



**Hochschule**  
**Bonn-Rhein-Sieg**  
University of Applied Sciences

**Masterarbeit**

# **Verfahren zur Erkennung von Manipulationen bei der Verkehrsschildbestimmung durch Künstliche Intelligenz**

Fachbereich Informatik  
Studiengang Master Informatik  
Erstprüfer: Prof. Markus Ullmann  
Zweitprüfer: Prof. Dr. Nico Hochgeschwender

eingereicht von:  
Mario Beckel  
Matr.-Nr. 9022094  
Tenktererstr. 6  
50679 Köln

Sankt Augustin, den 19.03.2021

# Inhaltsverzeichnis

<b>1 Einführung</b>	<b>1</b>
1.1 Motivation . . . . .	2
1.2 Zielsetzung . . . . .	3
1.3 Aufbau der Arbeit . . . . .	4
1.4 Stand der Forschung . . . . .	5
<b>2 Grundlagen</b>	<b>6</b>
2.1 Künstliche Neuronale Netze . . . . .	7
2.2 Distanzmetriken . . . . .	13
2.3 Heatmap . . . . .	13
2.4 Erklärbarkeit . . . . .	14
2.4.1 Eigenschaften von Erklärungen . . . . .	14
2.4.2 Nutzen von Erklärbarkeit . . . . .	16
2.5 Unterschiedliche Methoden zur Erzeugung von Erklärungen . . . . .	17
2.5.1 Klassifizierung von Methoden über die Erklärungen erzeugt werden können . . . . .	17
2.5.2 Verschiedene Erklär-Modelle . . . . .	18
<b>3 Erklär-Modelle</b>	<b>21</b>
3.1 Anchors . . . . .	21
3.2 Layerwise Relevance Propagation . . . . .	26
3.3 Theoretischer Vergleich von Anchors und LRP . . . . .	33
<b>4 Daten und Methoden</b>	<b>34</b>
4.1 Daten und Vorverarbeitung . . . . .	34
4.2 Netzwerkarchitektur . . . . .	34
4.3 LRP-Implementierung . . . . .	36
<b>5 Untersuchungen zu LRP</b>	<b>37</b>
5.1 Auswirkungen von Pixelmanipulationen im Eingangsbild . . . . .	38
5.2 Robustheit von LRP . . . . .	41
5.3 Qualität der Erklärung durch LRP . . . . .	44
5.3.1 Untersuchung mit dem Inception Netz . . . . .	46
5.3.2 Untersuchung mit dem einfachen Netz . . . . .	53
5.4 Manipulation über einen Aufkleber . . . . .	55
<b>6 Zusammenfassung und Ausblick</b>	<b>62</b>
<b>7 Literaturverzeichnis</b>	<b>65</b>

<b>Anhang</b>	<b>67</b>
7.1 Aufbau des verwendeten Inception Netzes . . . . .	68
7.2 Aufbau des verwendeten einfachen Netzes . . . . .	69
7.3 Verteilung der Relevanz-Werte bei den Heatmaps . . . . .	70
7.4 Heatmaps bei verschiedenen LRP-Parametern . . . . .	73

# Abbildungsverzeichnis

2.1	Vom Bild zur Zahlenmatrix. . . . .	8
2.2	Eine typische CNN-Architektur. . . . .	11
2.3	Die Faltung eines zweidimensionalen Arrays. . . . .	12
2.4	Beispiel einer Heatmap mit Farbleiste zur Erläuterung der Relevanz. .	14
3.1	Unterschiedliche Erklärung einer binären Entscheidung von LIME und Anchors. . . . .	22
3.2	Anchors: Erzeugung der relevantesten Superpixel. . . . .	25
3.3	Anchors: Schrittweise Ermittlung der relevantesten Superpixel. . . . .	25
3.4	Berechnung der Deep Taylor Decomposition. . . . .	28
3.5	Ermittlung der Relevanzwerte in der Backpropagation. . . . .	29
3.6	Der Verlauf der Propagation in einem Neuronalen Netz. . . . .	32
3.7	Einsatz unterschiedlicher LRP-Regeln auf den verschiedenen Layern. .	32
4.1	Die verschiedenen Klassen des GTSRB-Datensatzes. . . . .	35
4.2	Das Prinzip des Inception Moduls. . . . .	35
5.1	Ausschnitt aus der Umfrage. . . . .	39
5.2	Manhattan-Distanz und euklidische Distanz . . . . .	40
5.3	Verkehrszeichen: Original - Transformiert. . . . .	42
5.4	Heatmap: Original - Transformiert - Zurück-Transformiert . . . . .	42
5.5	Drei Arten des Pixel-Flippings. . . . .	48
5.6	Auswirkung der Anzahl der verwendeten Bilder auf den Kurvenverlauf.	49
5.7	Auswirkungen verschiedener LRP-Parameter. . . . .	50
5.8	Differenz der Kurven LRP-Parameter - Zufall. . . . .	52
5.9	Vergleich zweier Netze mit sechs verschiedenen LRP-Parameter Kombinationen, klassen-basiert. . . . .	53
5.10	Vergleich zweier Netze mit sechs verschiedenen LRP-Parameter Kombinationen, wahrscheinlichkeits-basiert. . . . .	54
5.11	Manipulation eines Verkehrszeichens über einen Patch. . . . .	56
5.12	Verteilung der Top 10 Relevanzwerte, die sich innerhalb des Patch-Bereiches befinden. (Bild mit Patch/ohne Patch.) . . . . .	58
5.13	Verteilung der Relevanzwerte, die sich innerhalb des Patch-Bereiches befinden. (Heatmap aus 10 Pixeln/gesamte Heatmap) . . . . .	59
5.14	Vergleich Heatmap mit Patch, Heatmap ohne Patch. . . . .	61

# **Tabellenverzeichnis**

5.1	LRP-Parameter beeinflussen mittlere Distanz zwischen Heatmaps . . . . .	44
5.2	In Untersuchung verwendete LRP-Parameter . . . . .	49
5.3	Vergleich der mittleren Relevanzwerte innerhalb und außerhalb des Patches . . . . .	56
5.4	Mittlere Relevanzwerte bei unterschiedlichen Patch-Positionen. . . . .	57
5.5	Vergleich der mittleren Relevanz-Wert bei verschiedenen LRP-Einstellungen (Grundlage Heatmap mit den 10 relevantesten Pixeln) . . . . .	60

# 1 Einführung

Künstliche Neuronale Netze haben sich in den vergangenen Jahren als Methoden für Klassifizierungsaufgaben etabliert. Aber auch in anderen Bereichen gehören sie inzwischen zum Standardrepertoire. Heute werden Neuronale Netze zur Optimierung, Klassifizierung, Mustererkennung und Bildverarbeitung eingesetzt. Es ist jedoch immer noch eine Herausforderung, Systeme von hoher Verlässlichkeit und Genauigkeit zu erstellen, die unter sich ändernden Umgebungsvariablen gute Ergebnisse liefern.

Einer Gruppe von Forschern aus den USA ist es gelungen, den Prozess des Erkennens von Verkehrsschildern durch Künstliche Intelligenz (KI) zu manipulieren. Sie konnten die Verkehrsschilder so präparieren, dass die eingesetzten Verfahren der KI die Schilder falsch klassifiziert haben. Dazu war nur ein kleiner Aufkleber auf dem Verkehrsschild notwendig, um eine komplett andere Bestimmung des Schildes zu erzielen. Die korrekte Interpretation des Schildes durch einen Menschen, wäre durch diese Manipulation nicht beeinträchtigt worden.<sup>1</sup>

In einer ähnlichen Studie im Jahr 2019 hat ein Forscherteam aus Tübingen gezeigt [21], dass die Technik der optischen Flussalgorithmen, die auf tiefen Neuronalen Netzen beruht und wahrscheinlich in zukünftigen autonomen Fahrzeugen eingesetzt werden wird, verwundbar gegenüber Hackerangriffen ist. Hier wurden durch ein Farbmuster im Wahrnehmungsbereich der optischen Sensoren Störsignale hervorgerufen. Dadurch wurde der Klassifizierungsprozess des Neuronalen Netzes beeinträchtigt.

Wie diese Studien zeigen, können die Vorhersagen von tiefen Neuronalen Netzen, über das Einbringen von kleinen Störungen manipuliert werden. Das untersuchte Problem ist besonders dann relevant, wenn Methoden der künstlichen Intelligenz eingesetzt werden, um autonom fahrende Fahrzeuge durch den Verkehr zu navigieren. In diesem Fall könnte eine falsche Interpretation eines Verkehrsschildes Unfälle verursachen und dadurch Gesundheit und Leben von Menschen bedrohen. Außerdem würde die Akzeptanz der Bevölkerung für autonom fahrende Fahrzeuge beeinträchtigt, wenn die zugrundeliegende Technologie nicht robust gegen zufällig oder gezielte Änderungen der Eingangsdaten ist.

Um Vertrauen in die korrekte Arbeitsweise von KI-basierten Systemen zu erlangen sind Kriterien erforderlich, mit denen die Qualität dieser Systeme beurteilt werden kann. Damit Fehlinterpretationen erkannt und zukünftig vermieden werden können, ist es notwendig, dass der Klassifizierungsprozess von Künstlicher Intelligenz für

---

<sup>1</sup><https://spectrum.ieee.org/cars-that-think/transportation/sensors/slight-street-sign-modifications-can-fool-machine-learning-algorithms>

Menschen nachvollziehbar gemacht wird. Wenn künstliche Intelligenz in der Bildverarbeitung eingesetzt wird, sollte ermittelt werden können, auf welche Bereiche des Bildes die Entscheidung der KI beeinflusst haben.

In meinem voran gegangenen Praxisprojekt wurden verschiedene Verfahren untersucht, die Transparenz in den Entscheidungsprozess einer KI-Architektur bringen können. Diese Masterarbeit soll auf den im Praxisprojekt gewonnen Kenntnissen aufbauen. Am Beispiel des Erkennens von Manipulationen bei der Interpretation von Verkehrsschildern soll der Nutzen von Methoden zur Erklärung ermittelt werden, um daraus Rückschlüsse auf die allgemeine Relevanz dieser Methoden, für ähnliche Einsatzbereiche ziehen zu können.

### 1.1 Motivation

Je mehr künstlich intelligente Entscheidungssysteme in den Alltag der Menschen Einzug halten, desto stärker können Ängste und Zweifel gegenüber solchen Systemen hervortreten. Dies ist nachvollziehbar, da es für Laien nahezu unmöglich ist, die Chancen und Risiken von KI-Systemen abzuschätzen.

Als Reaktion auf drohende Gefahren bei dem Einsatz von Künstlicher Intelligenz in Entscheidungssystemen wurde unter anderem die Datenschutz-Grundverordnung eingeführt. Aus dieser kann ein Recht auf Erklärbarkeit abgeleitet werden, damit die von einer Entscheidung Betroffenen nachvollziehen können, wie eine Entscheidung zustande gekommen ist.

Aus technischer Sicht ist Erklärbarkeit bis heute kein Qualitätskriterium für die Beurteilung der Qualität eines Verfahrens des Maschinellen Lernens. Dennoch erläutert Shirin Glander [13], dass Erklärbarkeit und Transparenz in mehrfacher Hinsicht auch für die Funktionalität von KI-Modellen vorteilhaft sein können. Aus ihrer Sicht sollten aus folgenden Gründen transparente Entscheidungsprozesse angestrebt werden:

1. **Das Modell verbessern:** Sind die Grundlagen der Entscheidungen oder Vorfahrsagen bekannt, dann kann die Sinnhaftigkeit der beim Training des Modells verwendeten Regeln analysiert werden. Die Transparenz von KI-Systemen kann davor schützen, falsche Schlüsse zu ziehen, weil nachvollziehbar wird, auf welchen Informationen eine Entscheidung beruht. Basiert die Klassifizierung eines Neuronalen Netzes wirklich auf dem zu klassifizierenden Objekt oder auf Metadaten oder dem Hintergrund eines Bildes.
2. **Vertrauen und Akzeptanz ermöglichen:** Menschen vertrauen der Vorhersage eines Modells eher, wenn sie den kausalen Zusammenhang, der zu der Entscheidung geführt hat, verstehen können. Dies ist besonders in Bereichen wichtig, in denen eine Fehlentscheidung zu gravierenden Schäden führen kann, also wenn die Entscheidung Einfluss auf das Leben oder die Gesundheit eines Menschen hat oder bedeutende finanzielle Folgen haben kann. [17, Kap. 2.1]

3. **Vorurteile und Fehler im Modell beseitigen:** Modelle, die mit Datensätzen trainiert wurden, die versteckten Vorurteile enthalten, können zu einer selbst erfüllenden Prophezeiung oder zur Stärkung von vorhandenen Diskriminierungen führen. Es können kausale Zusammenhänge suggeriert werden, wo gar keine sind. Cathy O’Neil hat in ihrem Buch viele Beispiele aufgeführt, die zeigen, dass durch einen nicht reflektierten Einsatz von KI, großer Schaden angerichtet werden kann.<sup>2</sup>

Da der Einsatz von Künstlicher Intelligenz in vielen Fällen vorteilhaft sein kann, sollten Wege gefunden werden, diesen so zu gestalten, dass die Menschen der Technik vertrauen können. Dazu ist es notwendig, dass die Betroffenen nachvollziehen können, wie eine Entscheidung zustande gekommen ist.

