** Model-specific class, backpropagation-based approach (2.3.1)
2. **GradCAM:** Model-specific class, gradient-based approach (2.3.2)
3. **LIME:** Model-agnostic class, perturbation-based approach (2.4.1)
4. **SHAP:** Model-agnostic class[1], generic approach (2.4.2)

Other methods could have been included in this panel though as related work has shown in 2.5.1 (particularly by Ancona et al. (2017)), similarities in formulation should allow conclusions on one method to generalise well for close relatives. Project constraints also meant that a decision on the panel breadth had to be made according to at least some criteria, and representativeness helps for comparing approaches (the main project goal).

This chapter is presented in sequential order of project milestones, from initial data collection through to fine-tuning of the evaluation metrics designed. The evaluation methodology itself can be summarised in terms of the dataset used, 'off the shelf' underlying models relied upon and the evaluation metric approach:

[1] As a higher level formulation, SHAP has to be implemented for each model family. It's therefore better described as *partially* model-agnostic.

1. **Dataset:** ImageNet validation set, with ground truth bounding box annotations usually used for object localisation training.

2. **Models:** VGG16 as the primary model, InceptionV3 and ResNet50 for supportive analysis.

3. **Metrics:** Pixel-wise and mask-based WSOL, extending related work in Section 2.5.2.

These design decisions are mentioned in more detail in Sections 3.2, 3.3 and 3.7 below respectively. Finally, to support the metric analysis, qualitative analysis was also performed, though this is left to the Results and Discussion chapters.

3.2 Data Collection & Annotation

The ImageNet Large Scale Visual Recognition Challenge (ILSVRC) uses a subset of the well-known, hierarchically-labelled ImageNet database [44]. The competition's validation set consists of 50,000 images of 1000 categories, and includes annotations for ground truth class labels for image classification, and ground truth bounding boxes for object localisation.

This dataset was chosen because of the convenience of the bounding boxes for method evaluation, and the availability of models pre-trained on ImageNet. Images and bounding box XML files for all 50,000 instances in the 2012 validation set were acquired from an academically hosted torrent.

Each bounding box file contains one or several annotations (several if there are multiple instances of a single class, e.g Figure 3.1), and each annotation contains the ImageNet class ID[2] and $\{$*x-min, x-max, y-min, y-max*$\}$ fields for the box.

A script to draw rectangular annotations on the images was created to sanity check the acquired data. Example outputs are shown in Figure 3.1.

Figure 3.1: Collected ImageNet examples with annotations drawn.

[2]These ImageNet 'synsnet' IDs were converted to human-readable class labels using a script that connects to a standalone map of IDs to labels.

Much later into the project this script was upgraded to return the annotated image (resized for comparison) as a 2D array of 0's and 1's, with 1's inside the bounded regions to create a ground truth mask. The use of this mask for saliency calculation is described later (Section 3.7).

3.3 Models and Preprocessing

ImageBuilder class

3.4 Initial Method Investigation

3.5 Software Abstraction I

3.6 Adapting Methods for Compatibility

3.7 Evaluation Metric Design

The motivation to create a new variant of an intersection-over-union metric, rather than rely on existing bounding box based comparison metrics, was based on observations in Related Work that existing metrics do not account for pixel-level detail and punish perforated saliency maps based on their reliance on a single component to find the box.

Performance wise, it is more taxing to account for pixel-wise attribution **cite?**.

3.8 Software Abstraction II

3.9 Result Collection

GTX-1080.

3.10 Support for Other Models

Chapter 4

Results

Chapter 5

Discussion

KL-Lime discussion on role of itnerpretable models

Chapter 6

Conclusions

6.1 Summary and conclusions

6.2 Possible future work

Appendix A

Software Documentation

Appendix B

Software Descriptions

B.1 Attributer Class

Text

B.2 Evaluator Class

Appendix C

Software Repository Link

Bibliography

[1] J. R. Zech, M. A. Badgeley, M. Liu, A. B. Costa, J. J. Titano, and E. K. Oermann, "Confounding variables can degrade generalization performance of radiological deep learning models," *CoRR*, vol. abs/1807.00431, 2018. arXiv: `1807.00431`. [Online]. Available: `http://arxiv.org/abs/1807.00431`.

