

Quantifying the Performance of Explainability Algorithms

by

Zhong Qiu Lin

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Systems Design Engineering

Waterloo, Ontario, Canada, 2020

© Zhong Qiu Lin 2020

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

Supervisor(s): Alexander Wong
 Associate Professor,
 Dept. of Systems Design Engineering,
 University of Waterloo

Internal Member: David A Clausi
 Professor, Assoc Dean, Research & Extern,
 Dept. of Systems Design Engineering,
 University of Waterloo

Internal Member: Katharine Andrea Scott
 Assistant Professor,
 Dept. of Systems Design Engineering,
 University of Waterloo

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

The following paper is used in this thesis. It is described below:

Lin, Z.Q., Shafiee, M.J., Bochkarev, S., Jules, M.S., Wang, X.Y. and Wong, A., 2019. Explaining with Impact: A Machine-centric Strategy to Quantify the Performance of Explainability Algorithms. Proceedings of Conference on Neural Information Processing Systems (NeurIPS) 2020 Workshops

Contributor	Statement of Contribution
Lin, Z.Q. (Candidate)	Conceptual design (70%)
	Data collection and analysis (90%)
	Writing and editing (40%)
Shafiee, M.J.	Conceptual design (10%)
	Writing and editing (20%)
Bochkarev, S.	Data collection and analysis (5%)
Jules, M.S.	Conceptual design (5%)
	Data collection and analysis (5%)
Wang, X.Y.	Conceptual design (5%)
Wong, A.	Conceptual design (10%)
	Writing and editing (40%)

The following paper is used in this thesis. It is described below:

Lin, Z.Q. and Wong, A., 2019. Progressive Label Distillation: Learning Input-Efficient Deep Neural Networks. Proceedings of Conference on Neural Information Processing Systems (NeurIPS) 2020 Workshops

Contributor	Statement of Contribution
Lin, Z.Q. (Candidate)	Conceptual design (90%)
	Data collection and analysis (100%)
	Writing and editing (80%)
Wong, A.	Conceptual design (10%)
	Writing and editing (20%)

Abstract

Given the complexity of the deep neural network (DNN), DNN has long been criticized for its lack of interpretability in its decision-making process. This 'black box' nature has been preventing the adaption of DNN in life-critical tasks. In recent years, there has been a surge of interest around the concept of artificial intelligence explainability/interpretability (XAI), where the goal is to produce an interpretation for a decision made by a DNN algorithm. While many explainability algorithms have been proposed for peaking into the decision-making process of DNN, there has been a limited exploration into the assessment of the performance of explainability methods, with most evaluations centred around subjective human visual perception of the produced interpretations. In this study, we explore a more objective strategy for quantifying the performance of explainability algorithms on DNNs. More specifically, we propose two quantitative performance metrics: i) **Impact Score** and ii) **Impact Coverage**. Impact Score assesses the percentage of critical factors with either strong confidence reduction impact or decision shifting impact. Impact Coverage assesses the percentage overlapping of adversarially impacted factors in the input. Furthermore, a comprehensive analysis using this approach was conducted on several explainability methods (LIME, SHAP, and Expected Gradients) on different task domains, such as visual perception, speech recognition and natural language processing (NLP). The empirical evidence suggests that there is significant room for improvement for all evaluated explainability methods. At the same time, the evidence also suggests that even the latest explainability methods can not produce steady better results across different task domains and different test scenarios.

Acknowledgements

During the preparation of the master’s thesis, I have received tremendous help from many people. Their comments and advices contribute to the accomplishment of the thesis.

First and foremost, I would like to express my sincere gratitude to my supervisor Prof. Alexander Wong (fondly referred to as the “research dad”), whose unwavering support and eagerness to pursue all things computer vision and optimization related has long been a source of inspiration. Thank you for being a wonderful mentor, for guiding me as I grow as a researcher, and for unfolding a path in the cloud of confusion.

I would like to thank Prof. David Clausi and Prof. Andrea Scott for reviewing my thesis. Carving out the time from your exceedingly busy schedules to read and revise my thesis is greatly appreciated.

I would also like to thank my family and friends for their constant support and blind faith in my abilities. Thanks for listening to me spew information about topics you were unfamiliar with.

Dedication

This is dedicated to the people around the world fighting against coronavirus.

“Ye are all fruits of one tree, the leaves of one branch, the flowers of one garden.”

Table of Contents

List of Figures	xi
List of Tables	xiii
1 Introduction	1
1.1 State of the Art	1
1.1.1 Explainable AI (XAI)	3
1.2 Thesis Overview	6
1.2.1 Motivation	6
1.2.2 Contributions	7
1.2.3 Outline	8
2 Background	10
2.1 Deep Learning	10
2.1.1 Neural Networks	11
2.2 Convolutional Neural Networks (CNNs)	12
2.2.1 Convolutional Layers	12
2.3 Different Data Modalities Classification	12
2.3.1 Visual Image Classification	13
2.3.2 Speech Utterance Classification	14
2.3.3 Text Sentiment Classification	16

2.4	Adversarial Attack for Neural Network	20
2.4.1	Adversarial Example Generation	21
3	Explainability Methods and Assessments	22
3.1	Explainability methods	22
3.1.1	Local Interpretable Model-agnostic Explanations (LIME)	23
3.1.2	SHapley Additive exPlanations (SHAP)	23
3.1.3	Expected Gradients (EG)	24
3.1.4	Choices of Explainability methods	24
3.2	Explainability Assessment in Literatures	25
3.2.1	Human Assessment	25
3.2.2	RemOve And Retrain (ROAR)	28
3.2.3	Deletion and Insertion	28
4	Methodology	29
4.1	Quantifying Explainability	29
4.1.1	Assessing Impact on Decisions	30
4.1.2	Assessing Erroneous Coverage	33
5	Experiments	35
5.1	Experimental Setup	35
5.1.1	Experiment 1: General Scenario	35
5.1.2	Experiment 2: Adversarial Distraction	37
5.1.3	Explainability Methods Under Study	41
5.2	Experimental Results	41
5.2.1	Experiment 1: General Scenario	41
5.2.2	Experiment 2: Adversarial Distraction	42

6 Conclusion	47
6.1 Summary	47
6.2 Future Works	48
References	50

List of Figures

1.1	Progression of machine learning models [13]	1
1.2	Visual examples of saliency map in literatures.	4
1.3	Example of a decision change due to the absence of critical regions in the decision-making process.	5
1.4	Saliency explanation for a binary text classifier [64]	6
2.1	Illustration of deep learning neural network	11
2.2	Residual connection and ResNet architecture [27].	14
2.3	Speech utterance classification network architecture [52]	15
2.4	Comparison between CBOW and Skip-gram [54]	18
2.5	Text sentiment classification network architecture [41]	19
2.6	Adversarial example [24]	20
3.1	AMT interfaces for evaluating class discrimination ability [72]	26
3.2	Saliency maps for some common methods compared to an edge detector [2]	26
4.1	Impact Score flow diagram	31
4.2	Impact Coverage flow diagram	34
5.1	Example of a decision change due to the absence of critical regions in the decision-making process	36
5.2	Example of a decision change due to replacement of critical tokens in the decision-making process	37

5.3	Example of a directed erroneous decision due to the adversarially impacted area	38
5.4	Example of a decision change due to adversarially impacted area	39
5.5	Example of a decision change due to the adversarially tempered tokens	40
5.6	Examples of identified critical regions in images and corresponding confidences for image classification	43
5.7	Examples of identified critical tokens in sentences and corresponding confidences for text sentiment classification	45
5.8	Examples of adversarially modified images, identified critical regions and corresponding confidences	46

List of Tables

2.1	List of words associated with “Sweden” using Word2Vec [55]	17
5.1	Performance of tested explainability methods based on impact on network decisions.	41
5.2	Image Classification: Performance of tested explainability methods at different adversarial scales	42
5.3	Utterance Classification: Performance of tested explainability methods at different area of adversarial patch	42
5.4	Sentiment Classification: Performance of tested explainability methods at different number of adversarial tokens	44

Chapter 1

Introduction

1.1 State of the Art

In the quest for more accurate artificial intelligence, we have seen a progression from handcrafted rules and heuristics, to linear models and decision trees, ensembles and deep learning models to, most recently, meta-learning or models that create other models, as shown in Fig. 1.1. The advancement of computational hardware coupled with increasing dataset sizes and the availability of open-source learning frameworks have fueled a trend towards more complex non-linear models. Particularly, the recent significant advances in deep learning models [49] has resulted in a paradigm shift along multiple dimensions:

- **Expressiveness** unlocks the ability to fit a wide range of complex functions, as such it also enables deep learning models to extract high-level abstract representation from data.

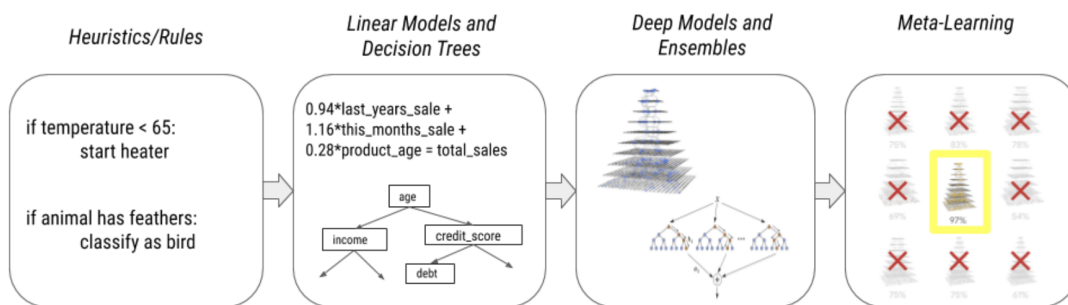


Figure 1.1: Progression of machine learning models [13]

- **Versatility** enables wide adoption across multiple data modalities (image, audio, speech, text, tabular, time series, etc.). Recent researches in deep learning models have led to state-of-the-art performance across various tasks such as visual perception [82, 32, 63], speech recognition [4], and natural language processing (NLP) [96, 21].
- **Adaptability** allows deep learning models being applied in small data regimes through transfer learning, meta-learning and multi-task learning.

On the flip side, these more complex models have become increasingly opaque. Combining with the fact that these models are still fundamentally built around correlation and association, several concerns have raised both in academia and industries challenging the usage of these complex models. Particularly, as the proliferation of deep learning continues, there is now a growing interest as well as concern over how deep neural networks are making decisions, especially for life-critical applications such as autonomous driving and clinical decision support. The major concerns are listed in the following:

- **Spurious correlations** can be learned from the data, often hampering the model’s ability to generalize and leading to poor real-world results.
- **Loss of debuggability and transparency** leading to low trust as well as the inability to fix or improve the models and/or outcomes. Given the sheer complexity of deep neural networks and how information propagates through such networks to form a decision, deep learning has been often viewed as a ‘black box’ machine learning method and very difficult to interpret and understand the decision-making process or the key factors involved in the decision. This lack of transparency impedes the adoption of these models, especially in regulated industries (e.g. Banking & Finance or Healthcare).
- **Loss of control** due to model practitioners’ reduced ability to locally adjust model behaviour in problematic instances. Furthermore, this challenge makes it very difficult for machine learning engineers and scientists to understand biases and error scenarios of the trained models to improve upon.
- **Indesirable data amplification** reflecting biases that don’t agree with our societal norms and principles. The lack of interpretability often results in oversight situations where the models are deciding based on unintended patterns in the dataset [58].
- **Vulnerable to malicious attacks**, even in the physical world. This is particularly critical given the recent rise of adversarial examples [83, 3, 24, 46, 73, 19, 23, 10].

These well-designed samples can easily fool a well-performed deep learning model and cause deep learning models to make erroneous decisions.

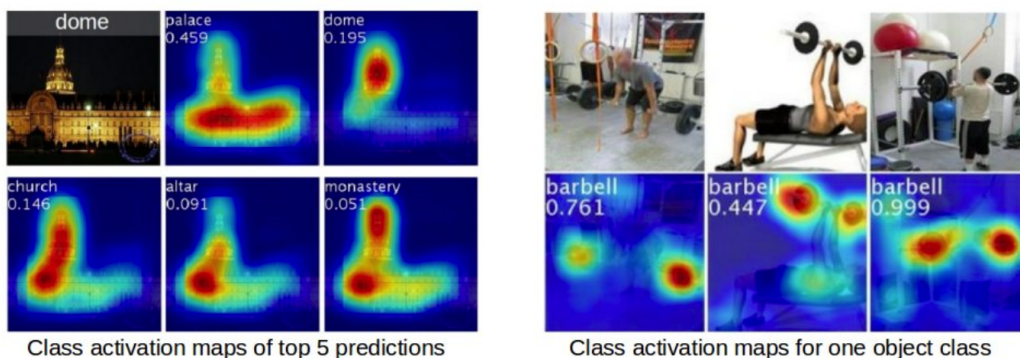
As such, the ability to explain the decision-making process of deep neural networks can be critical for enabling the development of improved, more dependable deep learning as well as enable the use of deep learning in a more trust-worthy manner in mission-critical scenarios.

1.1.1 Explainable AI (XAI)

Systems built around machine learning models will affect and, in many cases, redefine autonomous transportation, financial management, medical interventions, and many other areas of society. However, considering the challenges discussed in the previous section, the usefulness and fairness of these systems should be gated by the ability to understand, explain and control them. The field of explainable AI (XAI) has been a resurgence since the early days of expert systems [78] a few decades ago. In the work by Doshi et al. [16], the authors define XAI as “the ability to explain or to present in understandable terms to a human”. Recent research progress in XAI has been rapidly advancing. Different lines of XAI researches have been proposed trying to address the aforementioned concerns from different perspectives. These works include, but not limited to, input attribution [64, 53, 80, 72], concept testing/extraction [40, 90, 22], example influence/matching [93, 39, 43], distillation [29, 20]. Furthermore, new novel approaches have been proposed for building inherently interpretable and controllable models like Deep Lattice Networks [95] and Bayesian models. In addition to needing to probe the internals of increasingly complex models, which in and of itself is a challenging computational problem, a successful XAI system must provide explanations to people.

Input Attribution

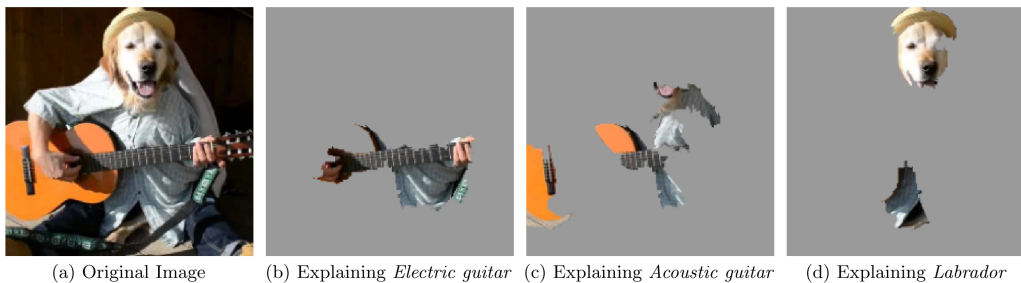
Input attribution (aka saliency map) is the most common type of explanation families [64, 53, 80, 72, 75, 17, 76, 65, 38]. It provides an explanation for an instance prediction of a model in terms of input features using importance scores. The individual importance scores are meant to communicate the relative contribution of each input feature to the instance prediction. In other words, the higher the score associated with, the more impactful the input feature is to the model’s decision. For instance, in the visual perception domain, Zhou et al. [99] first use class activation map as a saliency map for revealing important regions in the input image, as shown in Fig. 1.2a. Later, Lundberg et al. [53] proposed a



(a) Class activation map from the work by Zhou et al. [99]



(b) SHAP value saliency maps from the work by Lunderg et al. [53]



(c) Super-pixels saliency maps from the work by Ribeiro et al. [64]

Figure 1.2: Visual examples of saliency map in literatures.