Institutionen und Betriebe, die Modelle des maschinellen Lernens einsetzen sind einem großen Druck ausgesetzt, den Entscheidungsprozess dieser Modelle nachvollziehbar zu machen. Dieser Druck kann im Extremfall dazu führen, dass Blackbox-Algorithmen nicht mehr eingesetzt werden. Aus diesem Grund hat sich das Forschungsfeld von Erklärbarer Künstlicher Intelligenz (Explainable Artificial Intelligence(XAI)) als Teilgebiet des Maschinellen Lernens entwickelt. XAI möchte transparente nichtlineare Lern-Methoden schaffen, theoretische Grundlagen für Modelle des Maschinellen Lernens legen und Werkzeuge zur Erzeugung von mehr Transparenz der Entscheidungen von Künstlicher Intelligenz bereitstellen. [28, S. 248]

Wenn die Erkennung von Verkehrsschildern durch künstliche Intelligenz sich im Verkehrsalltag etablieren soll, muss das Identifizieren der Schilder zuverlässig funktionieren. Wird ein Verkehrsschild falsch interpretiert kann dies bei autonom fahrenden Autos gravierende Folgen haben. Ein KI-basiertes System der Verkehrsschilderkennung muss in der Lage sein, unter verschiedenen Bedingungen wie unterschiedlichen Lichtverhältnissen (z.B. Tag, Nach, Nebel, Regen) und perspektivischen Verzerrungen, die Verkehrsschilder zuverlässig zu deuten. Die Schilder können durch Äste teilweise verdeckt sein oder ein unebener Fahrbahnbelag kann zu verwackelten Aufnahmen führen.

Damit die richtige Klassifizierung der Verkehrsschilder durch Methoden der KI zuverlässig funktioniert, ist es wichtig zu verstehen, wie die Entscheidung für eine bestimmte Klasse zustande gekommen ist.

## 1.2 Zielsetzung

Ziel der Masterarbeit ist, das Verhalten und die Leistungsfähigkeit von Methoden zur Erklärung von KI-Modellen in einem praktischen Anwendungsfall zu untersuchen. Die Ergebnisse der Untersuchung sollen dabei helfen, für ein konkretes Einsatzszenario das passende Verfahren zur Erklärung des Entscheidungsprozesses zu finden.

---

<sup>2</sup>Cathy O’Neil: Weapons of Math Destruction, 2016

Die Masterarbeit wird in Zusammenarbeit mit dem Bundesamt für Sicherheit in der Informationstechnik (BSI) durchgeführt. Ausgangspunkt ist ein auf ein Datensatz von deutschen Verkehrsschildern trainiertes Neuronales Netz, dass vom BSI zur Verfügung gestellt wird.

Der Entscheidungsprozess dieses Neuronalen Netzes wird im Rahmen dieser Arbeit analysiert. Zunächst werden zwei unterschiedliche Verfahren, mit denen der Entscheidungsprozess für Menschen nachvollziehbar gemacht werden kann, erkundet. Zum einen wird das Verfahren Anchors und zum anderen das Verfahren Layerwise Relevance Propagation (LRP) analysiert. Nach einer Phase der Voruntersuchung in der beide Verfahren eingesetzt wurden, wird im Hauptteil der Untersuchung LRP verwendet.

LRP wurde für die Masterarbeit ausgewählt, weil die Methode im Bereich von Erklärbarer Künstlicher Intelligenz einen relativ hohen Bekanntheitsgrad erlangt hat. Das Verfahren ist gut dokumentiert und es existiert verschiedene Python-Implementierung mit denen das Verfahren angewendet werden kann. Hinzu kommt, dass das Verfahren gut dokumentiert ist und das Entwicklungsteam durch Teilnahme an vielen Fachkongressen im stetigen Austausch mit der Fachwelt ist.

Als zweites Verfahren wurde Anchors auserkoren. Auch hier existiert eine aktuelle Implementierung für Python. Das Verfahren ist eine gute Ergänzung zu LRP, weil es aufgrund einer anderen Herangehensweise einen anderen Blickwinkel ermöglicht. Anchors wurde vom selben Forscherteam erstellt, das auch LIME entwickelt hat und wird als Weiterentwicklung von LIME betrachtet. Auf LIME wird in vielen wissenschaftlichen Arbeiten über XAI Bezug genommen.

In dieser Arbeit wird untersucht, wie aussagekräftig die Erklärungen sind, die die vorgestellten Methoden liefern, wie hoch der Rechenaufwand ist und wie stabil die Klassifizierung ist, wenn sich die Eingabewerte ändern. Diese Ergebnisse lassen Rückschlüsse auf die Praxistauglichkeit des Verfahrens zu. Ausschlaggebend für die praktische Verwendbarkeit ist auch die Qualität des Verfahrens, die auch Gegenstand der Untersuchung ist.

### 1.3 Aufbau der Arbeit

Im Kapitel 2 werden grundlegende Begriffe geklärt, die wichtig für das Verständnis dieser Arbeit sind. Zunächst wird einem kurzen Überblick über die theoretische Grundlage Neuronaler Netze gegeben. Die Neuronalen Netze, sowie die Distanzmetriken *Manhattan-Distanz* und *euklidische Distanz*, die im Anschluss in Abschnitt 2.2 vorgestellt werden, sind Grundlagen für die durchgeführten Untersuchungen. Da Erklärungen in einigen Methoden und in besonders in dem untersuchten Verfahren LRP über Heatmaps erzeugt werden, wird in Abschnitt 2.3 auf diese Form der Visualisierung von Daten eingegangen.

Anschließend wird in Abschnitt 2.4 näher erläutert, welche Aspekte in Bezug auf Erklärbarkeit von Bedeutung sind und welcher Nutzen daraus gezogen werden kann, wenn der Entscheidungsprozess von Neuronalen Netzen für einen Menschen nachvollziehbar ist.

Die Methoden, über die Entscheidungsprozesse transparent gestaltet werden können, sind Thema des Abschnitts 2.5. Hier werden verschiedene Ansätze vorgestellt, über die ermittelt werden kann, auf welchen Informationen die Entscheidung eines Neuronalen Netzes beruht. Es gibt eine Vielzahl von verschiedenen Methoden, die diesen Zweck erfüllen. Diese Methoden sind teilweise ähnlich und lassen sich hinsichtlich ihrer Vorgehensweise in verschiedene Ansätze gruppieren. In diesem Abschnitt wird ein Überblick über die bestehenden Ansätze gegeben.

In Kapitel 3 werden theoretische Kenntnisse über die beiden in dieser Arbeit verwendeten Erklär-Modelle Anchors und Layerwise Relevance Propagation vermittelt. Ein theoretischer Vergleich zwischen den beiden Modellen wird hergestellt.

Kapitel 4 erläutert die für die Untersuchung verwendeten Werkzeuge. Zum einen sind dies, die im Experiment verwendeten Daten, der Aufbau des verwendeten Neuronalen Netzes und die Python-Implementierung von LRP, die zur Erzeugung der Erklärung in den Untersuchungen verwendet worden ist.

Nach der Erläuterung der theoretischen Grundlagen und der in den Untersuchungen eingesetzten Werkzeuge, werden im Kapitel 5 die durchgeführten Untersuchungen und ihre Ergebnisse dargestellt. Untersuchungsschwerpunkt sind die Stabilität und die Robustheit von LRP und die Qualität der erzeugten Erklärungen. Abschließend wird das Verhalten von LRP bei einer Manipulation des Bildes durch einen virtuellen Aufkleber analysiert.

Im letzten Kapitel werden die Ergebnisse der Arbeit zusammengefasst und ein Ausblick auf mögliche zukünftige Untersuchungen und den Entwicklungsbedarf im Bereich Erklärbarkeit bei Künstlicher Intelligenz gegeben. Im Anhang findet sich der ausführliche Aufbau der für die Untersuchung eingesetzten Netze und detailliertere Diagramme und Heatmaps zur Erläuterung der Untersuchung mit den virtuellen Aufklebern auf den Bildern.

## 1.4 Stand der Forschung

Eingeführt wurde Layerwise Relevance Propagation (LRP) von [2]. In dieser Arbeit wird der Bezug zwischen LRP und Deep Taylor Decomposition hergestellt und hier finden sich erste Definitionen von Propagationsregeln. LRP wurde ursprünglich zur Erklärung von Feed-Forward Netzen, wie Convolutional Netzen eingeführt. Später wurde der Einsatz für Recurrent Neuronale Netze ermöglicht. [1, S.3] Wissenschaftliche Publikationen zu LRP stammen hauptsächlich von dem Forscherteam zu LRP vom Fraunhofer Heinrich Hertz Institut und der Technischen Universität Berlin und

dem Umfeld. Zu dem Projekt existiert eine eigene Webseite<sup>3</sup>, auf der viele wissenschaftlichen Arbeiten zu LRP, Tutorials und Implementierungen von LRP verlinkt sind. Wissenschaftliche Arbeiten zu LRP, die nicht aus dem direkten Umfeld der Entwickler von LRP stammen, befassen sich unter anderem mit dem Einsatz von LRP im Gesundheitsbereich zur probabilistischen Vorhersage von Therapieentscheidungen [32] und zur Auswertung von Gehirn-Scans im Rahmen Alzheimer Vorhersage [5].

In der Untersuchung von Moritz Böhle [5] und seinen Kollegen [5] steht nicht LRP im Vordergrund, sondern die Anwendbarkeit von LRP zur Auswertung der Gehirn-Scans. Das Team vergleicht in ihrer Arbeit LRP mit der Methode Guided Backpropagation. Das Forscherteam untersucht die Auswirkung von verschiedenen Parameter-Einstellungen von LRP, auf die zur Erklärung erzeugte Heatmap. Hier bestehen Parallelen zu dieser Arbeit. Im Kontext der Analyse von Querschnittsbildern durch ein Gehirn ist es wichtig für einzelne Bereiche des Bildes Relevanzwerte zu ermitteln, da diese Bereiche für eine Alzheimer Diagnose relevant sein könnten. In diesem Umfeld sind Erklärungen lokaler Bereiche eines Bildes von höherer Bedeutung, als globale Erklärungen des gesamten Bildes.

Von Moritz Böhle und Team [5] stammt auch die PyTorch-Implementierung von LRP, die in dieser Arbeit verwendet wurde. Ansonsten existieren mit TensorFlow LRP Wrapper, die LRP-Toolbox und einer Keras Explanation Toolbox weitere Python-Implementierungen für LRP.

Die in dieser Arbeit durchgeführten Untersuchungen zur Qualität der Erklärungen, die über LRP erzeugt wurden, werden in einem anderen Untersuchungskontext in [26] geschildert und die Grundlagen für die Manipulation von Bildern über einen Patch werden in [6] gelegt.

Zu Anchors gibt neben der wissenschaftlichen Abhandlung des Autors relativ wenige Untersuchungen. Eine ausführliche Schilderung der Methode findet sich in [17]. Vom Entwickler selbst gibt es eine Python-Implementierung für Anchors. Daneben ist sie in das Python-Paket „Alibi“ integriert, dass verschiedene Erklär-Methoden beinhaltet. Es existiert auf eine Implementierung für Java und für die Programmiersprache R.