[2] S. Kiritchenko and S. M. Mohammad, *Examining gender and race bias in two hundred sentiment analysis systems*, 2018. arXiv: `1805.04508 [cs.CL]`.

[3] D. Diviny, *How to effectively use machine learning to implement public policy*, 2019. [Online]. Available: `https://www.nousgroup.com/insights/effectively-use-machine-learning-implement-public-policy/`.

[4] C. Rudin, *Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead*, 2018. arXiv: `1811.10154 [stat.ML]`.

[5] T. Peltola, "Local interpretable model-agnostic explanations of bayesian predictive models via kullback-leibler projections," *CoRR*, vol. abs/1810.02678, 2018. arXiv: `1810.02678`. [Online]. Available: `http://arxiv.org/abs/1810.02678`.

[6] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, *Object detectors emerge in deep scene cnns*, 2014. arXiv: `1412.6856 [cs.CV]`.

[7] R. Fong and A. Vedaldi, "Net2vec: Quantifying and explaining how concepts are encoded by filters in deep neural networks," *CoRR*, vol. abs/1801.03454, 2018. arXiv: `1801.03454`. [Online]. Available: `http://arxiv.org/abs/1801.03454`.

[8] B. Kim, M. Wattenberg, J. Gilmer, C. Cai, J. Wexler, F. Viegas, and R. Sayres, *Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav)*, 2017. arXiv: `1711.11279 [stat.ML]`.

[9] I. J. Goodfellow, J. Shlens, and C. Szegedy, *Explaining and harnessing adversarial examples*, 2014. arXiv: `1412.6572 [stat.ML]`.

[10] A. Shrikumar, P. Greenside, and A. Kundaje, *Learning important features through propagating activation differences*, 2017. arXiv: `1704.02685 [cs.CV]`.

[11] P.-J. Kindermans, K. T. Schütt, M. Alber, K.-R. Müller, D. Erhan, B. Kim, and S. Dähne, *Learning how to explain neural networks: Patternnet and patternattribution*, 2017. arXiv: 1705.05598 [stat.ML].

[12] J. Adebayo, J. Gilmer, M. Muelly, I. J. Goodfellow, M. Hardt, and B. Kim, "Sanity checks for saliency maps," *CoRR*, vol. abs/1810.03292, 2018. arXiv: 1810.03292. [Online]. Available: http://arxiv.org/abs/1810.03292.

[13] M. D. Zeiler and R. Fergus, *Visualizing and understanding convolutional networks*, 2013. arXiv: 1311.2901 [cs.CV].

[14] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008. [Online]. Available: http://www.jmlr.org/papers/v9/vandermaaten08a.html.

[15] A. Fabisch. (2014). "T-sne in scikit learn," [Online]. Available: https://alexanderfabisch.github.io/t-sne-in-scikit-learn.html.

[16] J. Friedman, "Greedy function approximation: A gradient boosting machine," *The Annals of Statistics*, vol. 29, Nov. 2000. DOI: 10.1214/aos/1013203451.

[17] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, *Striving for simplicity: The all convolutional net*, 2014. arXiv: 1412.6806 [cs.LG].

[18] A. Shrikumar. (2019). "Deeplift github repository," [Online]. Available: https://github.com/kundajelab/deeplift.

[19] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek, "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation," *PLoS ONE*, vol. 10, 2015.

[20] G. Montavon, S. Lapuschkin, A. Binder, W. Samek, and K.-R. Müller, "Explaining nonlinear classification decisions with deep taylor decomposition," *Pattern Recognition*, vol. 65, pp. 211–222, May 2017, ISSN: 0031-3203. DOI: 10.1016/j.patcog.2016.11.008. [Online]. Available: http://dx.doi.org/10.1016/j.patcog.2016.11.008.

[21] D. Baehrens, T. Schroeter, S. Harmeling, M. Kawanabe, K. Hansen, and K.-R. Mueller, *How to explain individual classification decisions*, 2009. arXiv: 0912.1128 [stat.ML].