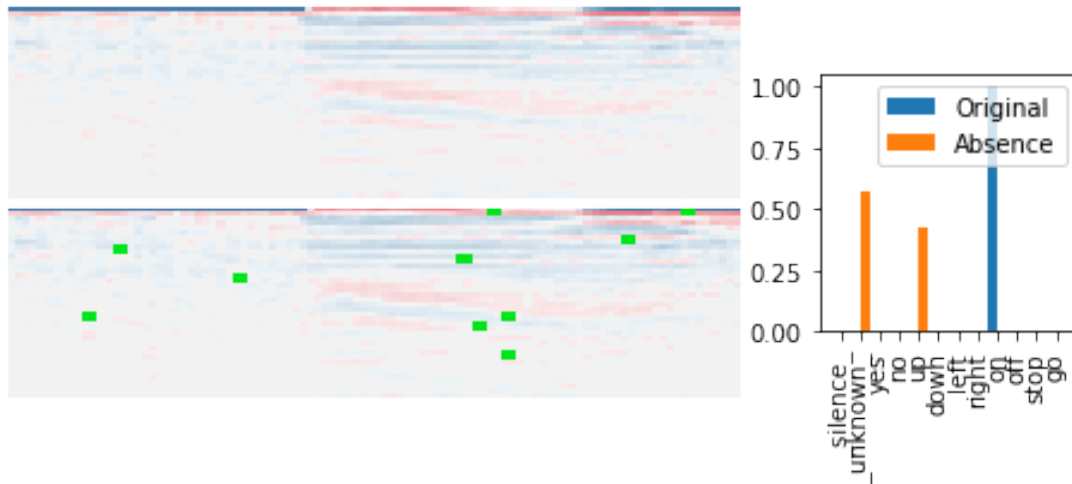


Figure 1.3: Example of a decision change due to the absence of critical regions in the decision-making process. (top-left) original MFCC representation of an utterance audio signal; (bottom-left) identified a critical region in MFCC; (right): prediction confidences for the decision made with original MFCC and with the absence of critical region.

unified framework for generating post-hoc local explanations using Shapley values, a classic approach from cooperative game theory, to estimate the importance of each input pixels (Fig. 1.2b). However, the raw features are not always the best choice of explanation. A successful XAI explanation must provide interpretable meaning to people. Explaining an image classification prediction in terms of individual input pixels can result in explanations that are too noisy, too expensive to compute, and more importantly, difficult to interpret. Alternatively, Ribeiro et al. [64] proposed to rely on contiguous patches of similar pixels (aka super-pixels), a more interpretable representation of image features, in the case of image classification prediction (Fig. 1.2c).

The concept of input attribution can also be extended to both speech recognition and NLP domain. Before presenting the explanation system for speech recognition and NLP, it is important to distinguish between input features and interpretable data representations. As mentioned before, interpretable explanations need to use a representation that is understandable to humans, regardless of the actual features used by the model. As for speech recognition systems, the input feature is the raw audio data. Relying solely on this form of data representation is difficult to analyze any meaningful information. Inspired by the traditional audio analysis techniques, the saliency map for speech recognition models can be interpreted as assigning importance factor to both time-domain features and frequency

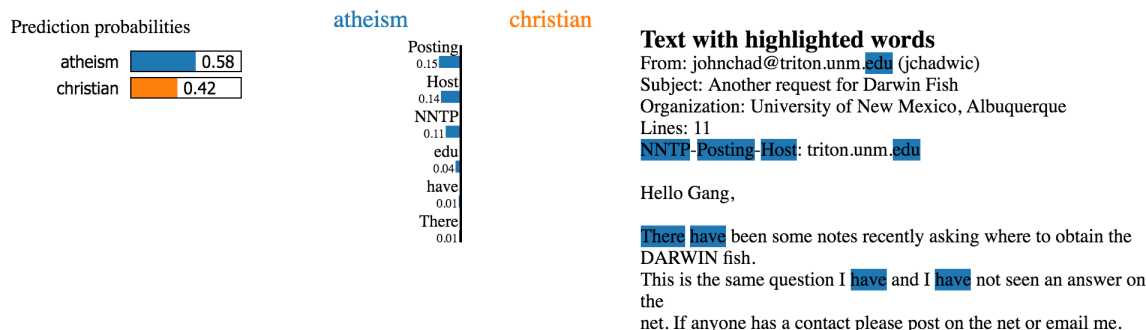


Figure 1.4: Saliency explanation for a binary text classifier [64]

domain features (Fig. 1.3). As for NLP models, a possible interpretable representation for text classification is a binary vector indicating the presence or absence of a word, even though the classifier may use more complex (and incomprehensible) features such as word embeddings. However, this choice of interpretable representation will have an inherent drawback. While the underlying model can be treated as a black-box, this binary feature representations will not be powerful enough to explain certain behaviours. For example, without the global context, a single positive or negative word can not conclude the overall sentiment of a sentence in the case of sarcasm. Thus, the saliency map for the NLP domain can be interpreted as assigning importance factors to each word in an input document. Fig. 1.4 shows a saliency explanation for a binary text classifier from the work by Ribeiro et al. [64].

1.2 Thesis Overview

1.2.1 Motivation

A constantly increasing number of real-world applications and systems have been powered by deep learning. Within these applications, many deep learning empowered applications are life crucial, raising great concerns in the field of safety and security. Due to the lack of interpretability and transparency in DNN, there has been a considerable surge of research interests in DNN explainability methods for shedding light into the “black box” nature of the DNN. Many explainability algorithms manifest their interpretation in a form of input attribution [99, 72, 75, 53, 17, 64]. These explainability methods aims to understand what features are important helps improve deep learning models, and build trust in the

model prediction and isolates undesirable behaviour. Furthermore, there are many open source and commercial explainability toolkit mainly featuring input attribution algorithms for understanding machine learning models [13, 5]. Due to the wide adoption of input attribution algorithms, we focus primarily on input attribution explainability methods and use explainability methods and input attribution explainability interchangeably in the following study.

While saliency map help researchers gaining new insight about DNN’s decision-making process, much of the evaluation for explainability methods have been largely subjective, where the produced explanation is up to the interpretation of the user. Ironically, no quantitative assessment is carried out to ensure neither correctness nor coherency of the explainability methods. This is largely attributed to the fact that it is challenging to evaluate whether an explanation of model behaviour is reliable. First, there is no ground truth. If what was important to the model for making a decision, we would not need to estimate feature importance in the first place. Second, it is unclear which of the numerous proposed interpretability methods that estimate feature importance one should select [99, 72, 75, 53, 17, 64]. Many feature importance estimators have interesting theoretical properties (e.g. preservation of relevance [6] or implementation invariance [80]). However, even these methods need to be configured correctly [57, 80] and it has been shown that using the wrong configuration can easily render them ineffective [42]. Furthermore, the subjectiveness around the current evaluation approach makes it difficult to judge and compare between different explainability methods. Last, through experiments, we found that tested explainability methods produced inconsistent, sometimes dramatically different, explanations for the same input and model. As such, it raises new concern that whether the identified saliency map is reflective of what DNN is leveraging for its decisions. These concerns and difficulty hinders the level of human trust in not just the DNNs themselves but also in the explainability methods as well. Therefore, an objective and machine-centric evaluation strategy is needed for both assessing the quality of the produced explanation and comparing the performance of different explainability methods.

1.2.2 Contributions

In this study, we propose two quantitative assessment metrics for evaluating the performance of explainability methods, namely **Impact Score** and **Impact Coverage**. Impact Score aims to evaluate whether the produced explanation is reflective of the DNN’s decision-making dependence in the input features. This is done via the notion of decision-making impact analysis. More specifically, **Impact Score** quantifies the level of impact over a DNN model’s decisions and confidences in the absence of the critical factors identified by

an explainability method. We also wish to assess the performance of explainability methods with the presence of directed erroneous decisions (e.g., under adversarial distractions). Under the directed erroneous scenarios, the critical factors are largely known, as the directed distraction is the casual of the wrongful decision. As such, we introduce **Impact Coverage**, which quantifies the coverage of the identified critical factors on the adversarial impacting factors. Based on these metrics, we conduct a comprehensive analysis of the performance of three different state-of-the-art explainability methods from the recent research literature on the three tasks across different domains, namely image classification, speech utterance classification, and sentiment classification. The explainability methods tested in our experiments are LIME [64], SHAP [53] and Expected Gradient (EG) [17]. Through experiments, we observe that the explanation produced by the explainability methods does not fully reflect the critical factors deemed by the neural network during its decision-making process. The empirical evidence also demonstrates that no single explainability methods can produce steady better results across different task domains and different test scenarios.

The major contributions of this thesis are:

- assessing the performance of explainability methods via the notion of decision impact analysis;
- assessing the performance of explainability methods under directed erroneous scenario by leveraging adversarial attack;
- quantitatively comparing the performance of the state-of-the-art explainability methods, such as LIME [64], SHAP [53] and Expected Gradient (EG) [17], through experiment.

1.2.3 Outline

The rest of the thesis proceeds as follows: Chapter 2 describes the background theory on deep learning in particular convolutional neural network(CNN) that is the main algorithm this work aims to gain more insight on. Furthermore, different CNN architectures and feature pre-processing techniques are elaborated in detail for classification tasks with different data modalities, such as image, audio, and natural language text. It is essential to understand the differences between task domains, as it improves the model accuracy and, more importantly, ensures the correct configuration and usage of explainability methods presented in this work. Chapter 3 first discusses two main strategies in current explainability methods researches: proxy approach and direct approach. Later, we illustrate the

high-level concepts of three explainability methods tested in this work and the reason behind choosing them. In the last section of Chapter 3, we compare proposed quantification metrics with other metrics presented in the recent literature. Chap 4 puts forward in detail the proposed quantification metrics, namely **Impact Score** and **Impact Coverage**. These proposed metrics aim to shed light on the behaviours of explainability methods under both general and erroneous scenarios. Experiment setting and results are presented in Chapter 5 along with some discussion on the obtained results. Finally, we conclude with Chapter 6 where a summary and insights of this work are presented.

Chapter 2

Background

2.1 Deep Learning

Deep learning is part of a boarder family of machine learning methods based largely on artificial neural networks. The artificial neural network, which was discovered first in the late '80s and early '90s, is based on a set of algorithms and mathematical models that try to learn high-level abstractions representation in data using multiple layers of non-linear transformation.

In recent years, deep learning [49] has been widely adapted to many different problems, such as image classification [45], speech recognition [28] and natural language processing [56], and has demonstrated state-of-the-art results for these problems. Apart from the fact that the design of deep learning architectures was initially inspired by the nervous system of humans, most of the success in the recent surge of deep learning architectures have also attributed to

- the advancement of computation power, such as usage of hardware accelerators(GPUs) for training,
- large scale of public available dataset, such ImageNet [45] and LibriSpeech [59],
- the availability of open-source deep learning frameworks, such as TensorFlow [1] and PyTorch [60].

Different deep learning models have been proposed using both, supervised and unsupervised approaches for learning high-level abstraction from given data. To name a

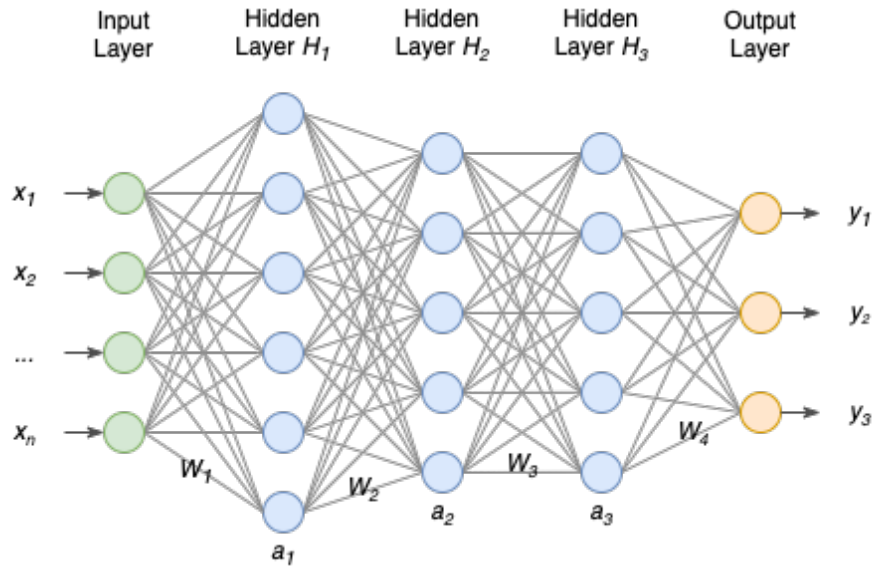


Figure 2.1: An illustration of a deep learning neural network

few deep learning models, these include convolutional neural networks (CNN), deep belief networks, autoencoders, recurrent neural networks (RNNs) and Restricted Boltzmann Machine (RBM). Despite the difference in model architectures, these deep learning models share the same underlying pipeline, as shown in Fig. 2.1. Recent studies [49, 27, 12] have shown that CNNs are the best architectures to perform recognition and classification tasks. This thesis studies the performance of explainability algorithms for CNNs.

2.1.1 Neural Networks

For conventional machine learning algorithms, it is difficult to extract well-represented features due to limitations such as the curse of dimensionality [8], computational bottleneck [77], and requirement of the domain and expert knowledge. The neural network is a type of deep learning method that is capable of learning useful patterns from raw data without explicit programming. It solves the problem of representation by building multiple simple features to represent sophisticated high-level concepts. For example, a neural network image classifier represents an object by describing edges, fabrics, and structures in its low-level hidden layers. A neural network is composed of a set of layers, where each network layer is a set of perceptrons (artificial neurons), as shown in Fig. 2.1. Each perceptron maps a set of input signals to output values with an activation function. The

function of a neural network is formed in a chain:

$$f(x) = f^{(k)}(f^{(k-1)}(\dots f^{(2)}(f^{(1)}(x)))) \tag{2.1}$$

2.2 Convolutional Neural Networks (CNNs)

Convolutional neural networks (CNNs or ConvNets) are feed-forward neural networks that are biologically-inspired variants of the multi-layer perceptrons (MLPs) with learnable weights and biases. Unlike the MLP, of which the neurons between layers are densely connected, the connection between different layers of CNNs is locally-connected to sub-regions of the previous layer. These sub-regions, denoted as receptive fields, are titled to cover the whole input spatial dimension. CNNs learn these sub-regions filters i.e., weights of the filters over the input space.