## 2 Grundlagen

Eine allgemein anerkannte einheitliche Definition von Interpretierbarkeit und Erklärbarkeit für das Forschungsgebiet der Erklärbaren Künstlichen Intelligenz existiert

---

<sup>3</sup><http://heatmapping.org/>

bis heute nicht. Auch wenn es verschiedene Versuche gibt die Begriffe zu umschreiben, konnte sich bisher keine Definition durchsetzen. Exemplarisch seien hier zwei Definitionen aufgeführt. Montavon et al. definiert in [19, S. 2] die beiden Begriffe. Interpretierbarkeit beschreibt er so:

Eine **Interpretation** ist die Abbildung eines abstrakten Konzepts (zum Beispiel einer vorhergesagten Klasse) auf einen Bereich, den der Mensch verstehen kann. [19, S. 2]

Zum Beispiel können Bilder und Texte interpretiert werden. Ein Mensch kann Bilder betrachten und Texte lesen, um sie zu verstehen, reicht die Wahrnehmung nicht aus, dazu müssen die sensorischen Daten interpretiert werden. Nicht interpretierbar sind zum Beispiel abstrakte Vektorräume oder Sequenzen von unbekannten Wörtern und Symbolen.

Demgegenüber grenzt der Autor den Begriff der Erklärbarkeit ab:

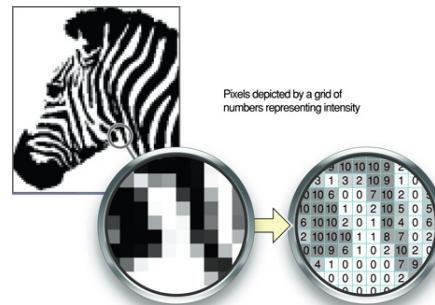
Eine **Erklärung** umfasst die gesamten Merkmale des zu interpretierenden Bereichs, die . . . dazu beigetragen haben, eine Entscheidung zu treffen (zum Beispiel eine Klassifikation oder eine Regression) [19, S. 2]

Die Merkmale, die für eine Erklärung ausschlaggebend sind, können hinsichtlich ihrer Relevanz, die sie für die Erzeugung der Vorhersage haben, bewertet werden. Einige Methode, deren Ziel es ist den Entscheidungsprozess von Neuronalen Netzen zu erklären, verwenden diese Relevanz-Werten, um ihr Ziel zu erreichen. Eine häufig eingesetzte Methode ist zum Beispiel, die Relevanz-Werte in Heatmaps visuell darzustellen. In der Heatmap werden die Bereiche des Bildes hervorgehoben, die maßgeblich zu einer Klassifikationsentscheidung beigetragen haben. Für die Klassifizierung von Bildern kann die Relevanz einzelner Pixel ermittelt und zur Erzeugung der Erklärung verwendet werden.

Nicht alle Autoren, die sich dem Thema Erklärbarkeit von Künstlicher Intelligenz widmen, folgen diesen Definitionen. Viele Autoren verwenden die beiden Begriffe synonym. Da der Unterschied der beiden Definitionen subtil ist und die Unterscheidung der beiden Begriffe für diese Arbeit nicht ausschlaggebend ist, werden die Begriffe hier synonym verwendet.

### 2.1 Künstliche Neuronale Netze

Künstliche Neuronale Netze sind eine häufig verwendete Methode, beispielsweise zur Klassifizierung von Objekten. Besonders im Bereich der Mustererkennung können sie überzeugende Ergebnisse erzielen und gehören daher zu den am häufigsten verwendeten Verfahren in der Bildverarbeitung. Das Konzept der Neuronalen Netze wurde aus der Biologie übernommen. Biologische Nervensysteme mit den darin



**Abbildung 2.1:** Ein Bild dargestellt als Zahlenmatrix<sup>1</sup>

stattfindenden neuronalen Prozessen unter Beteiligung von Neuronen und Synapsen lieferten die Vorlage für den Aufbau von Künstlichen Neuronalen Netzen.

Heute werden in vielen Bereichen Computer zur Auswertung von Bilddaten eingesetzt. Im Gegensatz zu Computern nehmen Menschen Bilder als Lichtreize, über im Auge vorhandene Photorezeptoren wahr. Diese Lichtreize werden als einzelne Bildpunkte vom Gehirn weiterverarbeitet. Computer verwenden dagegen Rastergrafiken, in denen Informationen zu Farbraum und Farbtiefe in Pixeln abgespeichert sind. Die Pixel werden digital in einem zweidimensionalen Raster abgespeichert. In Abbildung 2.1 ist zu sehen, wie ein Bild digital zur Weiterverarbeitung in eine Zahlenmatrix umgewandelt wird.

Ähnlich wie das menschliche Gehirn Lichtreize verarbeitet, nutzt das Neuronale Netz die Pixelinformationen, um vorgegebene Muster zu erkennen, oder selbständig zu erlernen.

Bestandteile eines Neuronalen Netzes sind Neuronen, die miteinander vernetzt sind. Die gerichtete und gewichtete Verbindung zwischen zwei Neuronen  $i$  und  $j$  wird mathematisch mit  $w_{ij}$  beschrieben. Diese Verbindung dient der Datenübertragung zwischen den Neuronen. Das Verbindungsgewicht kann stärkenden oder hemmenden Einfluss haben. In den meisten Fällen ist ein Neuron  $j$  mit vielen Vorgänger-Neuronen verbunden. Ein Vorgängerneuron von  $j$  wird hier als  $i$  bezeichnet. Eine Propagierungsfunktion nimmt die Ausgaben der Vorgänger-Neuronen als Vektor an und verarbeitet sie unter Einbezug der Verbindungsgewichte  $w_{ij}$ , das Resultat der Propagierungsfunktion ist ein Skalar. [8, S.35ff]

Jedes Neuron hat jederzeit einen gewissen Aktivierungsgrad. Über eine Aktivierungsfunktion und abhängig vom Schwellenwert, kann berechnet werden, wie stark die Aktivierung jedes Neurons ist. Der Schwellenwert kann als Reizschwelle des Neurons betrachtet werden, wird diese überwunden „feuert“ das Neuron. Die Aktivierungsfunktion ist oft global für alle Neuronen definiert, der Schwellenwert ist demgegenüber bei jedem Neuron unterschiedlich.

---

<sup>1</sup><https://developer.apple.com/library/archive/documentation/Performance/Conceptual/vImage/ConvolutionOperations/ConvolutionOperations.html>

## 2. Grundlagen

---

Es gibt verschiedene Schwellenwertfunktionen, die einfachste ist eine binäre Schwellenwertfunktion, bei der zu einem bestimmten Zeitpunkt der Wert von einem Zustand in einen anderen übergeht, ansonsten aber konstant bleibt. Es gibt also nur zwei mögliche Zustände und zu einem gewissen Zeitpunkt findet ein Übergang zwischen diesen statt. Über eine Ausgabefunktion kann definiert werden, welcher Wert von einem Neutron  $i$  an die Nachfolge-Neutronen, die mit  $i$  verbunden sind, weitergegeben wird. [8, S.37f]

Künstliche Neuronale Netze lassen sich hinsichtlich ihrer Topologie in unterschiedliche Kategorien einteilen. Es werden Feed-Forward-Netze, Rückkopplungsnetze und Vollständig-verbundene-Netze unterschieden. Feed-Forward-Netze bestehen aus verschiedenen Layern, die als Eingabe-Layer, verdeckte Layer und Ausgabe-Layer bezeichnet werden. Die Zwischen-Layer werden „versteckt“ genannt, weil sie von außen nicht sichtbar sind, sie liegen zwischen Eingabe- und Ausgabe-Layer. Je mehr Zwischenschichten es gibt, desto tiefer wird das Netz. Unter dem Begriff Deep Learning (deutsch: tiefes Lernen) wird daher eine Methode des Maschinellen Lernens verstanden, die auf Künstlichen Neuronalen Netzen beruht, die aus vielen Layern bestehen. Jeder dieser versteckten Layer ist in der Regel aus mehreren Neuronen zusammengesetzt. In den Feed-Forward-Netzen besteht die Verbindung zwischen den einzelnen Neuronen nur in Richtung Ausgabe-Layer. [8, S.41f]

Bei rückgekoppelten Netzen ist die Verbindung eines Neurons zu jedem beliebigen anderen Neuron möglich, auch zu sich selbst. In vollständig verbundenen Netzen sind grundsätzlich Verbindungen zu allen Neuronen des Netzes möglich. Direkte Rückkopplungen sind jedoch nicht erlaubt. Außerdem müssen die Verbindungen symmetrisch sein. [8, S.42ff]

Eine wichtige Eigenschaft eines Neuronalen Netzes ist seine Lernfähigkeit. Es gibt verschiedene Möglichkeiten wie das Netz lernt, das heißt, wie es seine Struktur anpassen kann, damit es zu besseren Ergebnissen kommt. Das Netz „lernt“ indem es neue Verbindungen zwischen den Neuronen herstellt, bestehende Verbindungen auflöst, die Gewichte der Verbindungen ändert, die Neuronenfunktion ändert, neue Neuronen hinzufügt oder bestehende Neuronen entfernt. Die Veränderung der Gewichte ist die Methode, die am häufigsten eingesetzt wird. Durch eine Gewichtsänderung können auch andere der genannten Lern-Effekte erzielt werden, zum Beispiel kann eine Verbindung durch Setzen des Gewichts auf null aufgelöst werden. [8, S.53f]

Es werden drei Arten von Lernen unterschieden. Beim *unüberwachten Lernen* lernt das Netz ohne äußere Hilfe. Das Netz bekommt die Eingabedaten und versucht typische Muster zu identifizieren und die Objekte in verschiedene Kategorien einzuteilen. Im *Reinforcement Learning* erhält das Netz nach jedem Durchlauf Information darüber, ob das ermittelte Ergebnis richtig oder falsch ist und kann daraufhin die Gewichte anpassen. Das *überwachte Lernen (Supervised Learning)* ist die dritte Art des Lernens. Dies ist die am häufigsten eingesetzte Methode des Machine Learning. [15, S.436] Hier bekommt das Netz richtige Lösungen präsentiert und kann diese mit dem berechneten Ergebnis vergleichen und daraus Schlüsse über die Veränderung der Gewichte ziehen.

## 2. Grundlagen

---

Um zu vermeiden, dass das Netz nur auswendig lernt, werden die vorhandenen Daten gemischt und aufgeteilt, so dass drei Datensätze entstehen: Trainingsdatensatz, Validierungsdatensatz und Testdatensatz. Der Trainingsdatensatz enthält die meisten Daten, Validierungs- und Testdatensatz sind ungefähr von gleicher Größe. Häufig verwendet wird die Aufteilung: 70% der Daten zum Training und jeweils 15% für die Validierung und den Test. Im Bereich Big Data werden bis zu 95% der Daten zum Training verwendet und der Rest auf die beiden anderen Datensätze gleichmäßig verteilt. Der Trainingsdatensatz wird zum Erlernen der Muster und Zusammenhänge in den Daten verwendet. Mit den Validierungsdaten wird der Lernalgorithmus ausgewählt und passende Parameterwerte ermittelt. Über den Testdatensatz wird das Modell beurteilt, bevor es wirklich zum Einsatz kommt. [7, S.74f]

Es ist nicht ungewöhnlich, dass in einem Neuronalen Netz viele Millionen Gewichte vorhanden sind, über die das Netz justiert werden kann. Die Gewichte werden als Gewichts-Vektoren betrachtet. Während des Trainings berechnet der Lern-Algorithmus für jedes Gewicht den Gradienten in Bezug zu einer vorgegebenen Fehlerfunktion. Durch den Gradienten kann ermittelt werden, ob der Fehler größer oder kleiner wird, wenn das Gewicht erhöht oder reduziert wird. Aufgrund dieser Informationen wird der Gewichtsvektor in die entgegen gesetzte Richtung zum Gradienten-Vektor angepasst. Die Anpassung der Gewichte mit Hilfe der Daten aus dem Trainings-Datensatz erfolgt solange, bis der Mittelwert der Zielfunktion nicht mehr abnimmt. Dies ist ein mögliches Abbruchkriterium, es existieren aber auch noch andere.