[22] K. Simonyan, A. Vedaldi, and A. Zisserman, *Deep inside convolutional networks: Visualising image classification models and saliency maps*, 2013. arXiv: 1312.6034 [cs.CV].

[23] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, *Learning deep features for discriminative localization*, 2015. arXiv: 1512.04150 [cs.CV].

[24] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," *International Journal of Computer Vision*, vol. 128, no. 2, pp. 336–359, Oct. 2019, ISSN: 1573-1405. DOI: `10.1007/s11263-019-01228-7`. [Online]. Available: `http://dx.doi.org/10.1007/s11263-019-01228-7`.

[25] M. Lin, Q. Chen, and S. Yan, *Network in network*, 2013. arXiv: `1312.4400 [cs.NE]`.

[26] A. Chattopadhay, A. Sarkar, P. Howlader, and V. N. Balasubramanian, "Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks," *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, Mar. 2018. DOI: `10.1109/wacv.2018.00097`. [Online]. Available: `http://dx.doi.org/10.1109/WACV.2018.00097`.

[27] M. Sundararajan, A. Taly, and Q. Yan, *Axiomatic attribution for deep networks*, 2017. arXiv: `1703.01365 [cs.LG]`.

[28] A. Shrikumar, P. Greenside, A. Shcherbina, and A. Kundaje, *Not just a black box: Learning important features through propagating activation differences*, 2016. arXiv: `1605.01713 [cs.LG]`.

[29] D. Smilkov, N. Thorat, B. Kim, F. Viégas, and M. Wattenberg, *Smoothgrad: Removing noise by adding noise*, 2017. arXiv: `1706.03825 [cs.LG]`.

[30] R. C. Fong and A. Vedaldi, "Interpretable explanations of black boxes by meaningful perturbation," *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct. 2017. DOI: `10.1109/iccv.2017.371`. [Online]. Available: `http://dx.doi.org/10.1109/ICCV.2017.371`.

[31] M. T. Ribeiro, S. Singh, and C. Guestrin, *"why should i trust you?": Explaining the predictions of any classifier*, 2016. arXiv: `1602.04938 [cs.LG]`.

[32] ——, "Anchors: High-precision model-agnostic explanations," in *AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

[33] M. T. Ribeiro. (2020). "Anchors github repository," [Online]. Available: `https://github.com/marcotcr/anchor`.

[34] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., Curran Associates, Inc., 2017, pp. 4765–4774. [Online]. Available: `http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf`.

[35] S. Lundberg. (2020). "Shap github repository," [Online]. Available: `https://github.com/slundberg/shap`.

[36] K. Aas, M. Jullum, and A. Løland, *Explaining individual predictions when features are dependent: More accurate approximations to shapley values*, 2019. arXiv: 1903.10464 [stat.ML].

[37] F. Doshi-Velez and B. Kim, *Towards a rigorous science of interpretable machine learning*, 2017. arXiv: 1702.08608 [stat.ML].

[38] M. Ancona, E. Ceolini, C. Öztireli, and M. Gross, *Towards better understanding of gradient-based attribution methods for deep neural networks*, 2017. arXiv: 1711.06104 [cs.LG].

[39] J. Choe, S. J. Oh, S. Lee, S. Chun, Z. Akata, and H. Shim, *Evaluating weakly supervised object localization methods right*, 2020. arXiv: 2001.07437 [cs.CV].

[40] P. Dabkowski and Y. Gal, *Real time image saliency for black box classifiers*, 2017. arXiv: 1705.07857 [stat.ML].

[41] M. T. Ribeiro, S. Singh, and C. Guestrin, *Model-agnostic interpretability of machine learning*, 2016. arXiv: 1606.05386 [stat.ML].

[42] M. Ancona. (2019). "Deep explainer github repository," [Online]. Available: https://github.com/marcoancona/DeepExplain.

[43] J. Adebayo. (2017). "Fairml github repository," [Online]. Available: https://github.com/adebayoj/FairML.

[44] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015. DOI: 10.1007/s11263-015-0816-y.