2.2.1 Convolutional Layers

Most of the commonly used complex CNN [45, 74, 81, 27] can be constituted by stacking convolutional, pooling and fully-connected layers together. Among these three types of layers, the convolutional (Conv) layer is the core building block of a CNN. The Conv layer consists of a set of learnable filters. Each filter is small spatially but extends through the full depth of the input volume. The number of filter spatial dimensions can be an arbitrary positive integer, but only the 1D, 2D and 3D Conv layers are commonly used. Without loss of generality, we denote a $(D + 1)$ -dimension input tensor as $I \in \mathbb{R}^{S \times C_{in}}$, where C_{in} is the number of channels, and $S \in \mathbb{R}^{[1 \dots D]}$ is the general representation of spatial dimensions. The filter of a D -dimension Conv layer is denoted as $W \in \mathbb{R}^{C_{in} \times K \times C_{out}}$, where C_{in} is the number of input channels, K is the kernel spatial size, and C_{out} is the number of filters. Here, some spatial dimension is omitted for a clearer presentation. The (i, j) element of a output tensor, $O \in \mathbb{R}^{S \times C_{in}}$, of a Conv layer can be computed with a corresponding bias b as follows:

$$O_{i,j} = I * W = \sum_m^{C_{in}} \sum_n^K I_{i,m} \times W_{m,n,j} + b \tag{2.2}$$

2.3 Different Data Modalities Classification

With rapid progress and significant successes in a wide spectrum of applications, deep learning models have demonstrated its efficacy on different data modalities [49, 45, 28, 56],

such as image, audio, text, etc. One of the key factors for such success is attributed to the expressiveness of deep learning models. Extreme expressiveness enables fitting a wide range of complex functions and extracting of high-level abstraction from data. Despite the theoretically identical expressiveness of neural networks, It has been demonstrated by empirical evidence that vanilla neural network (MLP) does not generalize well in high dimensional settings. This is arguably attributed to the overfitting issue. By incorporating priors about different data modalities, various neural network architectures, such as CNN, recurrent neural networks (RNNs) and attention neural networks, have been successfully applied on image [44, 27], audio [28, 25], and text [86, 15] data. While RNNs or attention neural networks may achieve the state-of-the-art performance on audio and text data, CNN-based architecture has not fallen out of favours given its simplicity and inference speed. What’s more, recent works [84, 41, 50, 98, 88, 55, 37, 87] have shown that CNN-based network architectures can achieve comparable performance on audio and text data. In this work, for a concise purpose, we will focus on the performance of explainability methods on CNN-based network architectures.

Despite the powerful expressiveness of CNNs, there is no one-fits-all model solution for different data modalities, since different data modalities are represented in fundamentally different formats. To be more concrete, a colour image is represented as 3-dimension volume; a PCA encoded audio is represented as a 1-dimension vector; a natural language text is represented as a string. Given these differences, unique data processing and modelling techniques have been proposed leveraging domain knowledge. These techniques have further advanced the frontier of deep learning model performance. In the remainder of this section, we will discuss different modality data processing techniques and their corresponding CNN architectures from past literature.

2.3.1 Visual Image Classification

In ILSVRC 2012 [69], Krizhevsky et al. [45] demonstrated the exceptional performance of a CNN architecture, namely AlexNet. Many different network architectures, such as VGGNet[74], InceptionNet [81], ResNet [27], ResNeXt [92] and SENet [31] have been proposed and continue to improve the performance of CNNs on image classification task. Among these network architectures, ResNet [27] has gained more traction because of its implementation simplicity, training stability and generalization ability.

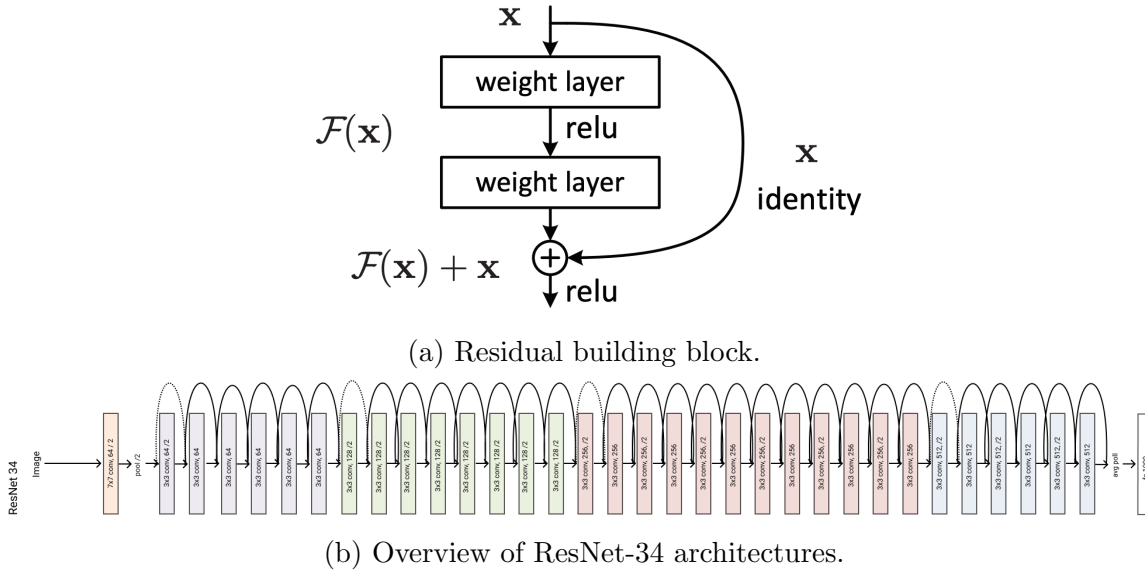


Figure 2.2: Residual connection and ResNet architecture [27].

CNN Architecture

The key idea from ResNet, by He et al. [27], is the concept of residual learning. Empirical evidence suggests that additional layers in deep CNNs cannot be merely “tacked on” to shallower networks. Specifically, He et al. proposed that it may be easier to learn the residual $H(x) = F(x) + x$ instead of the true mapping $F(x)$, since it is empirically easier for propagating first-order gradients back to shallow layers of deep CNNs when Backprop [68] optimization is applied. Recent literatures[51, 97, 26] have also shown theoretical evidence supporting the training stability and generalization ability of residual learning. In ResNet, residuals are expressed via connections between layers, shown in Fig. 2.2a, where an input x to a weight layer i is added to the output of some downstream weight layer $i+k$, enforcing the residual definition $H(x) = F(x) + x$. In this work, we leverage the ResNet-34 architecture (Fig. 2.2b) from the work by He et al. [27]. For the concise reason, the explainability methods are tested on the same ResNet-34 architecture for the image classification task.

2.3.2 Speech Utterance Classification

Limited-vocabulary speech recognition [89], also known as keyword spotting (KWS), has recently attracted much interest as an important application of voice-activation system

(e.g. "ok google" or "hey siri") for mobile, IoT, and other embedding devices. The primary goal of KWS is to detect a relatively small set of predefined keywords in a stream of user utterances. Such capability can enable voice interfaces with which the user can interact in a natural, verbal manner. This fully hand-free interface is ideal as a complement for full automatic speech recognition, which is typically performed in the cloud.

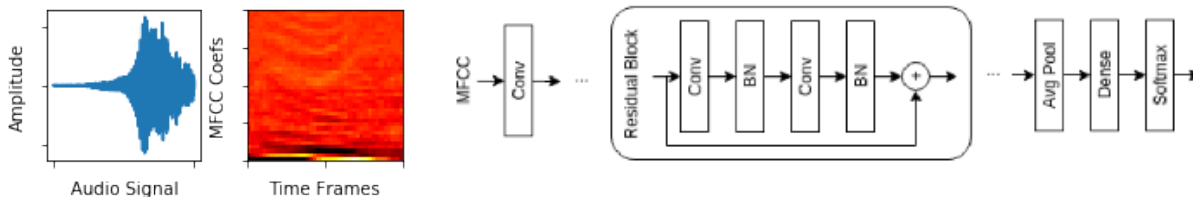


Figure 2.3: Speech utterance network architecture. (left) Audio signal; (middle) MFCC representation; (right) res15 architecture with its residual block [52].

Feature Extraction and Input Pre-processing

Based on the past literature [28], a very effective strategy for leveraging deep neural networks for limited vocabulary speech recognition is to first transform the input audio signal into Mel-Frequency Cepstrum Coefficient (MFCC) representations (see Fig. 2.3). For reducing audio signal noise, a band-pass filter of 20Hz/4kHz is applied to the input audio. Inspired by [70], the input feature is forty-dimensional MFCC frames stacked using a 30ms window and 10ms frameshift. Substantially, the MFCC representation of the audio signal is used as the input to the deep CNNs.

CNN Architecture

In this study, we leveraged the deep residual network architecture proposed by Tang et al. [84], which they refer to as res15 and was shown to provide state-of-the-art accuracy when it was first published. In particular, Tang et al. proposed to use a residual block architecture where the first layer of the block is a bias-free convolutional layer with weights $W \in \mathbb{R}^{(m \times r) \times (n_{i-1} \times n_i)}$, where m and r are the width and height of convolutional kernel, and n_{i-1} and n_i are the number of channels for the previous convolutional layer and the current convolutional layer, respectively. After the convolutional layer, a ReLU activation and batch normalization [33] is appended in the residual block. In addition, convolutional dilation, (d_w, d_h) , is used to increase the receptive field of the network. Increasing reception

fields in deeper layers allow the network to consider the input entirely without the need for very deep layers. Fig. 2.3 (right) shows the overall architecture and the detail of one of the residual blocks.

2.3.3 Text Sentiment Classification

Text sentiment classification is the interpretation of emotions within text data using natural language processing (NLP) techniques, allowing businesses to identify customer sentiment in online feedback. Within NLP, much of the work with deep learning approaches have involved learning word vector representations through neural language models [7, 94, 55, 86, 15, 62] and performing composition over the learned word vectors for classification [14]. In the recent development of NLP, self-supervised pre-trained deep learning models such as BERT [15] and GPT [62] have demonstrated dominating performance across all NLP tasks including sentiment classification. However, the dynamic and complex nature of these models has been preventing their wide adaption. What’s more, explainability methods in recent years were not proposed for these gigantic models (e.g. GPT-2, a successor to GPT, has 1.5 billion parameters). The correctness of explainability methods on these models was not validated. Thus, we, in this study, focus solely on the CNN-based models, specifically the work by Kim et al. [41], excluding the aforementioned pre-trained models.

Feature Extraction: Word2Vec [55]

Word vectors, wherein words are projected from a sparse, 1-of- V one-hot encoding (where V is the vocabulary size) onto a lower-dimensional vector space are essentially feature extractors that encode semantic features of words in their dimensions. In such dense representations, semantically close words are likewise close, in terms of Euclidean or cosine distance, in the lower dimensional vector space. Specifically, Word2vec [55], inspired by the Skip-gram model [54], uses shallow neural networks for learning distributed representation of words in a vector space from large amounts of unstructured text data. Somewhat surprisingly, the learned word representations display interesting properties:

- **Semantic Similarity:** Word2vec vectors implicitly encode semantic similarities of discrete words. Measuring cosine similarity, semantic similar words are grouped closely in the vector space. Table 2.1 shows the top nine words closest, in terms of cosine distance proximity, to the word “Sweden”.

Table 2.1: List of words associated with “Sweden” using Word2Vec [55], in order of proximity. The nations of Scandinavia and several wealthy northern European, Germanic countries are the closest nine words.

Word: Sweden	Cosine Distance
Norway	0.760124
Denmark	0.715460
Finland	0.620022
Switzerland	0.588132
Belgium	0.585835
Netherlands	0.574631
Iceland	0.562368
Estonia	0.547621
Slovenia	0.531408

- **Linear Translation:** Word2vec vectors explicitly encode linguistic regularities and patterns. Many of these patterns can be represented as linear translations. For example, the result of a vector calculation $vec(\text{“Madrid”}) - vec(\text{“Spain”}) + vec(\text{“France”})$ is closer to $vec(\text{“Paris”})$ than any other word.

Word2Vec [55] is similar to an autoencoder, encoding each word in a vector, but rather than training with the reconstruction objective, Word2Vec [55] trains words against words that neighbour them in the corpus. The intuition behind is that the semantic meaning of a word can be implicitly inferred by its surrounding context in a large corpus. This prior can be done in one of the two ways, either using context to predict a target word, which is known as the continuous bag of words (CBOW), or using a word to predict a target context, which is proposed in Skip-gram model [54]. For better illustration, Fig 2.4 shows the high-level training objective of CBOW and Skip-gram [54]. Mikolov et al. [54] demonstrate that Skip-gram can learn more efficiently and produce more accurate results. Thus, we will use Word2Vec [55], a successor of Skip-gram, as our feature extraction for the text sentiment classification task in this study.

The training objective of the Skip-gram model [54] is to find word representations that are useful for predicting the surrounding words in a sentence or a document. More formally, given a sequence of training words $w_1, w_2, w_3, \dots, w_T$, the training objective is to maximize the average log probability

$$\frac{1}{T} \sum_{t=1}^T \sum_{-c \leq j \leq c, j \neq 0} \log p(w_{t+j} | w_t) \quad (2.3)$$

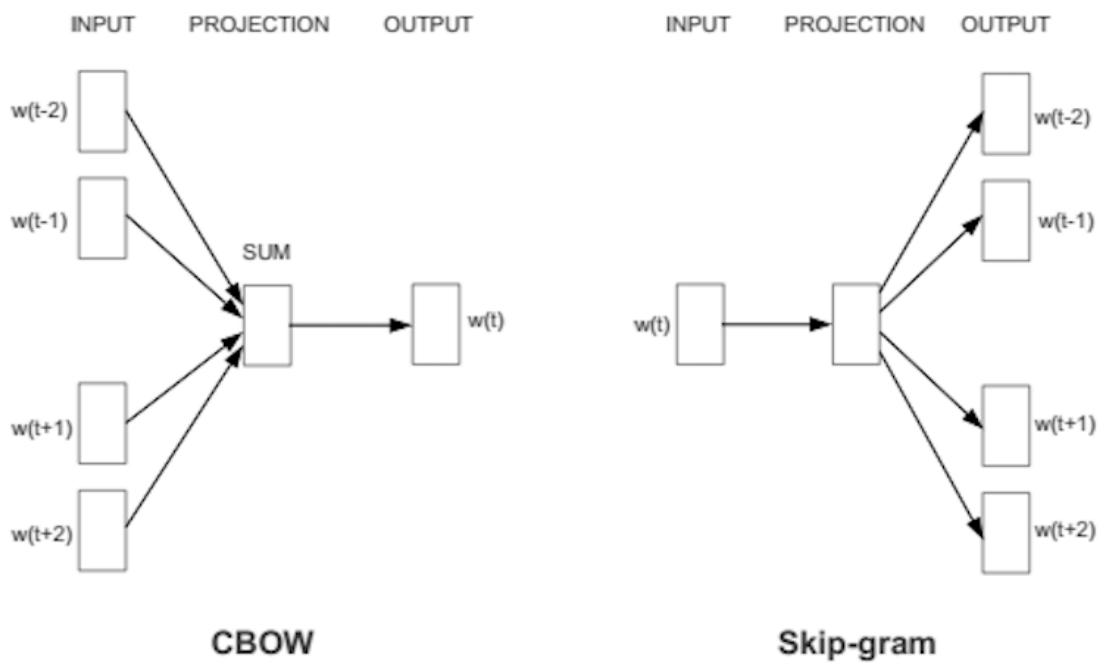


Figure 2.4: Comparison between CBOW and Skip-gram [54]. CBOW (left) predicts the current word based on the context and Skip-gram [54] (right) predicts surrounding words given the current words.