Über eine lineare Klassifizierungsfunktion kann die gewichtete Summe der Komponenten des Merkmals-Vektors berechnet werden. Wenn diese gewichtete Summe einen gewissen Schwellenwert überschreitet, kann die Eingabe einer bestimmten Klasse zugeordnet werden. Für die Klassifizierung von Bildern reicht die lineare Klassifizierung nicht aus, weil sich die zu klassifizierenden Objekt in den Bildern hinsichtlich Position, Ausrichtung, Hintergrund und Lichtverhältnissen unterscheiden. [15, S.437f]

Methoden des Deep Learnings liefern hier bessere Ergebnisse als lineare Klassifizierungswerzeuge. Schon durch eine Kombination von 5 bis 20 nicht-linearen Schichten kann das Netz dazu befähigt werden, nicht relevante Merkmale der Bilder, wie zum Beispiel Hintergrund, Position des Objektes und Beleuchtung bei der Klassifizierung zu vernachlässigen und sich auf das eigentliche Objekt zu fokussieren. Über Backpropagation können die Gradienten einer Zielfunktion bezüglich der Gewichte berechnet werden. Unter mathematischen Gesichtspunkten ist Backpropagation die Anwendung von Kettenregeln auf Ableitungen. Im Verfahren der Backpropagation werden die Gradienten rückwärts durch das Netz, vom Ausgabewert bis hin zum Eingabewert propagierte. In einer Deep Learning Architektur wird durch die Verkettung von linearen und nicht-linearen Modulen der Lernprozess erreicht und eine starke nicht-lineare Funktionalität erreicht. [15, S.438]

Zur Bildverarbeitung werden oft Convolutional Neural Networks (CNNs) eingesetzt. CNNs sind Feed-Forward Netze, die in mindestens an einer Stelle Faltungsschichten

statt Matrizenmultiplikation einsetzen. Ein CNN hat die Eigenschaft mit großen Datenmengen umgehen zu können, wie sie in der Bildverarbeitung anfallen, weil es die Datenmenge reduzieren kann, ohne dass dadurch die Qualität des Modells groß vermindert wird. Die Datenmenge in der Bildverarbeitung ist so groß, weil vom Computer jeder Pixel als Merkmal betrachtet wird. Ein Bild besteht aus drei zweidimensionalen Arrays, die Informationen über die Farbtiefe der Pixel in den drei Farbkanälen enthalten. Der Vorteil von Faltungsschichten ist, dass mit vergleichsweise wenig Parameter die translationsinvariante Extraktion geeigneter Merkmale (Merkmale, die nicht von der Position des Objektes im Bild abhängig sind) erlernt werden kann. [15, S.439]

Das CNN funktioniert so, dass über Filter vorhandene Strukturen in den Eingabedaten erkannt werden. In der ersten Schicht werden durch Filter einfache Strukturen wie Linien, Kanten und Farbfragmente ermittelt. Die nachfolgende Schicht kombiniert die einfachen Strukturen zum Beispiel Kurven und simple Formen. Jede weitere Filter-Schicht abstrahiert die Ergebnisse aus dem vorherigen Layer weiter und am Schluss wird eine passende Klasse für das Bild vorhergesagt.

Ein CNN setzt sich im Allgemeinen aus folgenden Schichten zusammen:

- Convolutional Schicht
- Pooling Schicht
- vollständig verbundene Schicht (Fully-connected Layer)

Auf eine oder mehrere Convolutional Schichten folgt in der Regel eine Pooling Schicht. Diese Kombination der zwei Schichten kann in einem CNN mehrfach hintereinander eingesetzt werden. Den Abschluss des Netzes bildet eine vollständig verbundene Schicht. Abbildung 2.2 zeigt die typische Struktur eines CNNs. Die Funktion dieser Schichten wird nachfolgend beschrieben.

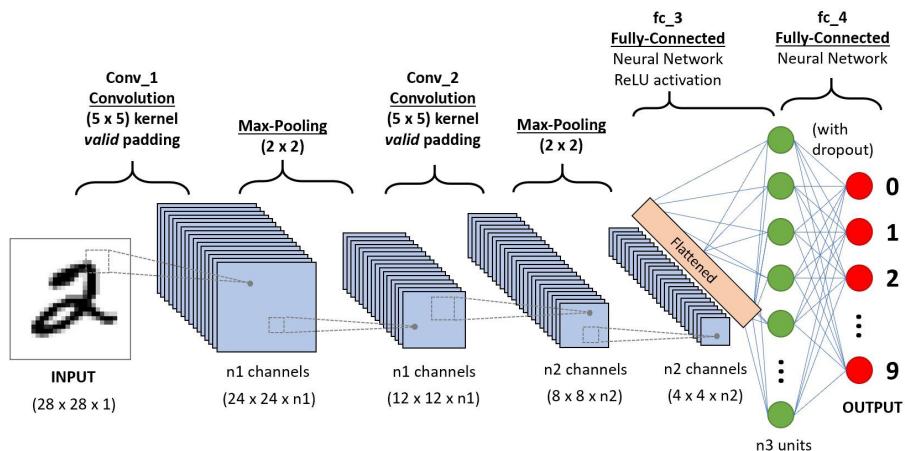
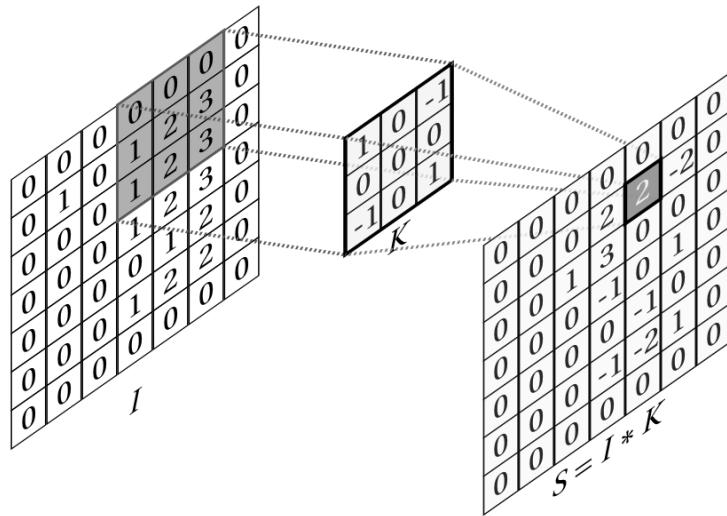


Abbildung 2.2: Eine typische CNN-Architektur.<sup>2</sup>

In der **Convolutional Schicht** findet, wie der Name Convolutional (deutsch: Faltung) schon vermuten lässt, die eigentliche Faltung statt. Diese Schicht wird auch als Filter bezeichnet, weil Filter mit fester Pixelgröße über die Eingabematrix geführt werden. Durch diesen Prozess werden schrittweise einzelne Merkmale wie Linien, Kanten und Formen des Bildes identifiziert. Während des Vorgangs der Faltung wird der Kernel von links nach rechts schrittweise und dann zeilenweise nach unten über die Eingangsmatrix geführt. Wie in Abbildung 2.3 zu sehen ist, wird der  $3 \times 3$  Kernel  $K$  auf ein  $3 \times 3$  Ausschnitt der Eingangsmatrix  $I$  angewendet. Das Ergebnis dieser Operation ist ein  $1 \times 1$  Feld in der Feature-Map  $S$ .



**Abbildung 2.3:** Die Faltung eines zweidimensionalen Arrays:  
Eingabedaten( $I$ ), Kernel/Filter( $K$ ), Feature-Map( $S$ ) [11, S.230]

In der **Pooling Schicht** werden die erkannten Merkmale verdichtet. Beim „Max-Pooling“ wird zum Beispiel nur das jeweils stärkste Signal an die nächste Schicht weitergegeben. Auf diese Weise werden überflüssige Informationen verworfen. In Abbildung 2.2 ist erkennbar, dass die Größe des Inputs durch Pooling immer weiter reduziert wird. Durch die so reduzierte Datenmenge kann die Verarbeitungsgeschwindigkeit erhöht werden. Es gibt verschiedene Pooling-Verfahren, oft verwendet werden Pooling über den Maximalwert oder den Mittelwert.

Der **Fully Connected Layer** ist eine neuronale Netzstruktur, in der jedes Neuron eines Layers mit jedem Neuron des Nachfolge-Layers verbunden ist. In der Praxis wird diese Schicht oft am Ende des Netzes eingesetzt. Sie hat die Aufgabe, die Ergebnisse aus den vorangegangenen Schichten zusammenzuführen und die Klassifizierung des Bildes vorzunehmen. Dazu muss erst das Ergebnis der Convolutional- und Pooling-Layer ausgerollt (*flatten*) werden, damit diese in diesem Layer verarbeitet werden können. Der Fully Connected Layer erzeugt im Kontext der Klassifikation eine Ausgabe, bei der die Anzahl an Neuronen, der Anzahl der zu erkennenden Klassen entspricht. [11, S.228ff]

<sup>2</sup><https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>

## 2.2 Distanzmetriken

Wie im vorherigen Abschnitt gezeigt wurde, werden Bilder vom Computer als Zahlenmatrizen verarbeitet. In den in dieser Arbeit vorgestellten Verfahren werden Heatmaps zur Erklärung der Vorhersagen von Neuronalen Netzen eingesetzt. Heatmaps sind aus Perspektive des Computers auch Bilder, die durch eine Zahlenmatrix repräsentiert werden. Zur Ermittlung der Ähnlichkeit zwischen zwei Heatmaps können mathematische Verfahren eingesetzt werden, wie sie auch in der Cluster-Analyse zum Einsatz kommen. Mit diesen kann die Distanz zwischen zwei Matrizen berechnet werden. Die in dieser Arbeit eingesetzten Verfahren sind die Manhattan-Distanz und die euklidische Distanz. Beide Verfahren werden hier kurz vorgestellt.

Die Manhattan Distanz ist auch unter dem Namen L1-Norm oder City-Block-Metrik bekannt. Der Bezug zu Manhattan ist dadurch gegeben, dass Manhattan als Stadt mit rechtwinkligen Straßenzügen betrachtet wird und die Manhattan Distanz zwischen zwei Punkten nicht die Luftlinie zwischen den beiden Punkten ist, sondern die Summe aus dem waagerechten und dem senkrechten Abstand der zwei Punkte. In mathematischer Ausdrucksweise ist sie definiert als die Summe der absoluten Differenz der Vektoren [9, S.240]:

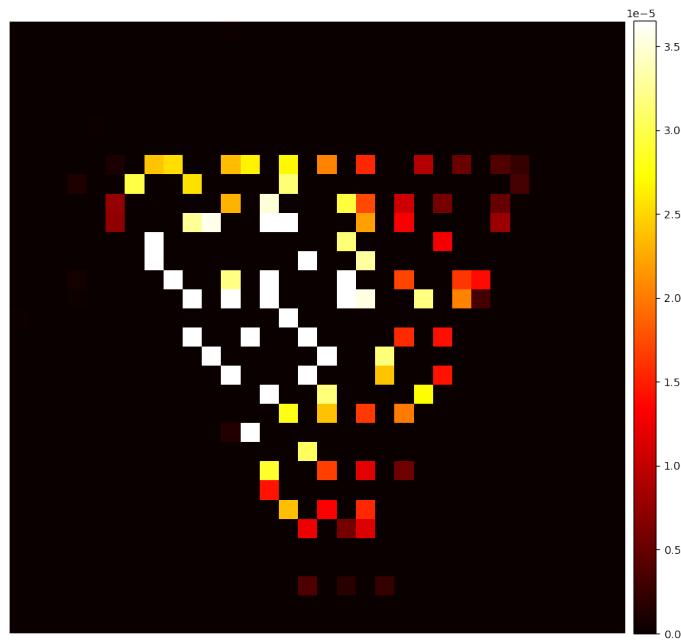
$$\|x - y\|_1 = \sum_{i=1}^n |x_i - y_i| \quad (2.1)$$

Die Euklidische Distanz wird auch als L2-Norm bezeichnet. Diese Distanz umschreibt den kürzesten Weg zwischen zwei Punkten. Im obigen Beispiel würde sie der Luftlinie zwischen den beiden Punkten entsprechen. Sie ist folgendermaßen definiert [9, S.54]:

$$\|x - y\|_2 = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (2.2)$$

## 2.3 Heatmap

Heatmaps werden zur grafischen Darstellung von Daten verwendet. In dieser Arbeit werden Heatmaps zur Visualisierung von Bereichen eines Bildes eingesetzt, die für die Klassifizierung ausschlaggebend sind. Die verwendeten Verfahren zur Erklärung einer Entscheidung eines Neuronalen Netzes, wie Layerwise Relevance Propagation, setzen Heatmaps als Methode ein, um die für die Klassifikationsentscheidung relevanten Pixel grafisch darzustellen. Abbildung 2.4 zeigt das Beispiel einer Heatmap, wie sie im Rahmen dieser Arbeit erzeugt wurde. Auf der linken Seite ist zur Erläuterung eine Farbleiste angezeigt, aus der die Bedeutung der unterschiedlichen Farben abgelesen werden kann. Die schwarzen und dunkelroten Pixel sind in der dargestellten Heatmap weniger relevant für die Vorhersage, die hell-gelben und weißen Pixel haben dagegen eine hohe Relevanz im Klassifizierungsprozess.