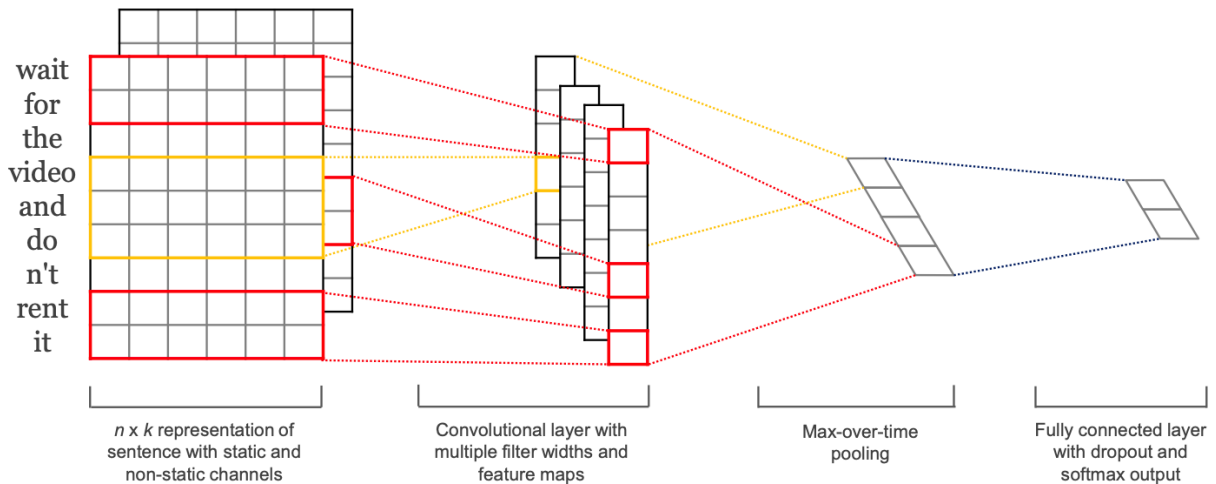


Figure 2.5: The text sentiment classification network architecture from the work by Kim et al. [41].

where c is the size of the training context. Larger c results in more training examples and thus can potentially lead to higher performance, at the expense of the training time.

CNN Architecture

In the work by Kim et al. [41], the authors show that a simple 1D CNN on top of Word2vec [55] word vectors obtained from an unsupervised neural language model can achieve comparable performance on text sentiment classification task. Let $I_m \in \mathbb{R}^K$ be the K -dimensional word vector corresponding to the m -th word in the sentence. A sentence of length N (padded with *zeros* where necessary) is concatenated sequentially forming an input matrix $I_{1:N} \in \mathbb{R}^{N \times K}$ as following

$$I_{1:N} = I_1 \oplus I_2 \oplus \dots \oplus I_N \quad (2.4)$$

where \oplus is the concatenation operator.

A 1D convolution operation involving a filter $W \in \mathbb{R}^{H \times K}$ is applied to a window of H words to produce a new feature. An output feature o_m is formally computed from a window of words $I_{m:m+h-1}$ as following

$$o_m = \sigma(W \cdot I_{m:m+h-1} + b) \quad (2.5)$$

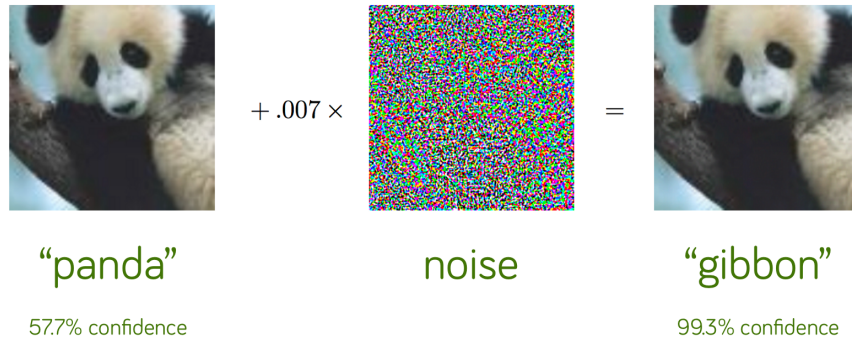


Figure 2.6: A demonstration of adversarial example applied to a deep learning model. By adding an imperceptibly small vector, the classification result is changed [24].

where σ is a non-linear activation function and $b \in \mathbb{R}$ is a bias term. V numbers of 1D convolutional filters are applied across the sentence length dimension to produce a feature map, $O_{1:N} \in \mathbb{R}^{N \times V}$,

$$\begin{aligned}
 O_{1:N} &= O_1 \oplus O_2 \oplus \dots \oplus O_N \\
 \text{where } O_i &= [o_{i,0}, o_{i,1}, \dots, o_{i,V}]
 \end{aligned}
 \tag{2.6}$$

After the 1D Conv layer, a max-over-time pooling operation [14] is applied over the feature map and take the maximum value $\hat{O} = \max\{O\}$. The idea is to capture the most important feature for each feature map. This pooling scheme naturally deals with variable sentence lengths. The overall CNN architecture is shown in Fig. 2.5.

2.4 Adversarial Attack for Neural Network

Driven by the emergence of massive data and hardware acceleration, deep learning requires less hand-engineered features and expert knowledge. Despite great successes in numerous applications, recent studies [83, 24, 46, 73, 19, 23, 10] find that deep learning model is vulnerable against well-designed input samples. In the work by Szegedy et al. [83], the authors first generated small perturbations on the images and fooled a well-performed deep learning model with high probability. These misclassified samples were named as *Adversarial Examples*, as shown in Fig. 2.6 [24] where the deep learning model is fooled by an adversarial example to classify a “panda” as a “gibbon”. Not isolated to the image

classification task, the adversarial attack has been used to manipulate stop signs in a traffic sign recognition system, or remove pedestrians segmentations in an object recognition system. Furthermore, this line of work has also been extended to the speech recognition (ASR) and NLP models.

2.4.1 Adversarial Example Generation

Given a trained deep learning model M , and an original input data sample x , generating an adversarial example x' can generally be described as a box-constrained optimization problem:

$$\begin{aligned} \min_{x'} & \|x' - x\| \\ \text{s.t.} & M(x') = y', \\ & M(x) = y, \\ & y' \neq y \end{aligned} \tag{2.7}$$

where y and y' denote the output label of x and x' , $\|\cdot\|$ denotes the custom constraint (e.g. the L2 distance) between two data sample.

Chapter 3

Explainability Methods and Assessments

3.1 Explainability methods

The explainability methods in current research literature can generally be divided into two main categories [85]. In the first category of explainability methods, which we will refer to as **proxy** strategies [64, 53], a deep neural network is approximated by a proxy model and the decision-making of the deep neural network is interpreted by querying the proxy model. In the second category, which we will refer to as **direct** strategies [80, 76, 72, 75, 17, 91] the decision-making process of a deep neural network is mainly interpreted by studying the internal behaviour within a deep neural network directly and then surfacing that information as an explanation for the decision-making process of the network. The most well-known of the proxy method is LIME [64], which takes advantage of a linear proxy model to approximate the behavioural of the targeted machine learning model and then interprets the original model based on the learnt proxy. Proxy approaches are considered as “black box” approaches where the explainability method does not have direct access to the inner workings of the network and the proxy model approximates it given the input and the output to the network. On the other hand, direct explainability algorithms are usually considered as “white box” methods as they require access to the inner workings of a deep neural network such as gradients and activations at different layers for a given input to identify the key factors within the input that is critical to the decision-making process. For example, by leveraging information about gradients, it is possible to quantify how much change in the input data would turn the decision of the network to another

output and as such measure the importance of each input in the decision-making process. Notable gradient-based direct explainability approaches include Integrated Gradient [80], Guided Backpropagation [76], Guided GradCAM [72], SmoothGrad [75] and Expected Gradients [17].

3.1.1 Local Interpretable Model-agnostic Explanations (LIME)

Local Interpretable Model-agnostic Explanations (LIME) [64] is a proxy method that leverages the learning of local surrogate models to explain deep neural networks. In this approach, a surrogate model is used to approximate the underlying behavioural of the deep neural network. The deep neural network is probed and the surrogate model is trained based on the prediction outputs of the deep learning model. Different permutation of samples is generated and a new dataset is constructed based on the generated samples and the corresponding predictions of the deep neural network. Then, an interpretable model (i.e., surrogate model) is used and is trained by the generated dataset. As such, the training process can be formulated as follows:

$$\mathcal{I}(x) = \underset{g \in G}{\operatorname{argmin}} \mathcal{L}(f, g, \theta) + \Gamma(g) \quad (3.1)$$

where $\mathcal{I}(x)$ encodes the explanation for instance x via the optimal function g in the possible set of function G . f represents the deep neural network, $\Gamma(\cdot)$ identifies the complexity of the function g to be used in the explainability process, and L is the loss function which measures the similarity of surrogate model and the original model (i.e., here the deep neural network). In this study, we leverage ridge regression models as the surrogate models.

3.1.2 SHapley Additive exPlanations (SHAP)

SHapley Additive exPlanations (SHAP) [53] is an explainability method that measures the importance of each feature for a particular prediction. This algorithm belongs to the family of additive feature attribution methods, where the explanation is expressed as a linear function of features. To do so, SHAP replaces each feature x_i in the model with a random variable y to determine whether the feature x_i is present or not:

$$g(y) = \gamma_0 + \sum_{i=1}^m \gamma_i \cdot y_i \quad (3.2)$$

where $g(y)$ is a local surrogate model of the original model and γ_i encodes how much the presence of feature i contributes to the final output. γ_i is calculated based on the difference to the output of the original model made by including the feature i for all the combinations of features other than i .

3.1.3 Expected Gradients (EG)

Expected gradient (EG) algorithm is an extension of integrated gradient method [80] with lower hyper-parameters. Being a method in the family of feature attribution algorithms [71, 80], this method like other approaches in this family tries to identify the difference between a model’s current prediction given the changes in the input and the prediction based on the baseline input. While the baseline input is application dependent in the integrated gradient method [80] and usually is a black image where all pixels are zero, the expected gradient approach addresses this by defining the value of a feature based on integrating over the interested dataset. Therefore, the expected gradients can be expressed as follows:

$$G_i(x) = \mathbf{E}_{x' \sim D, \alpha \sim U(0,1)} \left[(x_i - x'_i) \times \frac{\delta f(x' - \alpha(x - x'))}{\delta x_i} \right] \quad (3.3)$$

where x' is the sample drawn from dataset D and α is sampled from uniform distribution, x represents the target input, E is the expectation, x' is baseline input. By doing so, it does not need to specify the baseline x' since the expected value is calculated based on the sampling trick.

3.1.4 Choices of Explainability methods

The field of XAI has seen a resurgence in recent years. There are many more recent and interesting explainability methods being proposed. However, given the scope of this study, we only experimented with three of the exiting explainability methods considering their impactness, representativeness and accessibility of open source code. We chose LIME as the reference baseline, because of its popularity and simplicity. There are many open-source and commercial explainability toolkit featuring LIME as one of the algorithms for understanding machine learning models [13, 5]. As for SHAP rather than other methods like LRP, besides its popularity [13, 5], past literature [53] has shown that SHAP generalizes several explainability methods including LRP, thus making it a good choice to represent many methods that it generalizes. According to the work by Erion et al. [18], Expected-Gradient outperforms other gradient-based explainability methods such as GradCAM and

Integrated-Gradient. As such, we chose these explainability methods, to provide a fair representation of the different existing explainability methods. Lastly, it is a time-consuming and error-probing task to implement an explainability method from literature. Thus, we chose the aforementioned three explainability methods since their code, from their original authors, are publicly available. As one of the directions for future work, we would like to include more explainability methods in the future.

3.2 Explainability Assessment in Literatures

Understanding what input feature is important helps researchers gaining new insight into the models’ decision-making process, builds trust in the model prediction and isolates undesirable behaviours. Also, many of the explainability methods proposed in recent literature demonstrate interesting theoretical properties, such as preservation of relevance [6] and implementation invariance [80]. Unfortunately, it is challenging to evaluate whether an explanation of model behaviours is trustworthy. Unlike the machine learning process building process, where the “ground-truth” data is provided, which is important to a model’s decision-making is completely unknown. Due to this reason, we see a gap between the current explainability methods and their assessment. This gap hinders the level of human trust in not just the deep neural networks but also in the explainability methods themselves. In fact, quantitative methods to assess the performance of explainability methods is critical to not only trust in decisions made but also in the choice of method for deployment and research development, especially since different explainability methods can produce drastically different explanations given the same input data and same model and so it is difficult to know if algorithmic extensions on such explainability approaches actually improves interpretability. In this section, we will discuss different explainability assessments in the literature.

3.2.1 Human Assessment

Since there is no clear way to measure “correctness”, much of the evaluation for explainability methods have been largely subjective, where the produced explanation is up to the interpretation of the user. It is most common that comparing the relative merit of different explainability methods is based upon human studies [72, 66, 47, 48] which interrogate whether the ranking is meaningful to a human. In the work by Selvaraju et al.[72], the authors proposed to evaluate the class discrimination ability of the saliency map as the proxy evaluation metric for explainability assessment. The class discrimination ability is based

What do you see?



Your options:

- Horse
- Person

Figure 3.1: AMT interfaces for evaluating class discrimination ability from the work by Selvaraju et al. [72]

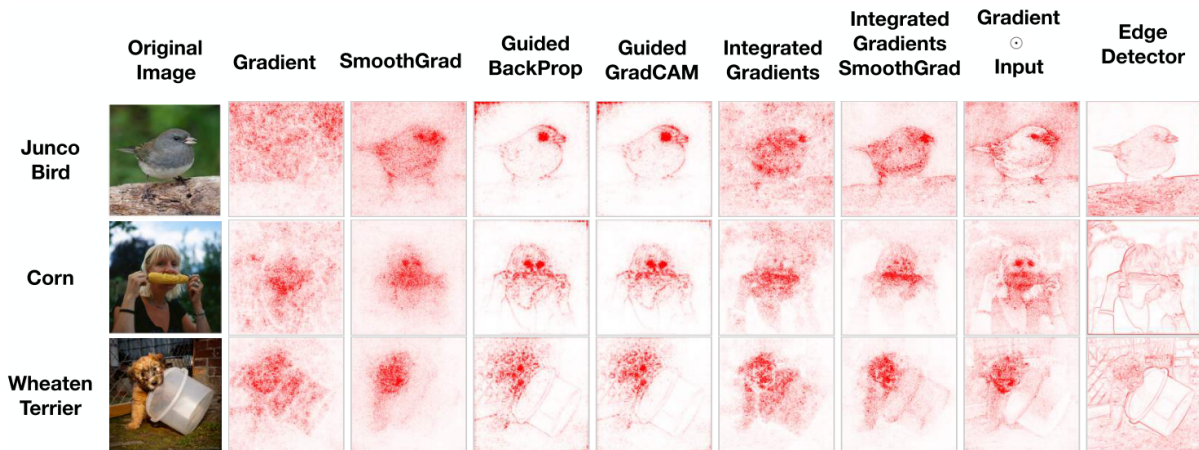


Figure 3.2: Saliency maps for some common methods compared to an edge detector from the work by Adebayo et al. [2]. An edge detector produces outputs that are strikingly similar to the outputs of some saliency methods. In fact, edge detectors can also produce masks that highlight features which coincide with what appears to be relevant to a model's class prediction.

on a questionnaire asking human workers on Amazon Mechanical Turk (AMT): “Which of the two object categories is depicted in the image?”, as shown in Fig. 3.1. However, this evaluation metric can produce misleading results for inherently flawed explainability methods. As shown by Adebayo et al. [2], a simple edge detector can produce outputs that are strikingly similar to the outputs of some saliency methods (Fig. 3.2). In fact, edge detectors can also produce masks that highlight features that coincide with what appears to be relevant to a model’s class prediction.

Human visual attention has been extensively studied for decades not only in cognitive psychology and neuroscience [11], but also in computer vision community [35, 9]. This is because such a selective visual attention mechanism has an essential role in human perception. Inspired by the selective attention in the visual cortex, artificial attention is designed to the most task-relevant input signal. The notion of artificial attention shares a similar concept as the saliency map does, where both aim to highlight the critical region in the input signal. In the work by Lai et al. [48], the authors proposed to evaluate the level of alignment between machine saliency map and human attention. Arguably, measuring the level of alignment only considers the interpretability of the saliency map, but neglects the faithfulness of the saliency map. A saliency map that overlaps significantly with the human gaze area offers is easier to interpret, as human attention concentrates more on the task-relevant parts of a visual stimuli [34]. Yet, a good saliency explanation should be “faithful” to the model, where faithfulness is the ability to accurately explain the function learned by the model. Naturally, there exists a tradeoff between the interpretability and the faithfulness of an explanation: a perfectly interpretable explanation can have no correlation with the function learned by the model [2]. Using human attention reveals the interpretability of the explainability methods, but does not reflect the faithfulness of these methods.

Lastly, even though the human-in-the-loop evaluation offers an intuitively interpretable solution, besides the aforementioned issues, human-centric evaluation methods suffer from two major issues: subjectiveness and cost-inefficiency. The “correctness” of a saliency explanation may be interpreted differently by different end-users. As such, it is less reliable and more difficult for quantitative comparison between different explainability methods. Furthermore, an objective and trustworthy evaluation involves assessing a massive amount of explanation, but not a countable amount of examples. Due to the large amount required, having a human in the evaluation loop is subject to substantial cost in terms of both time and finance.

3.2.2 RemOve And Retrain (ROAR)

The RemOve And Retrain (ROAR) metric introduced by Hooker and et al. [30] re-trains the neural network with the feature-modified data for aligning the training and evaluation distributions and assess explainability performance. Our approach differs significantly from ROAR in several key ways. **Network modification:** while ROAR modifies the network weights through retraining during the assessment, the proposed method does not need to change the network parameters for the explainability purposes. We argue that by modifying the weights, the network has fundamentally very different behaviour and thus does not reflect the decisions made by the original network. As such, our approach serves to better explain the original network’s behaviour. **Assessment criteria based on feature removal:** both ROAR and the proposed Impact Score assess the performance based on feature removal techniques; however, ROAR assesses an explainability method based on model accuracy degradation via re-training, whereas the proposed metric, Impact Score, measures the performance by the level of decision impact. We argue that understanding what is critical to a model’s decision-making process gives a more direct assessment than indirectly assessing through model degradation via re-training. **Directed Injection Perturbation:** unlike ROAR, the proposed method further injects directed features for testing the explainability methods under the erroneous setting.

3.2.3 Deletion and Insertion

The deletion metric, proposed by Petsiuk et al. [61], measures the confidence changes when the identified feature is gradually removed. While our Impact Score metric might look similar to the deletion metric, the deletion metric only considers the change in class confidence. This solo dependence overlooks the decision level impact. For cases where the class confidences are close (e.g. fine-grain classification), the discriminative feature can be small and only changes the class confidence value by a small amount, but enough to shift the original decision. The deletion metric under-estimates the explainability methods’ ability for identifying such discriminative features. The proposed Impact Score metric considers both the confidence level impact and the decision level impact. Besides, our impact coverage measures the performance of explainability methods directly without the aims of proxy metrics, such as accuracy or confidence.

Chapter 4

Methodology

4.1 Quantifying Explainability

Much of research literature around explainability, particularly for visual perception tasks such as image classification, has revolved around the subjective visual interpretation of the explanations produced by the explainability method. This usually takes on the form of visual saliency maps, where salient regions in the map produced using the explainability method of choice are considered as critical regions influencing the decision made by a network. However, due to the purely qualitative nature of such visual assessments, it is very challenging to get a good sense as to how well an explainability method is performing, how useful or meaningful the provide an explanation is relative to its influence over the network’s decision and its associated confidence, and more importantly how well it performs compared to other explainability methods. As such, this can limit progress in the field of explainable artificial intelligence since there is no method of benchmarking based on subjective visual assessment.

More recently, there have been explorations into human-centric strategies for quantifying explainability performance in the case of visual perception, where the visual saliency map produced using a given explainability method for a given image is compared with a visual attention map created based on gaze information collected from human subjects [48]. While such an approach is a step towards the quantification of explanations produced by explainability methods, one of the biggest limitations of such an approach is the underlying assumption that a deep neural network makes decisions in a similar manner as human subjects, which is often not true. As such, this human-centric approach to quantifying explainability performance provides very little insight into the actual driving factors of

the decision-making process of deep neural networks. Furthermore, this approach requires considerable human gaze information to be collected, which is simply impractical for most real-world scenarios.

To address the limitations of human-centric strategies for quantifying the performance of explainability methods, we take a drastically different direction by instead exploring a more machine-centric strategy where we quantify performance based on the decision-making behaviour of the network itself. More specifically, we aim to quantify the performance of explainability methods on deep neural networks via the notion of decision-making impact analysis, where we instead study the quantitative impact of critical factors identified by an explainability method for a given decision made by a network based on the changes in decisions and associated confidences in the decisions of the network itself.

In the below sections, we will first define a performance metric for quantifying the impact of critical factors identified by an explainability method on decisions and the confidence in those decisions made by a given deep neural network. Next, we introduce an additional performance metric for directed erroneous decision scenarios based around the concept of impact coverage.

4.1.1 Assessing Impact on Decisions

In order to facilitate for the quantitative assessment of the performance of a given explainability method, the first step is to first define and formulate a performance metric for performing such an assessment. Motivated towards taking a machine-centric strategy to quantitative performance assessment of a given explainability method on a particular deep neural network, we aim to develop metrics that quantify the importance of critical factors identified by the explainability method for a given decision made by a network based on the impact these factors have over network decisions and the associated confidences. We consider the critical factors c identified by an explainability method M to be important to a decision y made by a deep neural network N for a given input x if either of the following conditions is met:

- **Decision-level impact:** The decision made by the deep neural network changes in the absence of critical factors.
- **Confidence-level impact:** The confidence of the deep neural network in its decision z changes by $\tau\%$ in the absence of the critical factors.

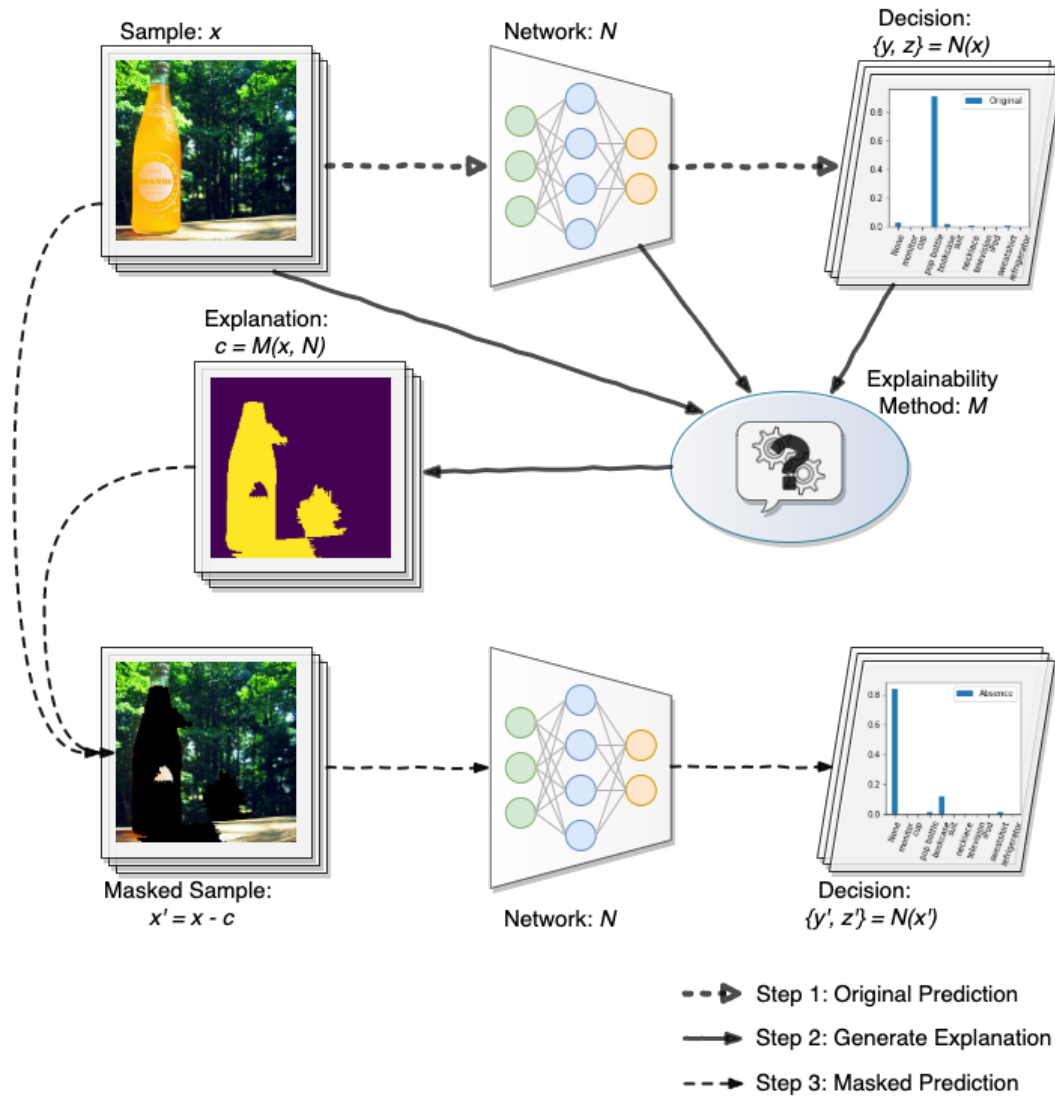


Figure 4.1: **Impact Score** flow diagram. Firstly, the prediction, $\{y, z\}$, of an input sample, x , is inferred with the target network, N . Secondly, the explanation, c , is computed with the testing explainability method, M . Thirdly, the prediction, $\{y', z'\}$, of critical-factor-masked sample, x' , is inferred with the same target network, N . Lastly, **Impact Score** is calculated with the original prediction, $\{y, z\}$, and the masked prediction, $\{y', z'\}$.

The motivation behind this definition of importance for critical factors as identified by a given explainability method is based on the idea that, if the critical factors are indeed

crucial to the decision-making process of the deep neural network, then the absence of these critical factors in the given input will have such an impact that the network behaves in a way that it would either be significantly less confident in its current decision, or so unconfident in its decision that its confidence in another decision is higher and thus leads the network to make a different decision altogether.

In this study, we formulate the performance metric I , which we will refer to as the Impact Score, as follows. Let the relationships between the critical factors c , explainability method M , the input x , the decision y , confidence in the decision z , and the network N be expressed by the following equations:

$$\{y, z\} = N(x), \quad (4.1)$$

$$c = M(x, N), \quad (4.2)$$

where $c \in x$. Based on this, we can define the input in the absence of c as identified by M as,

$$x' = x - c, \quad (4.3)$$

and the decision given x' as input into N as,

$$\{y', z'\} = N(x'). \quad (4.4)$$

Therefore, in the general scenario, based on the conditions defined above that the critical factors c for a given input x as identified by M must meet to be deemed as important, we can define the Impact Score I across a set of n inputs $X = \{x_1, x_2, \dots, x_n\}$ as:

$$I = \frac{1}{n} \sum_{i=1}^n ((y'_i \neq y_i) \vee (z'_i \leq \tau z_i)). \quad (4.5)$$

where i denotes the i^{th} input. In this study, we set $\tau = 0.5$ to indicate that the network has lost half of the confidence it had on its original decision. Finally, we also introduce a stricter variant of the above Impact Score, denoted by I_{strict} where we only consider decision-level impact:

$$I_{strict} = \frac{1}{n} \sum_{i=1}^n (y'_i \neq y_i). \quad (4.6)$$

4.1.2 Assessing Erroneous Coverage

In the scenario where we wish to study the impact in directed erroneous decisions (e.g., decisions made under the influence of adversarial examples), we introduce an additional approach to quantitatively assessing the performance of the different explainability methods since the critical factors that the network leverages to make a decision are largely known a priori to the evaluation (e.g., in the case of an adversarial patch, the critical region that is important to the decision-making process is the adversarial patch itself) More specifically, we can further quantify the importance of the identified critical factors c based the amount of coverage of the adversarially impacted factors in x by the critical factors c . Let us define the Impact Coverage metric $I_{coverage}$ across a set of n inputs $X = \{x_1, x_2, \dots, x_n\}$ based on the intersection-over-union between the adversarially impacted factors and the critical factors across the given set of inputs:

$$I_{coverage} = \frac{1}{n} \sum_{i=1}^n \frac{(a_i \vee c_i)}{(a_i \cup c_i)}. \quad (4.7)$$

where a_i is the adversarially impacted factors in input x_i . As such, the Impact Coverage metric is designed to be high when heavy overlapping between the identified critical factors and the adversarially impacted factors to reward strong alignment between the explanation produced by the explainability method and the actual factors impacting decision.