**Abbildung 2.4:** Beispiel einer Heatmap mit Farbleiste, aus der die Stärke der Relevanz einzelner Pixel entnommen werden kann.

## 2.4 Erklärbarkeit

### 2.4.1 Eigenschaften von Erklärungen

Vorhersagen von Modellen des Maschinellen Lernens können mit Hilfe von verschiedenen Methoden erklärbar gemacht werden. Es gibt eine Vielzahl von Modellen des Maschinellen Lernens, zum Beispiel Entscheidungsbäume, Support Vector Machines und verschiedene Typen von Neuronalen Netzen. Um eine für den Menschen verständliche Erklärung zu generieren, kann die Beziehung zwischen den Eingabe-Merkmalen und der Vorhersage des Modells verwendet werden. Die Autoren [23, S.162] und [17, Kapitel 2.5] haben zur Bewertung der Qualität von Erklär-Modellen verschiedene Kriterien aufgestellt, von denen die Wichtigsten im Folgenden näher erläutert werden. Ein Bewertungsmodell mit dem der Grad der Einhaltung dieser Kriterien beurteilt werden könnte, existiert bisher noch nicht. [17, Kapitel 2.5]

**Exaktheit** Ein Modell arbeitet exakt, wenn es auch für bisher unbekannte Instanzen gute Vorhersagen erzeugt. Arbeit ein Modell zum Beispiel regelbasiert und die Regeln sind so allgemein formuliert, dass sie auch für bisher unbekannte Instanzen

richtige Vorsagen erzeugen, dann wird die Vorgehensweise des Modells als exakt beschrieben. [23, S.162]

Die Exaktheit ist von Bedeutung, wenn die Erklärung für die Entscheidung eines Modells durch ein anderes, interpretierbares Modell, erzeugt werden soll. Diese Modelle werden auch als Surrogat-Modelle bezeichnet, weil sich ähnlich Verhalten wie das ursprüngliche Modell, aber im Gegensatz zu diesem, erkläbar sind. Dies ist ein Ansatz um Erklärbarkeit zu erzeugen, der in Abschnitt 2.5 näher beschrieben wird.

**Wiedergabtreue** ist ein Maß für die Annäherung der Erklärung an die Klassifikationsentscheidung des Neuronalen Netzes. Wiedergabtreue und Exaktheit liegen nah beieinander. Wenn ein Modell exakte Vorhersagen trifft und die Erklärung von hoher Wiedergabtreue ist, dann ist auch die Erklärung exakt.

Man kann zwischen lokaler und globaler Wiedergabtreue unterscheiden. Bei einer lokalen Wiedergabtreue ist die Annäherung nur für einen Teilbereich passend, die globale Wiedergabtreue umfasst dagegen den gesamten Kontext des Modells.

**Konsistenz** ist gegeben, wenn unterschiedliche Modell, die unter gleichen Bedingungen trainiert wurden, ähnliche Vorhersagen erzeugen. Diese Eigenschaft ist ein Maß für die Gleichartigkeit von Erklärungen.

**Stabilität** Im Gegensatz zur Konsistenz, bei der mehrere Modelle des Maschinel- len Lernens miteinander verglichen werden, werden zur Beurteilung der Stabilität verschiedene Erklärungen eines Modells miteinander verglichen. Die Stabilität eines Verfahrens ist hoch, wenn für ähnliche Instanzen ähnliche Erklärungen erzeugt werden, das heißt, eine geringe Änderung der Eingabewerte sollte nicht zu einer wesentlichen Änderung der Erklärung führen.

**Verständlichkeit** informiert darüber, wie gut die erzeugten Erklärungen von einem Menschen verstanden werden können. Diese Eigenschaft ist wichtig, da sie das Ziel des Erklärungsmodells ist, sie ist aber oft schwer zu messen, weil sie stark vom Wissensstand des Empfängers abhängt.

**Gewissheit** Diese Eigenschaft liefert Informationen darüber, mit welcher Wahrscheinlichkeit die erzeugten Erklärungen einer Methode korrekt sind.

**Grad der Wichtigkeit** Wird durch die erzeugte Erklärung erkennbar, welche Eingabe-Merkmale oder welche Aspekte der Erklärung wichtig sind? Ist zum Beispiel erkennbar welche Variablen und welche Regeln für eine Erklärung am wichtigsten waren?

Leider findet man zu diesen Kriterien in den wissenschaftlichen Arbeiten, die die einzelnen Methoden beschreiben, wenig Informationen. Dies mag zum einen darin begründet sein, dass sich die wissenschaftliche Gemeinschaft der Erklärbaren Künstlichen Intelligenz bisher nicht auf einen allgemeingültigen Kriterienkatalog zur Beschreibung von Methoden geeinigt hat, zum anderen sind die Wissenschaftler oft daran interessiert, ihre Methode möglichst positiv darzustellen. Eine Einschätzung darüber, bis zu welchem Grad jede Methode diese Eigenschaften erfüllt, würde eine genauere Analyse der einzelnen Methoden erfordern.

### 2.4.2 Nutzen von Erklärbarkeit

Nachfolgend werden die Gründe dafür erläutert, warum es vorteilhaft ist, auf Algorithmen beruhende Entscheidungssysteme transparent zu gestalten. [24, S.6ff]

#### 1. Korrektheit

Es soll sichergestellt werden, dass die Klassifizierungsmethode so wie erwartet funktioniert. Wenn das Modell falsche Entscheidungen trifft kann dies gefährlich und teuer sein, zum Beispiel können autonom fahrende Autos einen Unfall verursachen, wenn sie Hindernisse auf der Straße nicht erkennen oder falsch interpretieren. Fehlerhafte Klassifizierungsentscheidungen von KI bei medizinischen Diagnosen, können zu falschen Behandlungsstrategien führen.

#### 2. Vertrauen

Menschen können einem maschinellen Verfahren mehr Vertrauen entgegenbringen, wenn für sie der Ablauf von Entscheidungsprozessen nachvollziehbar ist und wenn erkennbar ist, welche Kriterien eine Entscheidung beeinflussen.

#### 3. Neue Einsichten

Menschen können ihr Wissen durch einen transparenten Entscheidungsprozess eines Modells des Maschinellen Lernens erweitern. So hat zum Beispiel das auf KI beruhende Computerprogramm AlphaGo im März 2016 den weltweit besten Profispielern besiegt. Entscheidend zum Sieg beigetragen haben Spielzüge, die der Fachwelt bisher noch unbekannt waren.<sup>3</sup> Auch im Bereich der Wissenschaft, wie zum Beispiel in Biologie, Chemie und Physik können transparente Entscheidungsprozesse auf KI basierender Systeme zu neuen Einsichten führen.

#### 4. Verbesserung des Modells

Wenn das Vorgehen eines KI-Modells für Menschen nachvollziehbar ist, können Bereiche des Modells ermittelt werden, in denen das Modell noch nicht optimal arbeitet. Zusätzlich kann Expertenwissen aus der entsprechenden Einsatzgebieten von KI an der passenden Stelle in das Modell implementiert werden, um das Modell zu optimieren.

---

<sup>3</sup><https://de.wikipedia.org/wiki/AlphaGo>

Bei der Erzeugung von Transparenz in Neuronalen Netzen können versteckte Merkmale zum Vorschein kommen, die in den Eingangsdaten nicht vorhanden sind und die erst durch das Zusammenwirken von verschiedenen Eingangsdaten erzeugt worden sind. [33, S. 1]

### 5. Einhalten von Gesetzen

Nach der Datenschutz Grundverordnung haben die Menschen in der EU ein Recht auf Erklärung. Transparente Künstliche Intelligenz kann helfen, dieses Recht umzusetzen. Ein Beispiel dafür sind Entscheidungen bei der Kreditvergabe. In diesem Fall hilft Transparenz zu verstehen, warum ein Modell die Entscheidung für oder gegen die Kreditvergabe an eine bestimmte Person gefällt hat.

## 2.5 Unterschiedliche Methoden zur Erzeugung von Erklärungen

### 2.5.1 Klassifizierung von Methoden über die Erklärungen erzeugt werden können

Es existiert eine Vielzahl von Methoden, die sich zum Ziel gesetzt haben, die Prozesse des Maschinellen Lernens erklärbar zu machen und fast täglich erscheinen neue wissenschaftliche Arbeiten, die sich mit dem Thema befassen. Einige Autoren [17, 24] haben versucht eine Taxonomie der Erklärbarkeitsansätze zu erstellen, doch nicht immer sind alle Methoden eindeutig einem Ansatz zuzuordnen. Hier werden zunächst drei Kriterien erläutert, mit denen eine Erklärungsmethode grob klassifiziert werden kann. Die geschilderten Gegensatzpaare werden in [17] erläutert.

#### Intrinsisch oder Post-hoc

Die Erklärbarkeit eines Modells kann auf zwei verschiedene Arten hergestellt werden. Das Vorgehensmodell von intrinsischen Modellen ist auf Grund der einfachen Struktur für einen Menschen leicht nachvollziehbar. Hier ist kein zusätzliches Modell notwendig, das eine Erklärung erzeugt. Zu dieser Kategorie gehören zum Beispiel flache Entscheidungsbäume und einfache lineare Modelle.

Post-hoc Modelle sind in der Regel komplexer. Modelle, die nach diesem Ansatz vorgehen, erzeugen die Erklärung für das ursprünglich Modell, nachdem (Post-hoc) es eine Entscheidung getroffen hat. Zu dieser Kategorie gehören Methoden wie Surrogat Modelle, LIME und Layerwise Relevance Propagation (LRP). Post-hoc-Methoden können auch Erklärungen für intrinsisch interpretierbare Modelle erzeugen.

### **Modell-spezifisch oder Modell-agnostisch**

Modell-spezifische Erklärungsansätze können nur Erklärungen für eine bestimmte Modellklasse erzeugen. Da intrinsisch interpretierbare Methoden immer nur auf ein Modell ausgerichtet sind, sind sie Modell-spezifisch. Darüber hinaus gibt es Methoden, die nur für eine bestimmte Art von Modell funktionieren, zum Beispiel nur für Neuronale Netze.

Modell-agnostische Methoden können für jedes Modell des Maschinellen Lernens Erklärungen erzeugen. Diese Methoden verwenden die Zuordnungen zwischen Eingabe- und Ausgabewerten um die Erklärungen zu generieren. Dazu verwenden sie keine internen Informationen, wie Gewichte und strukturelle Eigenschaften, des zu analysierenden Modells.