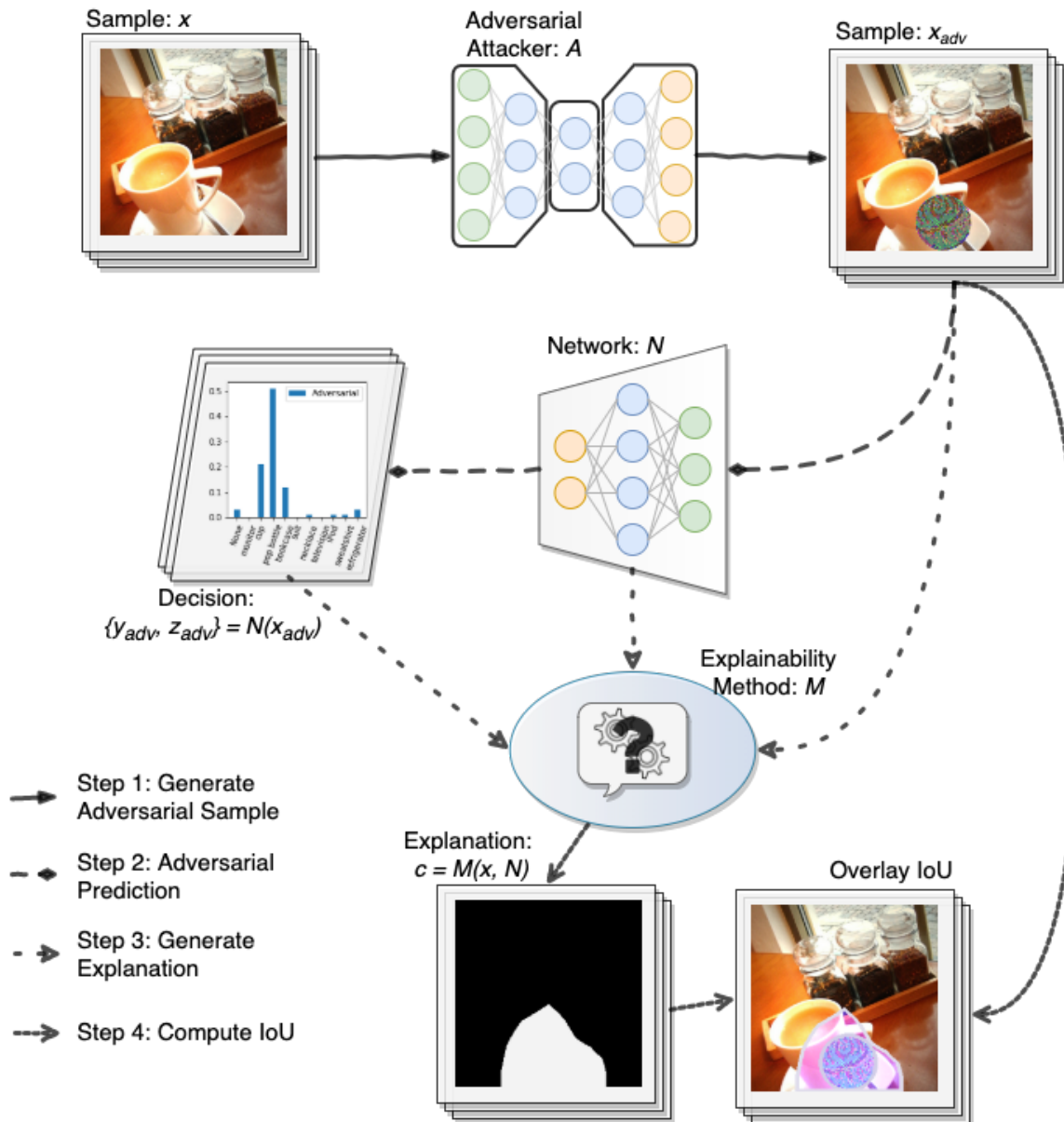


Figure 4.2: **Impact Coverage** flow diagram. Firstly, a input sample, x , is adversarially modified, x_{adv} , by an attacker, A . Secondly, the prediction, $\{y_{adv}, z_{adv}\}$, of the modified sample, x_{adv} , is inferred with the target network, N . Thirdly, the explanation, c , is computed with the testing explainability method, M . Lastly, **Impact Coverage** is computed by overlaying the adversarially modified set and the critical set.

Chapter 5

Experiments

5.1 Experimental Setup

The conducted experiments and the explainability methods used in this study are described below.

5.1.1 Experiment 1: General Scenario

For the first experiment, we quantitatively evaluate the performance of several state-of-the-art explainability methods using the two variants of Impact Score (i.e., I and I_{strict}). The purpose of this first experiment is the quantitatively evaluate explainability performance under a more general scenario where decisions are made on untampered data inputs and decisions are made by the network on such data inputs and is representative of the general use case.

Image Classification

For each explainability method M , a ResNet-50 deep convolutional neural network is designed for the task of image classification as the reference network N_{image} . A subset of the ImageNet [69] dataset is leveraged as input X . This experiment tasks the different explainability methods to identify critical regions within a natural image that is important to the class prediction made by the network. In order to test whether the identified region is truly reflective of the importance deemed by the network N_{image} , the I and I_{strict} are

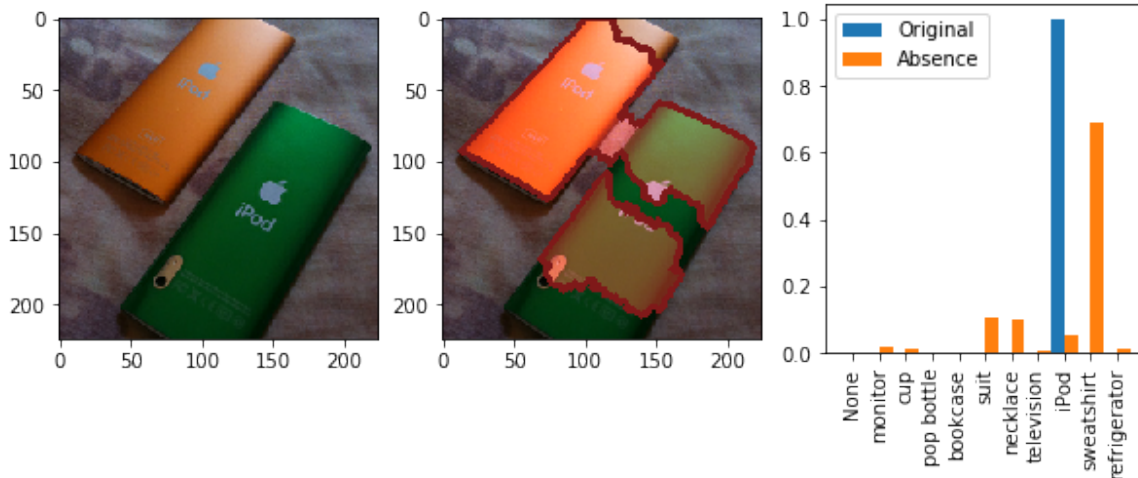


Figure 5.1: Example of a decision change due to the absence of critical regions in the decision-making process. (left) original image; (center) identified critical region; (right) prediction confidences for decisions made with the original image and with the absence of critical regions. The absence of critical regions led to a change in decision, which means the explanation reflects the impact on the decision.

computed across 410 different images from the ImageNet dataset, all of which are correctly classified by the network N_{image} . An example of a decision change that results from the absence of critical regions identified by an explainability method is shown in Fig. 5.1.

Utterance Classification

To evaluate the performance of each explainability methods in the speech recognition domain, a ResNet variant network by the work of Tang et al. [84] is leveraged for the limited vocabulary utterance classification task. More specifically, a ResNet-15 deep convolutional neural network is trained as the reference network $N_{utterance}$ using Speech Command dataset [89]. The audio signal feature is transformed into Mel-frequency cepstral coefficients (MFCC) as the input feature of the reference network $N_{utterance}$. The explainability methods are tasked to identify the critical region within the MFCC representation of an audio signal. Similar to the image classification task, the I and I_{strict} is evaluated across 820 correctly classified utterance audio samples in the absence of the identified critical region. Unlike the image classification task, the identified critical region is removed by setting the value to -60 . An example of a decision change that results from the absence

of the identified critical region is showed in Fig. 1.3.

Sentiment Classification

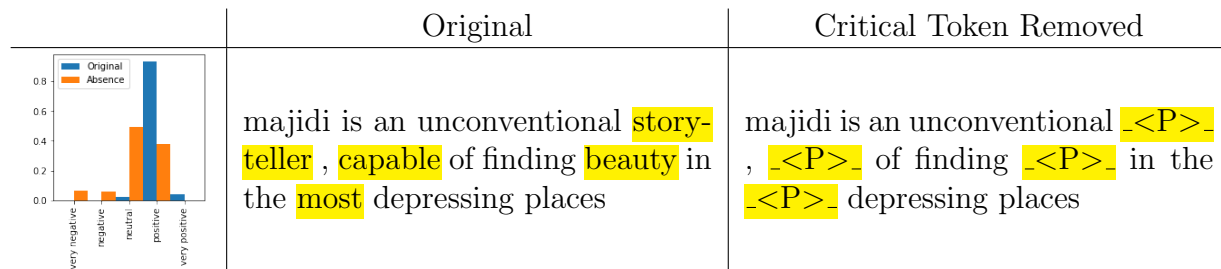


Figure 5.2: Example of a decision change due to replacement of critical tokens in the decision-making process. (left) prediction confidences for decision made with original sentences and with the critical tokens replaced sentence; (middle) the original sentence; (right) the critical tokens replaced sentence. For keeping the sentence structure consistent, the identified critical tokens is replaced with the padding token ‘<P>’. The differences between two sentences are highlighted.

In NLP domain, we use Stanford Sentiment Treebank dataset for sentence sentiment classification task. Following the work by Kim et al. [41], a convolutional neural network is leveraged as the reference network $N_{sentiment}$ for evaluating the performance of the explainability methods. Each token in the sentence is converted into its word2vec embedding [55] representation and later used as the network’s input feature for better classification accuracy. Each explainability method is tasked to identify the critical factors (the critical tokens) that is important for the decision-making process of the network $N_{sentiment}$ within the input sentence. Similar to previous two task domains, the identified critical factor is removed for the impact analysis metric I and I_{strict} . For removing the effect of the tokens but keeping the sentence structure unaffected, the identified tokens is replaced with special padding token ‘<P>’, which corresponding to the zero value vector in the word2vec embedding space. An example of a decision change that results from the replacement of the identified critical tokens is shown in Fig. 5.2.

5.1.2 Experiment 2: Adversarial Distraction

For the second experiment, we quantitatively evaluate the performance of several state-of-the-art explainability methods using the two variants of Impact Score (i.e., I and I_{strict}),

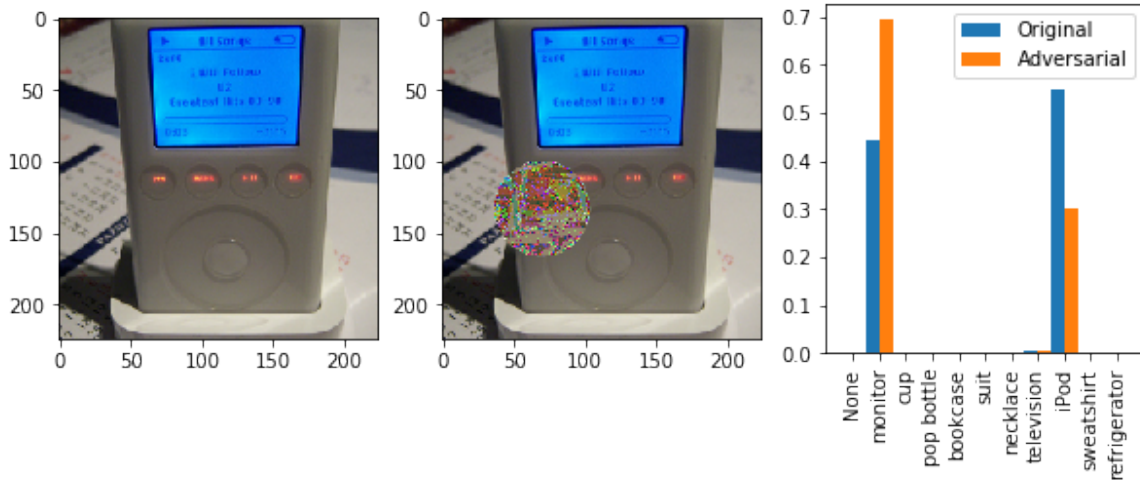


Figure 5.3: Example of a directed erroneous decision due to the adversarially impacted area. (left) original untampered image, (center) tampered image with an adversarial patch, (right) prediction confidences of decisions made with an untampered image and adversarially tampered image. The adversarial patch leads to a change in decision.

as well as $I_{coverage}$ for each explainability method M in the presence of 'distractions' in the form of adversarial patches to better study the impact in directed erroneous decisions. With the adversarial patches being the control variable, the critical region that is important to the decision-making process is largely known a priori to be the adversarial patch itself, and as such $I_{coverage}$ provides an additional quantitative indicator for the ability of the explainability method to identify such adversarially impacted areas within the images that has a direct impact in the decisions made by the deep neural network.

Image Classification

To study the performance of explainability methods in an erroneous scenario, we introduce a visual 'distraction' for fooling the reference network N_{image} . More specifically, we leverage the adversarial patches from the work of Brown et al. [10] as such 'distraction'. For generating the adversarial patch, we fix the reference network N_{image} aforementioned in Experiment 1 image classification task and apply adversarial training for the same subset of the ImageNet [69] dataset as Experiment 1 image classification task. Later, we randomly (e.g. random translation and random rotation of the patch) overlay the resulting adversarial patches on the same subset of images with different patch scales ranging from

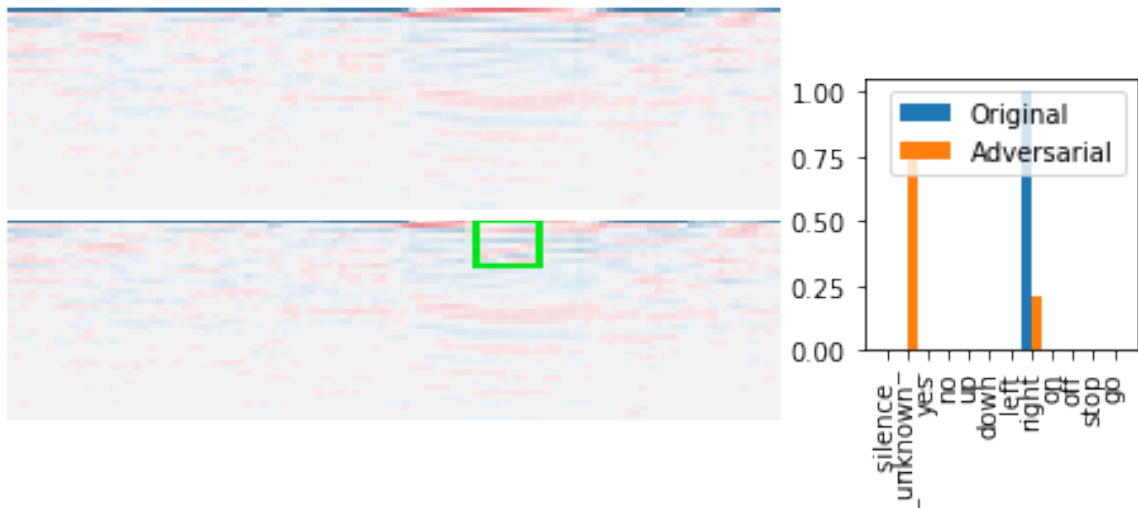


Figure 5.4: Example of a decision change due to adversarially impacted area. (top-left) original MFCC representation of an utterance audio signal; (bottom-left) adversarially tampered MFCC; (right): prediction confidences for the decision made with original MFCC and with the adversarially tampered of the critical region. The adversarial patch leads to a change in decision.

0.3 to 0.7. An example of a directed erroneous decision due to the adversarially impacted area is known in Fig. 5.3. We compute I , I_{strict} , and $I_{coverage}$ for each patch scale over the test images, of which the prediction classes change to the adversarially targeted classes.

Utterance Classification

For creating the same directed erroneous scenario in the speech recognition domain, we again apply the same adversarial training technique from the work of Brown et al. [10]. The same aforementioned MFCC feature extraction and network $N_{utterance}$ is used for evaluating the performance of the explainability methods. Due to the gradient descent training instability of MFCC transformation, we are not able to train and apply the adversarial patch directly on the input audio signal. However, this does not affect the usage of the $I_{coverage}$ metric, as the directed erroneous distraction can be presented in the MFCC feature space. As such, we fix the reference network $N_{utterance}$ and apply the adversarial patch training in the MFCC feature space. The trained adversarial patch is later randomly overlaid (random translation) on the MFCC feature of an audio input signal. An example of a directed erroneous decision due to adversarially modified MFCC feature is shown in

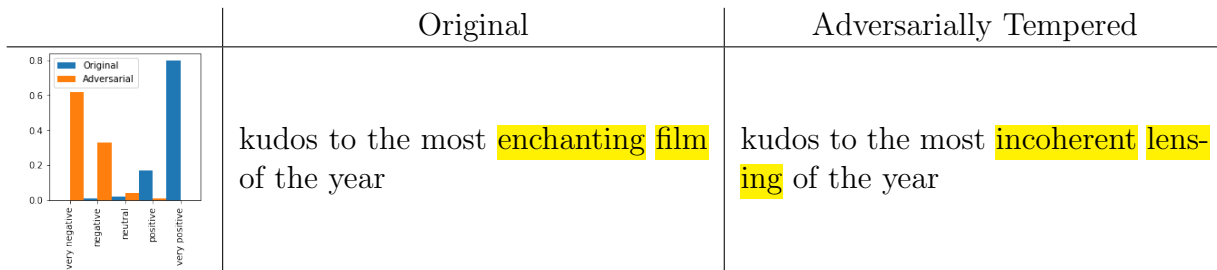


Figure 5.5: Example of a decision change due to the adversarially tempered tokens. (left) prediction confidences for the decision made with original sentences and with the adversarially tempered sentence. The adversarial tokens lead to a change in the decision; (middle) the original sentence; (right) the adversarially tempered sentence. The differences between the two sentences are highlighted.

Fig. 5.4, where the highlighted and adversarially overlaid area fools the reference network $N_{utterance}$.

Sentiment Classification

It is much more difficult to create an erroneous scenario for the sentiment classification task because the aforementioned adversarial patch technique can not be applied in the natural language setting without impacting the structure and semantic of an input sentence. In order to solve this issue, we instead create a learn-able adversarial network $N_{sentiment}^{adversarial}$ that acts as a 'man-in-middle' attacking network between the input feature and the reference network $N_{sentiment}$. Due to the discrete nature of the input token and the un-differentiable nature of the word2vec transformation, the adversarial modification is not directly applied to the input sentence but the continuous word2vec embedding space. With the reference network, $N_{sentiment}$ The attacking network $N_{sentiment}^{adversarial}$ is trained to predict both which tokens to attack and the modification offset in the embedding space, of which targets to change the prediction of the reference network $N_{sentiment}$. To better understand the adversarially attacked word2vec embedding, we reconstruct the embedding back to its natural language token representation by greedy nearest cosine distance matching. An example of the adversarially modified sentence that leads to wrongful prediction is shown in Fig. 5.5.

5.1.3 Explainability Methods Under Study

In this study, the proposed Impact Score and Impact Coverage is leveraged to perform a comprehensive analysis of several state-of-the-art explainability methods in the research literature. More specifically, the methods under study are: i) LIME [64], ii) SHAP [53], and iii) Expected Gradients [17]. These methods were selected as they represent a good coverage of both popular and state-of-the-art methods from both the proxy and direct categories of explainability methods.

5.2 Experimental Results

The experimental results for the two experiments conducted in this study are presented below.

5.2.1 Experiment 1: General Scenario

Table 5.1: Performance of tested explainability methods based on impact on network decisions.

Method	Image		Sentiment		Utterance	
	I_{Strict}	I	I_{Strict}	I	I_{Strict}	I
LIME	35.12%	38.05%	50.32%	50.96%	42.83%	43.06%
SHAP	68.54%	73.90%	24.95%	25.16%	62.52%	62.59%
EG	72.93%	77.80%	56.43%	56.85%	63.87%	64.10%

The quantitative performance of the three tested explainability methods as determined by the proposed Impact Scores in the first experiment is shown in Table 1. Many interesting observations can be made. In term of the image classification task, it can be observed that LIME achieved the lowest I and I_{strict} scores, thus indicating that the critical regions identified by LIME had the lowest impact on the actual decision-making process of the network in identifying the class for a given image when compared to the other tested methods. It can also be observed that there is a progressive increase in decision-making impact from SHAP to EG, with a significant absolute increase in I and I_{Strict} by over 3.9% and over 4.4%, respectively. The same trend is also observed in the utterance classification

Table 5.2: Image Classification: Performance of tested explainability methods at different adversarial scales

Scale	LIME [64]			SHAP [53]			EG [17]		
	$I_{coverage}$	I	I_{strict}	$I_{coverage}$	I	I_{strict}	$I_{coverage}$	I	I_{strict}
0.3	0.64%	9.70%	9.80%	3.53%	40.41%	41.32%	2.57%	36.00%	36.80%
0.4	1.53%	9.90%	10.00%	3.33%	36.73%	37.54%	2.31%	35.00%	35.40%
0.5	0.67%	8.70%	8.80%	3.08%	36.28%	36.62%	2.09%	39.20%	39.40%
0.6	0.37%	10.50%	10.60%	3.04%	38.20%	38.78%	1.88%	39.00%	39.40%
0.7	0.41%	10.80%	10.80%	2.87%	43.16%	43.61%	1.80%	42.80%	43.20%

Table 5.3: Utterance Classification: Performance of tested explainability methods at different area of adversarial patch

Area	LIME [64]			SHAP [53]			EG [17]		
	$I_{coverage}$	I	I_{strict}	$I_{coverage}$	I	I_{strict}	$I_{coverage}$	I	I_{strict}
0.10	3.59%	33.79%	33.79%	0.66%	10.30%	10.30%	0.47%	4.44%	4.44%
0.40	13.98%	15.59%	15.65%	2.33%	8.09%	8.09%	1.64%	4.67%	4.67%
0.70	22.99%	13.72%	13.78%	3.68%	6.75%	6.75%	2.45%	4.38%	4.48%
1.00	29.72%	13.89%	13.95%	4.74%	5.70%	5.70%	3.01%	3.52%	3.52%

task. However, LIME achieves significantly higher I_{strict} and I scores over the SHAP for the sentiment classification task. And the EG still achieves the highest score among all tested methods. In order to gain a more intuitive understanding of the performance, an example image, the critical regions identified by tested explainability methods, and the prediction confidences with and in absence of the identified critical regions are shown in Fig. 5.6.

5.2.2 Experiment 2: Adversarial Distraction

The quantitative performance of the three tested explainability methods as determined by the proposed Impact Score and Impact Coverage in the second experiment is shown in Table 5.2, Table 5.3, and Table 5.4. A number of interesting observations can be made. For the image classification task, it can be observed that LIME achieved the lowest I , I_{strict} , and $I_{coverage}$ scores across all adversarial patch scales, thus indicating that the critical regions identified by LIME have the lowest impact as well as coverage of the adversarially impacted areas in the test images amongst the tested methods. Unlike Experiment 1, SHAP performs better than EG for I , I_{strict} , and $I_{coverage}$. Meanwhile, the interesting

Figure 5.6: Example images, the corresponding critical regions identified by tested explainability methods, and prediction confidences with and in absence of the identified critical regions.

LIME [64]

SHAP [53]

EG [17]

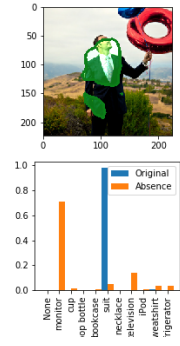
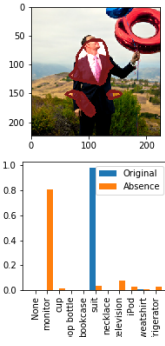
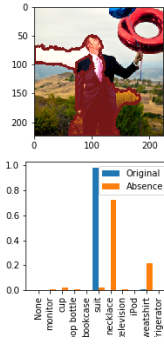
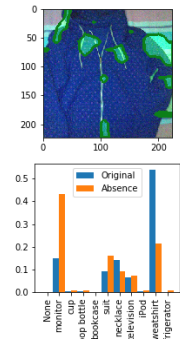
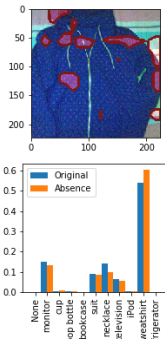
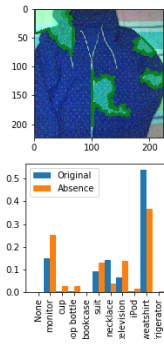
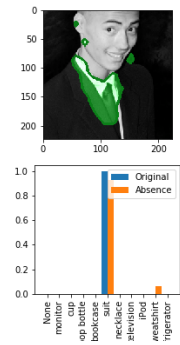
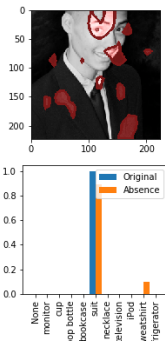
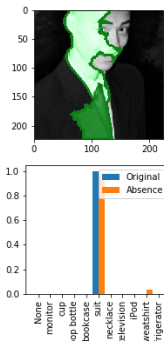


Table 5.4: Sentiment Classification: Performance of tested explainability methods at different number of adversarial tokens

#Adversarial	LIME [64]			SHAP [53]			EG [17]		
	$I_{coverage}$	I	I_{strict}	$I_{coverage}$	I	I_{strict}	$I_{coverage}$	I	I_{strict}
1	12.32%	15.01%	15.25%	13.38%	5.15%	8.96%	18.89%	27.36%	28.33%
2	22.19%	16.51%	16.99%	22.05%	11.24%	12.68%	32.45%	29.90%	30.62%
3	26.77%	13.64%	14.35%	28.71%	10.53%	11.72%	40.30%	23.68%	25.12%
4	33.82%	15.14%	15.38%	30.91%	11.30%	12.02%	44.02%	28.85%	29.33%

observation is that the $I_{coverage}$ decreases as the patch scale increases for all three tested methods. We argue this potentially indicates that different regions within the adversarial patches play different importance for the network’s decision-making process. For visual inspecting the quality of the three tested explainability methods in the directed erroneous scenario, samples with adversarial patches at different scales are displayed in Fig. 5.8. As for the utterance classification task, it is surprising to observe that LIME outperforms both SHAP and EG by a significant margin across all three metrics. What’s more, it is counter-intuitive to observe that both the I and I_{strict} decreases as the $I_{coverage}$ increases. To further interpret this observation, more experiments are needed for gaining more insights. In terms of the sentiment classification task, we observe that the performance difference between the three methods is less dramatic than the previous two classification tasks. Similar to the trend in experiment 1, EG’s performance comes before two other methods, LIME and SHAP, by a clear margin.

Figure 5.7: Example sentences, the corresponding critical tokens identified by tested explainability methods, and prediction confidences with and in absence of the identified critical tokens.

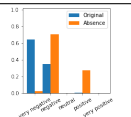
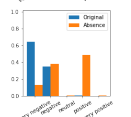
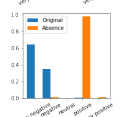
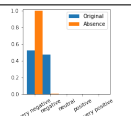
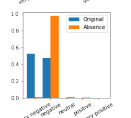
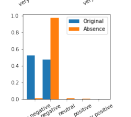
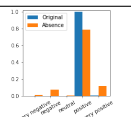
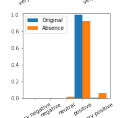
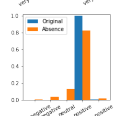






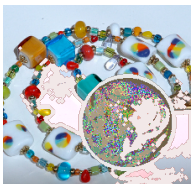
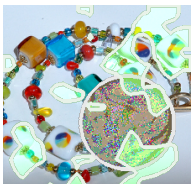
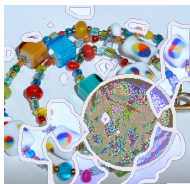

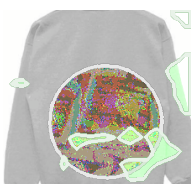
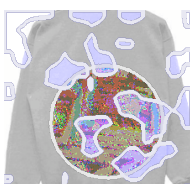
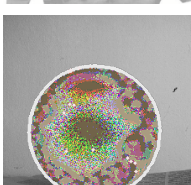
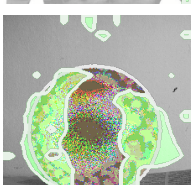
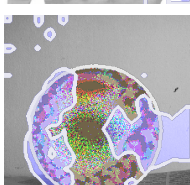
Original	a gob of drivel so sickly sweet , even the eager consumers of moore 's pasteurized ditties will retch it up like rancid crème brûlée	
LIME	a gob of <P> so sickly sweet , even <P> eager <P> of moore 's pasteurized ditties will retch it up like <P> cr me br l e	
SHAP	a gob of <P> so sickly sweet , even the eager consumers of moore 's pasteurized ditties will retch it up like <P> cr <P> br l e	
EG	a gob of <P> so <P> sweet , even the eager consumers of moore 's pasteurized ditties will retch it up like <P> cr me br l e	
Original	a dreary , incoherent , self-indulgent mess of a movie in which a bunch of pompous windbags drone on inanely for two hours ... a cacophony of pretentious , meaningless prattle	
LIME	<P> dreary , incoherent , self <P> mess of <P> movie in which <P> bunch of pompous windbags drone on inanely for two hours <P> cacophony of pretentious , meaningless prattle	
SHAP	a <P> , <P> , self indulgent mess of a movie in which a bunch of pompous windbags drone on inanely for two hours a cacophony of pretentious , <P> prattle	
EG	a <P> , <P> , self indulgent mess of a movie in which a bunch of pompous windbags drone on inanely for two hours a cacophony of pretentious , <P> prattle	
Original	an intelligent fiction about learning through cultural clash	
LIME	<P> intelligent fiction about learning through <P> <P>	
SHAP	an intelligent <P> about <P> through cultural <P>	
EG	an <P> <P> about learning through cultural <P>	

Figure 5.8: Example adversarially modified erroneous images via adversarial patches at different scales, and the corresponding critical regions identified by tested explainability methods as being important to the decision made by the network.