Obwohl modellunabhängige Interpretationsmethoden praktisch sind, verwenden sie oft Ersatzmodelle oder erzeugen anderweitig Näherungswerte. Dies kann die Genauigkeit der erzeugten Erklärungen beeinträchtigen. Modellspezifische Interpretationstechniken gründen ihre Erklärungen oft direkt auf dem zu interpretierenden Modell und können daher genauer sein. [12, S. 14]

### **Lokale oder globale Erklärungen**

Mit diesem Kriterium wird unterschieden, ob die Interpretationsmethode das gesamte Modellverhalten erklärt oder nur eine einzelne Vorhersage. Lokale Methoden erläutern die Vorhersage für einen einzelnen Datensatz. Im Gegensatz dazu betrachten globale Modelle das vollständige Modell und versuchen den Entscheidungsprozess als Ganzes zu erfassen.

Globale Erklärungen sind in der Regel weniger genau. Lokale Erklärungen haben den Vorteil, dass für kleine Bereiche des Modells eine Annäherung über eine lineare oder monotone Funktion möglich ist und somit eine höhere Genauigkeit erreicht werden kann. Oft liefert eine Erklärung mittels einer Kombination von lokalen und globalen Interpretationstechniken die besten Ergebnisse. [12, S.13]

### **2.5.2 Verschiedene Erklär-Modelle**

In diesem Abschnitt werden verschiedene Ansätze vorgestellt, über die Erklärungen für die Entscheidung eines Machine Learning Modells erzeugt werden können. Unter diesen Ansätzen sind Methoden zusammengefasst, die ein ähnliches Vorgehen haben. Da einige Methoden verschiedene Lösungsansätze miteinander kombinieren, ist die Zuordnung zu einem bestimmten Ansatz nicht immer eindeutig.

Die hier vorgestellte Klassifizierung von verschiedenen Methoden unter einem Ansatz entspricht der Einteilung, wie sie von Wissenschaftlern des Fraunhofer Heinrich Hertz Institute HHI und der TU Berlin vorgenommen worden ist. Diese Einteilung gibt eine grobe Orientierung, sie ist jedoch nicht in Stein gemeißelt. Wenn man die

## 2. Grundlagen

---

Präsentationen dieser Wissenschaftler in den letzten Jahren betrachtet, sieht man, dass die Einteilung einem ständigen Wandel unterworfen ist.

### **Surrogat Modelle**

Surrogat Modelle sollen eine möglichst gute Annäherung an das zu erklärende Modell sein, aber im Gegensatz zu dem Ausgangsmodell sind sie selbsterklärend. Zur Erzeugung des Surrogate Modells sind keine internen Informationen über das zu erklärende Modell notwendig. Das Surrogat Modell wird nur mit Hilfe der Eingangsdaten und der darauf beruhenden Vorhersage des Ursprungsmodells, erstellt. [17, Kap. 5.6]

Eine vielfach zitierte Surrogat-Methode ist LIME, sie kann zur Erklärung von Klassifikationsentscheidungen durch Neuronale Netze bei Bildern und Texten eingesetzt werden. Weitere Methoden sind Entscheidungsbäume und Entscheidungsregeln. Durch Entscheidungsbäume können Entscheidungen schrittweise nachvollzogen werden. Ein Entscheidungsbaum hat jedoch den Nachteil, dass er sehr groß werden kann, was die Nachvollziehbarkeit wiederum erschwert. Um dieses Problem zu umgehen und die Komplexität der Erklärung zu begrenzen, verwendet die Surrogat-Methode Anchors Entscheidungsregeln für einzelne Instanz-Vorhersagen.

### **Perturbation-basierte Modelle**

Diese Methoden erzeugen Erklärungen über das Einbringen von Störungen (engl. perturbation) in das zu erklärende Modell. Methoden, die über gestörte Eingabewerte arbeiten, werden je nach Schwerpunkt des methodischen Vorgehens in Gradienten-basierte, Störungs-basierte, Propagations-basiert und Optimierungs-basierte Methoden unterteilt.

Die Gradienten-basierten Methoden betrachten das zu erklärende Neuronale Netz als Funktion und erzeugen die Erklärung über den Gradienten. [18, S. 253f] Sie errechnen die Relevanz, die die jeweiligen Eingabemerkmale für die Vorhersage haben, in dem sie den Gradienten der Vorhersage in Bezug zu den Eingangswerten ermitteln. Dabei unterscheiden sich die einzelnen Methoden hauptsächlich dadurch, wie der Gradient durch die nicht-linearen Schichten des Neuronalen Netzes geführt wird. [14]

Gradienten-basierte Methoden haben den Vorteil, dass sie ebenso wie Propagations-basierte Methoden mit einem Vorwärts- und einem Rückwärtsdurchlauf durch das Neuronale Netz auskommen und dadurch relativ wenig rechnerischer Aufwand notwendig ist. [10, S. 158] Nachteilig wirken sich bei diesem Ansatz jedoch nicht-lineare Gradientenverläufe aus. In diesem Fall können missverständliche Relevanzwerte für die Merkmale das Ergebnis sein, wodurch es zu Verzerrungen kommen kann. Darüber hinaus ist es möglich, dass die Relevanz des Beitrags, den ein Merkmal zum Ergebnis geliefert hat, wegen bereits erfolgter Sättigung durch ein anderes Merkmal, unterschätzt wird. [29, S.1ff]

Dieser Effekt, die falsche Beurteilung der Relevanz einzelner Merkmale auf Grund von bestehender Sättigung, tritt auch bei störungs-basierten Ansätzen auf. Viele Methoden, die Störung als Technik verwenden, setzen auf Optimierungsverfahren. [10, S. 158] Die Methoden sind rechnerisch nicht effizient und langsam, da für jede ins Neuronale Netz eingebrachte Störung ein neuer Vorwärtsdurchlauf durch das Netz notwendig ist. [29, S.1ff]

Der Vorteil von störungs-basierten Ansätzen ist ihre leichte Implementierbarkeit und ihre Flexibilität hinsichtlich der zu erklärenden Modellarchitektur. [14] Die Ergebnisse von Störungs-basierten Verfahren sind jedoch nicht immer stabil [29, S.1ff]

### **Propagations-basierte Modelle**

Im Gegensatz zu störungs-basierten Vorgehensweisen sind Methoden die über Propagation arbeiten recheneffizienter, da sie nur wenige Durchläufe durch das Netz benötigen, um eine Erklärung zu erzeugen. Dieser Ansatz funktioniert jedoch nur, wenn das zu erklärende Modell ein Neuronales Netz ist.

Einige Propagation-basierte Verfahren verwenden Gradienten zur Erzeugung der Erklärung. Es gibt verschiedene Methoden, die diese beiden Ansätze kombinieren, zum Beispiel Sensivity Analysis und Deconvolution. Die Methoden, die beide Ansätze kombinieren unterscheiden sich hauptsächlich dadurch, wie sie den Gradienten in der Backpropagation durch die nicht-linearen Schichten des Neuronalen Netzes führen. [29, S.2]

Viele propagations-basierte Methoden haben den Nachteil, dass ein Eingriff in das Ausgangsmodell notwendig ist. [10, S. 159] Bei einigen Methoden ist eine Änderung der Backpropagation-Regeln erforderlich (Guided Backpropagation, Deconvolution, Network Dissection), andere Modell (LRP, Grad-CAM, Excitation Backpropagation und Network Dissection) benötigen einen Zugriff auf die Zwischenschichten des Modells.

Layerwise Relevance Propagation (LRP) ist eine Methode, die in der Fachwelt viel Aufmerksamkeit hervorgerufen hat. Es besteht eine gewisse Nähe zu Methoden wie DeepLift, Excitation Backpropagation, Gradient Times Input und Deep Taylor Decomposition einige Methoden, die eine ähnliche Vorgehensweise haben. Gegenüber Gradienten-basierten Ansätzen, hat LRP den Vorteil, dass die erzeugte Heatmaps leichter zu interpretieren sind. Wenn Gradienten-basierte Ansätze verwendet werden, können die Ergebnisse stark verrauscht und dadurch schwer zu interpretieren sein.

Ein weiterer Vorteil von LRP gegenüber Gradienten-basierten Ansätzen, ist, dass LRP auch die Merkmale in der Heatmap hervorheben kann, die einen negativen Einfluss auf die Vorhersage haben. Gradienten-basierte Ansätze liefern nur Informationen über Merkmale, die die Vorhersage unterstützen, Aussagen zu Merkmalen von negativem Einfluss werden nicht erzeugt. [19, S. 6]

Nach Einschätzung von [5, S. 2] weisen Gradienten-basierte Verfahren nur darauf hin, wie empfindlich das Modell auf Änderungen reagiert, während LRP die eigentliche Ursache für eine erfolgte Klassifizierung aufzeigt. Dadurch ist LRP seiner Ansicht nach den Gradienten-basierten Ansätzen und den Methoden, die auf Entfaltung beruhen, wie zum Beispiel Deconvolution und Guided Backpropagation, überlegen.

Der Autor von [14] sieht jedoch auch Kritikpunkte bei den Propagtions-basierten Ansätzen. Nach seiner Einschätzung sind die Erklärungen von LRP und DeepLift nicht immer zuverlässig. Die von DeepLift erzeugten Erklärungen sind stark von dem vom Nutzer gewählten Referenzpunkt abhängig und die Ergebnisse von LRP können numerisch instabil sein.

## 3 Erklär-Modelle

### 3.1 Anchors

Anchors wurde von den Entwicklern Marco Tulio Ribeiro, Sameer Singh und Carlos Guestrin im Jahr 2018 veröffentlicht. Dasselbe Forscherteam hatte zuvor den LIME-Algorithmus entwickelt. Anchors erklärt die Vorhersagen eines Neuronalen Netzes über Entscheidungsregeln, mit denen Vorhersagen verankert werden. Eine Entscheidungsregel ist dann in der Lage eine Vorhersage zu verankern, wenn sich die Vorhersage nicht ändert, wenn Änderungen an den Merkmalswerten erfolgen. Anchors kombiniert Techniken des Reinforcement Learnings mit Suchalgorithmen, die auf Graphen beruhen, um die Anzahl der Modellaufrufe zu minimieren und dadurch die Laufzeit zu reduzieren. [17, Kapitel 5.8]

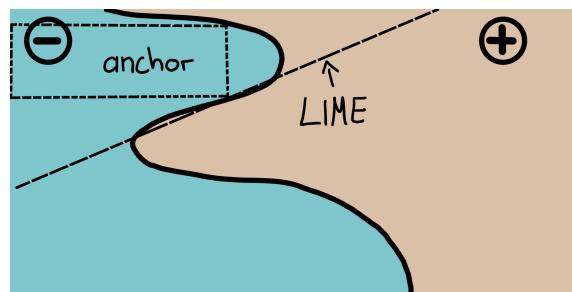
Die Verankerung der Erklärung des Algorithmus durch Regeln beruht auf leicht verständlichen *wenn-dann*-Entscheidungen. Diese Regeln decken einen gewissen Geltungsbereich ab und sind in diesem Rahmen wiederverwendbar. Für welchen Bereich eine Regel gilt wird als Abdeckung (*coverage*) bezeichnet. Der Bereich, den eine Anker-Regel abdeckt wird eindeutig definiert. Die Regeln sind nur gültig, wenn alle definierten Bedingungen erfüllt sind. Sind diese erfüllt, dann haben die Regeln eine hohe Präzision. [22, S.1f]

Der Prozess des Findens eines Ankers entspricht dem bekannten Dilemma des mehrarmigen Banditen. Bei diesem Dilemma geht es darum einen möglichst guten Kompromiss zwischen Exploration und Exploitation zu erzielen. Die Aufgabe ist es die beste Option aus einer Auswahl verschiedener Optionen zu wählen (Exploration/Erforschung), bei denen jede Option einen unterschiedlichen Grad von Gewinn be-

deutet (Exploitation/Verwertung). Dabei ist es das Ziel, den höchsten Gewinn zu erzielen.