Scale / GT / Adv	LIME [64]	SHAP [53]	EG [17]
0.30 / TV / Monitor			
0.40 / Suit / Cup			
0.50 / Necklace / Cup			
0.60 / Sweatshirt / Monitor			
0.70 / Cup / Necklace			

Chapter 6

Conclusion

6.1 Summary

In this study, we explored a more machine-centric strategy for quantifying the performance of explainability methods on deep convolutional neural networks by quantifying the importance of critical factors identified by an explainability method. For a given decision made by a network, we study the impact on both the decision and the confidence in the decision, and additional coverage of adversarially impacted factors in the directed erroneous decision scenario. A comprehensive analysis using this approach showed that, in the case of visual perception tasks, speech recognition tasks and natural language processing tasks, some of the most popular and widely-used methods such as LIME, SHAP and EG may produce explanations that may not be as reflective as expected of what the deep neural network is leveraging to make decisions. The results in three different task domains also indicate an unclear conclusion. In the general testing scenario, EG outperforms both LIME and SHAP by a clear margin across all three task domains. However, EG’s performance is less convincing in the erroneous testing scenario. Under the adversarial attack, SHAP performs better than EG and LIME in the image classification task; LIME comes first by a significant margin over SHAP and LIME in the utterance classification; EG leads the all three performance metrics in the sentiment classification. With these being said, we observe that no explainability method can steadily outperform others in all test scenarios and all test task domains. What’s more, there is significant room for improvement for all explainability methods. While by no means perfect, the hope is that the proposed machine-centric strategy helps push the conversation forward towards better metrics for evaluating explainability methods in a manner that gives insights to guide network error

mitigation as well as improve trust in deep neural networks.

6.2 Future Works

In future, we would like to explore our explainability assessment framework in the following different directions:

- further extend the explainability experiment on to more different explainability methods;
- apply our assessment framework to more complicated models and tasks;
- incorporating new metrics that can reflect the level of “interpretability” of a given explanation.

Expand Coverage on Explainability Algorithms

In this thesis, we carried out experiments for quantitatively assessing the performance of three state-of-the-art explainability methods, LIME [64], SHAP [53] and Expected Gradient (EG) [17]. Despite these explainability methods are the most popular and representative ones in their categories, there are more different variants of these explainability methods targeting to overcome certain drawbacks of the aforementioned methods. For comprehensive analysis, we want to experiment with more explainability methods, such as Integrated Gradient [80], Guided Backpropagation [76], Guided GradCAM [72], SmoothGrad [75] and Expected Gradients [17].

Explore Possibility with Complicated Models and Tasks

Due to the different limitations discussed previously, we were only able to experiment with simpler and less dynamic models in the NLP and audio understanding domains. A recent trend suggests more complicated models, such as recurrent neural network [67, 36] and transformer [86, 15], are attracting more and more attention. To accommodate this irreversible trend, we would like to study how explainability methods behave with more complicated and dynamic models. In addition, we only studied the explainability methods for classification tasks in this thesis. This is mainly because those many explainability methods were proposed and demonstrated solely for the classification task. What and how

to interpret for DNN in other tasks, such as semantic segmentation in computer vision, machine translation in NLP, and speech diarization in speech understanding, remain as open questions and ongoing active research. As one of the future directions, we want to study how to apply the principle of our assessment framework on different tasks.

Consideration for Human Interpretability

As discussed in Chapter 3, explainability methods need to balance between the faithfulness and interpretability. The proposed evaluation metrics, namely **Impact Score** and **Impact Coverage**, focus primarily on the faithfulness of the explainability methods. To one extreme, an explainability method can theoretically achieve high **Impact Score** and **Impact Coverage** by providing one-pixel explanations for image classification since it has been shown that altering a single pixel in an image can change the decision of a DNN [79]. In this extreme case, the one-pixel explanation can provide little insight for the human to understand the decision-making process of a DNN. Due to this reason, we want to study additional metrics that can evaluate the level of interpretability of explainability methods. Accommodating with the proposed two metrics, we hope that this machine-centric evaluation framework can provide a comprehensive perspective on the performance of explainability methods.

References

- [1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pages 265–283, 2016.
- [2] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. In *Advances in Neural Information Processing Systems*, pages 9505–9515, 2018.
- [3] Naveed Akhtar and Ajmal Mian. Threat of adversarial attacks on deep learning in computer vision: A survey. *CoRR*, abs/1801.00553, 2018.
- [4] Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Qiang Cheng, Guoliang Chen, et al. Deep speech 2: End-to-end speech recognition in english and mandarin. In *International Conference on Machine Learning*, pages 173–182, 2016.
- [5] Microsoft Azure. Model interpretability in azure, 2019.
- [6] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7), 2015.
- [7] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155, 2003.
- [8] Yoshua Bengio, Yann LeCun, et al. Scaling learning algorithms towards ai. *Large-scale kernel machines*, 34(5):1–41, 2007.

- [9] Ali Borji and Laurent Itti. State-of-the-art in visual attention modeling. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):185–207, 2012.
- [10] Tom B Brown, Dandelion Mané, Aurko Roy, Martín Abadi, and Justin Gilmer. Adversarial patch. *arXiv preprint arXiv:1712.09665*, 2017.
- [11] Marisa Carrasco. Visual attention: The past 25 years. *Vision research*, 51(13):1484–1525, 2011.
- [12] Tsung-Han Chan, Kui Jia, Shenghua Gao, Jiwen Lu, Zinan Zeng, and Yi Ma. Pcanet: A simple deep learning baseline for image classification? *IEEE transactions on image processing*, 24(12):5017–5032, 2015.
- [13] Google Cloud. Ai explainability whitepaper, 2019.
- [14] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *Journal of machine learning research*, 12(Aug):2493–2537, 2011.
- [15] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
- [16] Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*, 2017.
- [17] Gabriel Erion, Joseph D Janizek, Pascal Sturmfels, Scott Lundberg, and Su-In Lee. Learning explainable models using attribution priors. *arXiv preprint arXiv:1906.10670*, 2019.
- [18] Gabriel Erion, Joseph D Janizek, Pascal Sturmfels, Scott Lundberg, and Su-In Lee. Learning explainable models using attribution priors. *arXiv preprint arXiv:1906.10670*, 2019.
- [19] Kevin Eykholt, Ivan Evtimov, Earlene Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Robust physical-world attacks on deep learning visual classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1625–1634, 2018.
- [20] Nicholas Frosst and Geoffrey Hinton. Distilling a neural network into a soft decision tree. *arXiv preprint arXiv:1711.09784*, 2017.

- [21] Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. Allennlp: A deep semantic natural language processing platform. *arXiv preprint arXiv:1803.07640*, 2018.
- [22] Amirata Ghorbani, James Wexler, James Y Zou, and Been Kim. Towards automatic concept-based explanations. In *Advances in Neural Information Processing Systems*, pages 9273–9282, 2019.
- [23] Ian Goodfellow, Nicolas Papernot, Sandy Huang, Yan Duan, Pieter Abbeel, and Jack Clark. Attacking machine learning with adversarial examples. *OpenAI*. <https://blog.openai.com/adversarial-example-research/>, 2017.
- [24] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.
- [25] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 6645–6649. IEEE, 2013.
- [26] Fengxiang He, Tongliang Liu, and Dacheng Tao. Why resnet works? residuals generalize. *arXiv preprint arXiv:1904.01367*, 2019.
- [27] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [28] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal processing magazine*, 29(6):82–97, 2012.
- [29] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.
- [30] Hooker and et al. A benchmark for interpretability methods in deep neural networks. In *NeurIPS*, pages 9734–9745, 2019.
- [31] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.

- [32] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *CVPR*, volume 1, page 3, 2017.
- [33] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.
- [34] Laurent Itti and Christof Koch. Computational modelling of visual attention. *Nature reviews neuroscience*, 2(3):194–203, 2001.
- [35] Laurent Itti, Christof Koch, and Ernst Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 20(11):1254–1259, 1998.
- [36] Michael I Jordan. Serial order: A parallel distributed processing approach. In *Advances in psychology*, volume 121, pages 471–495. Elsevier, 1997.
- [37] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188*, 2014.
- [38] Andrei Kapishnikov, Tolga Bolukbasi, Fernanda Viégas, and Michael Terry. Xrai: Better attributions through regions. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4948–4957, 2019.
- [39] Been Kim, Rajiv Khanna, and Oluwasanmi O Koyejo. Examples are not enough, learn to criticize! criticism for interpretability. In *Advances in neural information processing systems*, pages 2280–2288, 2016.
- [40] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, and Rory Sayres. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). *arXiv preprint arXiv:1711.11279*, 2017.
- [41] Yoon Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.
- [42] Pieter-Jan Kindermans, Sara Hooker, Julius Adebayo, Maximilian Alber, Kristof T Schütt, Sven Dähne, Dumitru Erhan, and Been Kim. The (un) reliability of saliency methods. In *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, pages 267–280. Springer, 2019.
- [43] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1885–1894. JMLR. org, 2017.

- [44] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.
- [45] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.
- [46] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016.
- [47] Isaac Lage, Andrew Ross, Samuel J Gershman, Been Kim, and Finale Doshi-Velez. Human-in-the-loop interpretability prior. In *Advances in Neural Information Processing Systems*, pages 10159–10168, 2018.
- [48] Qiuxia Lai, Wenguan Wang, Salman Khan, Jianbing Shen, Hanqiu Sun, and Ling Shao. Human vs. machine attention in neural networks: A comparative study. *arXiv preprint arXiv:1906.08764*, 2019.
- [49] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436, 2015.
- [50] Bo Li, Tara N Sainath, Arun Narayanan, Joe Caroselli, Michiel Bacchiani, Ananya Misra, Izhak Shafran, Hasim Sak, Golan Pundak, Kean K Chin, et al. Acoustic modeling for google home. In *Interspeech*, pages 399–403, 2017.
- [51] Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. In *Advances in Neural Information Processing Systems*, pages 6389–6399, 2018.
- [52] Zhong Qiu Lin, Audrey G Chung, and Alexander Wong. Edgespeechnets: Highly efficient deep neural networks for speech recognition on the edge. *arXiv preprint arXiv:1810.08559*, 2018.
- [53] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, pages 4765–4774, 2017.
- [54] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.
- [55] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

- [56] Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černocký, and Sanjeev Khudanpur. Recurrent neural network based language model. In *Eleventh Annual Conference of the International Speech Communication Association*, 2010.
- [57] Grégoire Montavon, Sebastian Lapuschkin, Alexander Binder, Wojciech Samek, and Klaus-Robert Müller. Explaining nonlinear classification decisions with deep taylor decomposition. *Pattern Recognition*, 65:211–222, 2017.
- [58] Timothy Niven and Hung-Yu Kao. Probing neural network comprehension of natural language arguments. *arXiv preprint arXiv:1907.07355*, 2019.
- [59] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: an asr corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5206–5210. IEEE, 2015.
- [60] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pages 8024–8035, 2019.
- [61] Petsiuk and et al. Rise: Randomized input sampling for explanation of black-box models. *arXiv preprint arXiv:1806.07421*, 2018.
- [62] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9, 2019.
- [63] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. *arXiv preprint*, 2017.
- [64] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144. ACM, 2016.
- [65] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Anchors: High-precision model-agnostic explanations. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

- [66] Andrew Slavin Ross and Finale Doshi-Velez. Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients. In *Thirty-second AAAI conference on artificial intelligence*, 2018.
- [67] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science, 1985.
- [68] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.
- [69] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.
- [70] Tara N Sainath and Carolina Parada. Convolutional neural networks for small-footprint keyword spotting. In *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.
- [71] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 618–626, 2017.
- [72] Ramprasaath R Selvaraju, Abhishek Das, Ramakrishna Vedantam, Michael Cogswell, Devi Parikh, and Dhruv Batra. Grad-cam: Why did you say that? *arXiv preprint arXiv:1611.07450*, 2016.
- [73] Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K Reiter. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In *Proceedings of the 2016 acm sigsac conference on computer and communications security*, pages 1528–1540, 2016.
- [74] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- [75] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017.

- [76] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014.
- [77] Dmitry Storcheus, Afshin Rostamizadeh, and Sanjiv Kumar. A survey of modern questions and challenges in feature extraction. In *Feature Extraction: Modern Questions and Challenges*, pages 1–18, 2015.
- [78] Peter Norvig Stuart J. Russell. *Artificial Intelligence A Modern Approach*.
- [79] Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation*, 23(5):828–841, 2019.
- [80] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3319–3328. JMLR. org, 2017.
- [81] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.
- [82] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.
- [83] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.
- [84] Raphael Tang and Jimmy Lin. Deep residual learning for small-footprint keyword spotting. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5484–5488. IEEE, 2018.
- [85] Erico Tjoa and Cuntai Guan. A survey on explainable artificial intelligence (xai): Towards medical xai. *arXiv preprint arXiv:1907.07374*, 2019.
- [86] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

- [87] Alex Waibel, Toshiyuki Hanazawa, Geoffrey Hinton, Kiyohiro Shikano, and Kevin J Lang. Phoneme recognition using time-delay neural networks. *IEEE transactions on acoustics, speech, and signal processing*, 37(3):328–339, 1989.
- [88] Yisen Wang, Xuejiao Deng, Songbai Pu, and Zhiheng Huang. Residual convolutional etc networks for automatic speech recognition. *arXiv preprint arXiv:1702.07793*, 2017.
- [89] Pete Warden. Speech commands: A dataset for limited-vocabulary speech recognition. *arXiv preprint arXiv:1804.03209*, 2018.
- [90] Nilmini Wickramasinghe. Deepr: a convolutional net for medical records. 2017.
- [91] Alexander Wong, Mohammad Javad Shafiee, Brendan Chwyl, and Francis Li. Fermi-nets: Learning generative machines to generate efficient neural networks via generative synthesis. *arXiv preprint arXiv:1809.05989*, 2018.
- [92] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. 2016. *arXiv preprint arXiv:1611.05431*, 2016.
- [93] Chih-Kuan Yeh, Joon Kim, Ian En-Hsu Yen, and Pradeep K Ravikumar. Representer point selection for explaining deep neural networks. In *Advances in Neural Information Processing Systems*, pages 9291–9301, 2018.
- [94] Wen-tau Yih, Kristina Toutanova, John C Platt, and Christopher Meek. Learning discriminative projections for text similarity measures. In *Proceedings of the fifteenth conference on computational natural language learning*, pages 247–256. Association for Computational Linguistics, 2011.
- [95] Seungil You, David Ding, Kevin Canini, Jan Pfeifer, and Maya Gupta. Deep lattice networks and partial monotonic functions. In *Advances in neural information processing systems*, pages 2981–2989, 2017.
- [96] Tom Young, Devamanyu Hazarika, Soujanya Poria, and Erik Cambria. Recent trends in deep learning based natural language processing. *IEEE Computational Intelligence Magazine*, 13(3):55–75, 2018.
- [97] Alireza Zaeemzadeh, Nazanin Rahnavard, and Mubarak Shah. Norm-preservation: Why residual networks can become extremely deep? *arXiv preprint arXiv:1805.07477*, 2018.

- [98] Yu Zhang, William Chan, and Navdeep Jaitly. Very deep convolutional networks for end-to-end speech recognition. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4845–4849. IEEE, 2017.
- [99] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016.