Im Rahmen der Erzeugung der Regeln für eine zu erklärende Instanz wird das Umfeld der Instanz betrachtet. In dieses Umfeld werden Störungen eingebracht. Auf diese Weise arbeitet der Algorithmus unabhängig vom Aufbau und den verwendeten Parametern der zu erklärenden Modell-Struktur. Daher ist Anchors modellunabhängig und kann für jede Art von Modell angewendet werden. [17, Kapitel 5.8]

Die Abbildung 3.1 zeigt den Unterschied zwischen Anchors und LIME. Aufgabe ist es, die Vorhersage für eine binären Entscheidung (+ oder -) zu treffen. LIME erzeugt eine lineare Entscheidungsgrenze zwischen den zwei Zuständen, hier dargestellt als gestrichelte Linie. Demgegenüber erzeugt Anchors eine Erklärung, die dem Modell besser angepasst ist und die Grenzen des Modells berücksichtigt (hier als Rechteck zu sehen.) Anchors macht die Grenzen seines Geltungsbereichs deutlich und ist für den Menschen leicht zu verstehen.



**Abbildung 3.1:** Unterschiedliche Erklärung einer binären Entscheidung von LIME und Anchors [17, Kapitel 5.8]

Wie schon erwähnt setzt der Anchors-Algorithmus Regeln ein, um Erklärungen zu generieren. Wie so eine Regel aussieht und welche Informationen sie beinhaltet, soll an einem Beispiel erklärt werden. Betrachtet wird die Aufgabe mit Hilfe eines Neuronalen Netzes die Ausfallwahrscheinlichkeit von Krediten bei Personen, die verschiedene Merkmale erfüllen, vorherzusagen. Anchors wird eingesetzt um die Erklärung zu liefern, warum das Netz eine spezifische Vorhersage getroffen hat. Eine erklärende Regel in Anchors könnte folgende Struktur haben:

**Listing 3.1:** Beispiel einer Anchor-Regel

```
Anchor = [ 'Kapitalertrag == Kein' ,
           'Geschlecht == Maennlich' ,
           'Beruf == Beamter' ,
           'Wochenarbeitszeit < 50' ,
           'Alter < 60' ]
```

```
Abdeckung = 0,15
Genauigkeit = 0,97
```

Die Beispiel-Regel zeigt, welche Attribute in dem Modell ausschlaggebend sind. Im Beispiel sind es der Kapitalertrag, das Geschlecht, der Beruf, die Wochenarbeitszeit und das Alter. Die Anker-Regel liefert zusätzlich die Informationen, dass sie für 15% der Instanzen des Störungsraums gilt und in diesen Fällen die erzeugte Erklärung zu 97% richtig ist, d.h. die genannten Attribute sind ausschlaggebend für die Vorhersage. [17, Kapitel 5.8]

Damit die Anker-Regeln möglichst aussagekräftig sind, sollten bei dem Erzeugen der Regeln einige Grundsätze beachtet werden. Es wird angestrebt Regeln zu finden, die für einen möglichst großen Teil der Eingabewerte gültig sind. Regeln, die eine hohe Abdeckung haben, die also einen großen Teil des Modells beschreiben, haben eine höhere Bedeutung in dem Algorithmus, als Regeln, die nur eine geringe Abdeckung haben.

Bezüglich der Genauigkeit der Regeln ist davon auszugehen, dass Regeln, die sich auf mehr Attribute stützen, genauere Ergebnisse liefern, als Regeln, die nur wenige Attribute beinhalten. Genauigkeit und Abdeckung sind wichtige Eigenschaften der Regeln. Bei der Erstellung von Regeln muss ein Kompromiss zwischen diesen beiden Eigenschaften gefunden werden. Eine sehr spezifische Regel, in der viele Merkmale festgelegt sind, kann nur auf wenige Instanzen angewendet werden, sie kann aber genauere Ergebnisse liefern. [17, Kapitel 5.8]

Der Prozess des Findens von Erklärungen besteht auf vier wichtigen Komponenten:

1. **Mögliche Regeln generieren** Im ersten Durchgang wird ein möglicher Regel-Kandidat pro Merkmal der zu erklärenden Instanz erzeugt. Im zweiten Durchgang werden die besten Regeln aus dem ersten Durchgang mit einem weiteren Merkmal versehen. Bei jedem weiteren Durchgang kommen neue Merkmale hinzu.
2. **Identifizierung der besten Regel** Regeln werden unter dem Gesichtspunkt verglichen, welche Regel die zu erklärende Instanz am besten beschreiben. Hier tritt das Dilemma des mehrarmigen Banditen auf. Jede Regel wird als ein Arm (eine Option) angesehen und wenn eine Regel bedient wird, dann wird das Ergebnis, dass über die anderen Arme entsteht, ausgewertet. Dadurch können Informationen über den Vorteil, der aus einer gezogenen Option entsteht, ermittelt werden. Der Vorteil ist in diesem Fall die Genauigkeit der Vorhersage.
3. **Validierung der Genauigkeit der Regeln** Wenn die angestrebte Genauigkeit noch nicht erreicht ist, erfolgen weitere Durchläufe.
4. **Modifizierte Strahlensuche (Beam-Search)** Alle obigen Komponenten werden mit einem Beam-Search-Verfahren ausgewertet. Beam-Search ist ein Suchalgorithmus der auf Graphen beruht. Es werden die  $B$  besten Kandidaten von jeder Runde in die nächste übernommen. Mit Hilfe dieser Regeln werden neue Regeln erstellt. Da jedes Merkmal der zu erklärenden Instanz nur einmal in einer Regel enthalten sein darf, werden maximal so viele Runden des

Beam-Search ausgeführt, wie es Merkmale gibt. Es werden in jeder Runde  $i$  Regel-Kandidaten erzeugt, mit genau  $i$  Prädikaten. Davon werden die  $B$  besten ausgesucht. Ein hoher  $B$ -Wert hilft lokale Optima zu vermeiden. Dadurch steigt aber die Anzahl der notwendigen Modellaufrufe, was zu einem erhöhten Rechenaufwand führt. [17, Kapitel 5.8]

Der Anchors-Algorithmus ist ein gutes Konzept, um nachvollziehen zu können, warum ein Modell eine bestimmte Klassifizierung vorgenommen hat. Der Algorithmus setzt anerkannte und erforschte Methoden des Maschinellen Lernens ein, wodurch die Anzahl der erforderlichen Durchgänge des Modells verringert werden kann. Dadurch wird die Laufzeit des Algorithmus reduziert. [17, Kapitel 5.8]

Anchors eignet sich besonders zur Interpretation von Verfahren, die Text- und Tabellendaten klassifizieren. Auch zur Erklärung von Bildklassifizierungen kann die Methode eingesetzt werden. Da es in dieser Arbeit hauptsächlich um die Klassifizierung von Bildern geht, wird auf die anderen Anwendungsbereiche nicht näher eingegangen.

Zur Vorhersage der richtigen Klasse für ein Bild, wird das Bild zunächst in verschiedene Bereiche aufgeteilt. Dieser Bereich wird Superpixel genannt, da er aus mehreren Pixeln bestehen. Bei der Segmentierung des Bildes bleibt die lokale Bildstruktur erhalten. Um nützliche Erklärungen zu erhalten, sind aussagekräftige Superpixel notwendig. Superpixel, die nicht in den Anker integriert werden, werden maskiert, indem sie entweder den Mittelwert des Superpixels erhalten oder mit einem anderen Bild überlagert werden. Der Anker in der Bildklassifizierung besteht aus einer Maske, die aus Superpixeln erstellt wurde.<sup>1</sup>

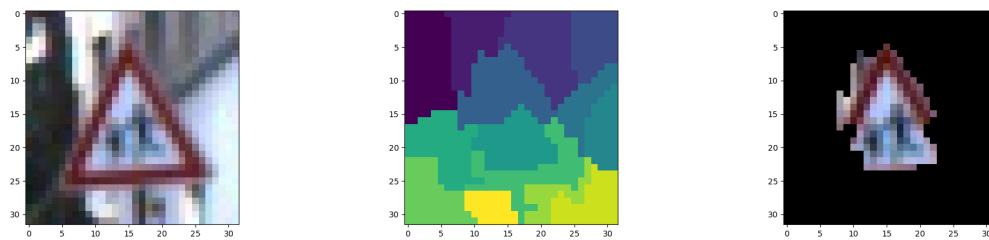
Die Interpretation des Bildes erfolgt über das Vorhandensein oder das Fehlen bestimmter Superpixel. Über die stufenweise Abdeckung einzelner Superpixel wird ermittelt, welche Bereiche (Superpixel) relevant für eine bestimmte Klassifizierung sind. Die relevanten Superpixel bilden dann den Anker für die korrekte Klassifizierung.[22, S.1f]

Am Beispiel der Erklärung der Klassifizierung eines Verkehrsschildes über ein Neuronales Netz soll die Vorgehensweise von Anchors erläutert werden. Wie in Abbildung 3.2 zu sehen ist, wird zunächst über eine bestimmte Methode das Bild in Superpixel aufgeteilt. In der Python-Implementierung von Anchors zur Aufteilung des Bildes in Superpixel kann zwischen den Methoden Felzenswalb , Slic und Quickshift gewählt werden. Die unterschiedlichen Farben der Superpixel in der Mitte der Abbildung enthalten keine Informationen über die Stärke der Relevanz für die Vorhersage, sie dienen nur der Unterscheidung der Superpixel. Auf der rechten Seite der Abbildung sind die beiden Superpixel, die für die Klassifizierung höchste Relevanz haben, dargestellt.

Die Suche nach dem relevantesten Superpixel erfolgt in Anchors so, dass schrittweise ein oder mehrere Superpixel des Bildes abgedeckt werden und auf dieser Grundlage

---

<sup>1</sup><https://docs.seldon.io/projects/alibi/en/stable/methods/Anchors.html>



**Abbildung 3.2:** Anchors: Originalbild (links), Aufteilung in Superpixel (Mittel) und die relevantesten Superpixel (rechts).

ermittelt wird, wie sich die Änderungen des Bildes auf die Vorhersagewahrscheinlichkeit für die korrekte Klasse auswirken. Bei diesem Schritt werden viele Kombinationen von abgedeckten Superpixeln ausprobiert und jeweils die Vorhersagewahrscheinlichkeit berechnet. In Abbildung 3.3 sind einige dieser Zwischenschritte auf dem Weg zur Ermittlung des relevantesten Superpixels dargestellt.



**Abbildung 3.3:** Anchors: Schrittweise Ermittlung der relevantesten Superpixel.

Für die Bilder in den Abbildungen 3.2 und 3.3 wurde die Python-Implementierung<sup>2</sup> verwendet. Hier kann Anchors über verschiedene Parameter dem eigenen Projekt angepasst werden. Zum Beispiel ist es möglich über einen Schwellenwert anzugeben, wie viele Bilder der Algorithmus zur Validierung eines Superpixels verwenden soll. Hier muss man einen Kompromiss finden, zwischen dem Maß der Zuverlässigkeit des Ankers und der Höhe der Rechenzeit finden.

Vorteile von Anchors:

- Die erzeugten Erklärungen sind für einen Menschen leicht nachvollziehbar, da sie auf Regeln basieren, die gut zu interpretieren sind.
- Die Regeln liefern Informationen zum Abdeckungsbereich und dadurch auch zur Bedeutung von einzelnen Merkmalen.
- Mit den Regeln können auch komplexere Umgebungen erfasst werden.
- Anchors funktioniert modellunabhängig.
- Der Algorithmus ist effizient und erlaubt parallele Verarbeitungsstränge.

---

<sup>2</sup>[https://docs.seldon.io/projects/alibi/en/stable/examples/anchor\\_image\\_imagenet.html](https://docs.seldon.io/projects/alibi/en/stable/examples/anchor_image_imagenet.html)

Nachteile von Anchors:

- Anchors hat eine hohe Konfigurierbarkeit, daher muss die Methode den jeweiligen Anwendungsszenarien angepasst werden, dies kann eine aufwändige und nicht immer ganz leichte Aufgabe sein. Einmal gefundene optimale Einstellungen sind nicht unbedingt übertragbar auf andere Einsatzszenarien.
- Abhängig vom verwendeten Datensatz kann es notwendig sein, eine diskrete Auswahl an Daten zu treffen, damit das Resultat nicht zu spezifisch wird, wodurch das Verständnis des Ergebnisses beeinträchtigt werden kann. Um eine gute Auswahl treffen zu können, ist es notwendig die Daten zu kennen.
- Zur Ermittlung der Anker-Regeln sind viele Aufrufe des Modells und damit ein erhöhter Rechenaufwand notwendig.

## 3.2 Layerwise Relevance Propagation

Layerwise Relevance Propagation (LRP) ist eine Erklärungstechnik für Modelle von Neuronalen Netzen. Die Untersuchungsgegenstände des Modells können zum Beispiel Texte, Bilder oder Videos sein. Zur Erzeugung der Erklärung nutzt die Methode lokale Propagations-Regeln mit denen sie die Vorhersage  $f(x)$  des Modells rückwärts durch das Netz hindurch leitet. [27, S.195]

Nachdem ein Netzwerk trainiert wurde ist es für die Entwickler oft hilfreich zu verstehen, wie die Vorhersage des Netzes zustande gekommen ist. Wie zuvor in Abschnitt 2.5 erläutert wurde gibt es verschiedene Ansätze, um dieses Problem zu lösen.

Ein Lösungsansatz zur Bewertung eines Modells des maschinellen Lernens basiert auf der Ermittlung der Merkmale des Eingabebildes, die für die Vorhersage des Netzes von Bedeutung sind. Um diese zu ermitteln, gibt es verschiedene Möglichkeiten. Dies kann zum Beispiel durch die Beobachtung und Analyse der Veränderung der Vorhersage, nachdem bestimmte Bereiche des Bildes ausgeblendet wurden, geschehen.

Wenn der Ausgangspunkt ein Modell ist, dass darauf trainiert wurde eine gewisse Anzahl von Klassen, zum Beispiel verschiedene Verkehrsschilder, zu erkennen und das Modell gut trainiert wurde, dann ist es in den meisten Fällen in der Lage das Verkehrsschild richtig zu klassifizieren. Aber wie ist diese spezielle Vorhersage zustande gekommen? Hat das Netz tatsächlich das Objekt im Bild erkannt oder hat sich das Modell auf den Kontext des eigentlichen Objekts konzentriert und ist über den Rückschluss über die Umgebung auf das Objekt gekommen. Dies kann der Fall sein, wenn das Modell in der Trainingsphase gelernt hat, dass im Fall eines entsprechenden Kontextes auch immer ein bestimmtes Objekt vorhanden ist. Zum Beispiel kann der Grund für die richtige Klassifizierung eines Schneeleoparden darin liegen,

dass das Netz im Training gelernt hat, dass das Vorkommen eines Tier-ähnlichen Objekts im Schnee, häufig die Klasse Schneeleopard ergibt.

Man kann dies prüfen, indem man Teilausschnitte des ursprünglichen Bildes dem Netz zur Vorhersage gibt. Ein Durchgang in dem der Kontext des Objekts abgedeckt wurde und ein Durchgang in dem das Objekt abgedeckt wurde. Durch Vergleich der Vorhersage des Modells für die beiden Teil-Bilder kann man ermitteln, ob das Objekt auch ohne Kontextinformationen richtig klassifiziert wird, ob die Kontextinformationen alleine ausreichen, um das Objekt vorherzusagen.

Wie gezeigt wurde, kann es bedeutsam sein zu erkunden, auf welche Informationen sich eine Klassifizierung durch eine Neuronale Netz stützt. Layerwise Relevance Propagation (LRP) ist eine Technik, mit der ermittelt werden kann, welche Eingabemerkmale am stärksten zur Ausgabe eines Neuronalen Netzes beigetragen haben. Die Methode wurde von einem Entwickler-Team des Fraunhofer Heinrich Hertz Institute und der Technischen Universität Berlin entwickelt. [2]

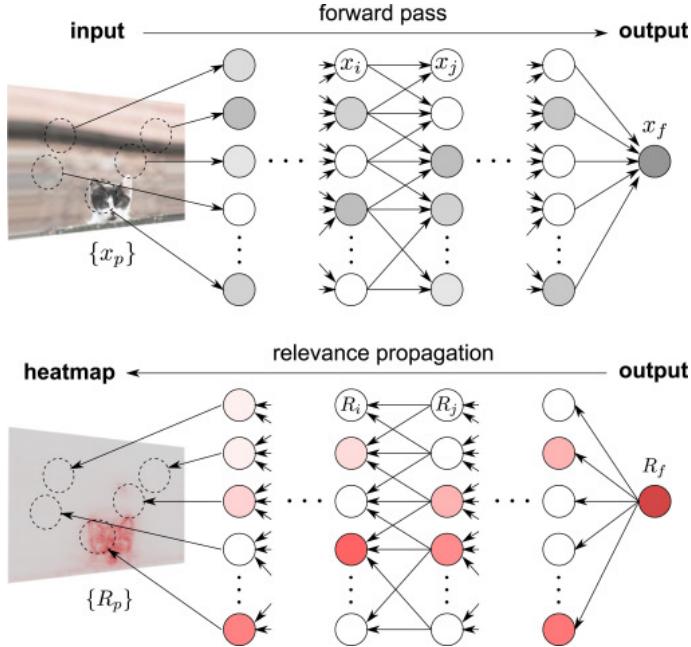
Zur Bestimmung der für die Ausgabe bedeutenden Eingabemerkmale verwendet die LRP-Methode ein schrittweises Vorgehen. Im Kontext der Klassifizierung von Bildern, wird der Beitrag jedes einzelnen Pixels des Bildes  $x$  für die Vorhersage  $f(x)$  ermittelt. Für jedes Bild wird berechnet, welches Pixel in welchem Ausmaß zu einem positiven oder einem negativen Klassifizierungsergebnis beigetragen hat. So kann ein Relevanz-Maß  $R$  für den Eingabevektor bestimmt werden, indem die Summe aller Relevanz-Maße der Eingabe-Pixel gebildet wird. [2, S. 3]

Das Ziel von LRP ist für jedes Pixel  $p$  eines Bildes  $x$  über eine Klassifizierungsfunktion  $f$  einen Relevanz-Wert  $R_p$  zu ermitteln:

$$f(x) \approx \sum_p R_p \quad (3.1)$$

Haben die Pixel  $p$  einen Relevanzwert unter null  $R_p < 0$  ist dies ein Hinweis für eine fehlende Zugehörigkeit zu einer Klasse und bei  $R_p > 0$  spricht dies für eine Zugehörigkeit zu einer bestimmten Klasse. Die pixelweisen Relevanzwerte eines Bildes können über eine Heatmap visualisiert werden. [2, S.2f]

Dabei ist  $f$  der Vorwärtsdurchlauf durch das Neuronale Netz. Wenn es sich bei dem Eingabewert in das Neuronale Netz um ein Bild handelt, kann der Ausgabewert als Summe der Relevanzwerte der einzelnen Pixel des Eingabebildes betrachtet und zerlegt werden. Diese Zerlegung (Dekomposition) kann vorgenommen werden, indem die Relevanz des Ausgabewertes rückwärts durch das gesamte Neuronale Netz propagiert wird. Das zugrundeliegende Prinzip dieser Vorgehensweise ist eine einfache Taylor Dekomposition, die die Entscheidung eines Modells erklärt, indem sie die zugrundeliegende Funktion als Summe von Relevanzwerten betrachtet. [19, S.4] Die Taylor Dekomposition erzeugt Erklärungen für die Vorhersage  $f(x)$  indem eine Taylor-Erweiterung für einen naheliegenden Bezugspunkt  $x$  durchgeführt wird. [27, S.194]



**Abbildung 3.4:** Berechnung der Deep Taylor Decomposition. Im oberen Bild findet der Prozess der Klassifikation des Bildes statt, im unteren Bild wird durch die Backpropagation die Erklärung erzeugt. [20, S.215]

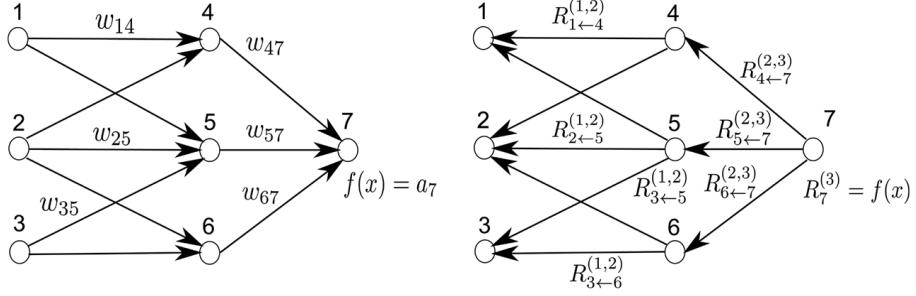
Die Klassifizierung des Eingangsbildes als „Katze“ wird im Vorwärtsdurchlauf durch das Neuronale Netz ausgehend von den Pixelwerten  $\{x_p\}$  durchgeführt und resultieren im Ausgangsbild  $x_f$ . Die Ausgabe hat einen Relevanzwert von  $R_f$ , dieser repräsentiert die Gesamt-Evidenz für die Klasse „Katze“.

Diese Relevanz wird im Rückwärtsdurchlauf (Backpropagation) durch das gesamte Netz bis zur Eingabe geführt. Die Relevanz  $\{R_p\}$  enthält die Relevanzwerte der einzelnen Pixel, die in ihrer Gesamtheit als Heatmap dargestellt werden können. [20, S.215]

Das Modell geht davon aus, dass auf jedem Layer die Gesamtsumme der Relevanzwerte gleichbleibt, nur die Verteilung der Relevanzwerte auf die einzelnen Knoten ist von Layer zu Layer unterschiedlich. Bei der Betrachtung von Abbildung 3.4 kann man ersten Eindruck davon bekommen, was unter Relevanz zu verstehen ist. Die Relevanz  $R$  ist der lokale Beitrag eines Knoten des Neuronalen Netzes zur Vorhersagefunktion  $f(x)$ .

Die folgende Abbildung 3.5 zeigt eine Klassifizierungsfunktion mit Neuronen und gewichteten Verbindungen zwischen diesen. Für jedes Neuron  $i$  wird über die Aktivierungsfunktion eine Ausgabe  $a_i$  ermittelt. Das linke Bild zeigt das Neuronale Netz im Vorhersageprozess. Hier werden die Gewichte  $w_{ij}$  zwischen den Knoten  $i$  und dem Knoten  $j$  dargestellt. Dabei bezeichnet  $a_i$  die Aktivierung des Neurons  $i$ .

Im rechten Bild wird die Berechnung der schichtweisen Relevanzwerte dargestellt.  $R_i^{(k)}$  ist die Relevanz des Neurons  $i$  im Layer  $k$ . Mit dem Ausdruck  $R_{i \leftarrow j}^{(k, k+1)}$  wird die



**Abbildung 3.5:** Ermittlung der Relevanzwerte in der Backpropagation. [2, S.5]

Beschreibung der Berechnung von der Relevanz  $R_i^{(k)}$  verdeutlicht.  $k$  ist die Variable für den Layer, dementsprechend umschreibt  $k + 1$  den Nachfolgelayer. Der Term  $i \leftarrow j$  bildet die Beziehung zwischen zwei Neuronen ab, in diesem Fall ist  $i$  das Eingangsneuron für  $j$ . Der Pfeil geht von rechts nach links, weil dies die Richtung der Ermittlung der Relevanzwerte ist.

Das rechte Bild in Abbildung 3.5 zeigt also, wie die Relevanzwerte zwischen den Knoten der einzelnen Layer berechnet werden. Während der Backpropagation muss die Relevanz eines Neurons  $n$  auf dem Layer  $k$  zurück auf die Neuronen des vorherigen Layers  $k-1$  propagiert werden. Dazu ist eine Vektor-Funktion notwendig, die folgende Informationen benötigt, um die Relevanz für jedes Neuron  $n_i$  auf dem vorherigen Layer zu berechnen:

- die Relevanz des Neurons  $n$
- die Aktivierung  $x$  dieses Neurons,
- die Aktivierungen  $x_i$  der Neuronen  $n_i$  im vorherigen Layer  $k-1$  und
- die Matrix der Gewichte  $w$ , die die Verbindung zwischen  $n$  und  $n_i$  abbildet.

Dabei sollte die Funktion Neuronen, die einen höheren Einfluss auf das Neuron  $n$  haben auch mit einem größeren Gewicht ausstatten.

Es existieren verschiedene Versionen des LRP-Algorithmus, sie haben aber alle die Eigenschaft, dass die Gesamtrelevanz auf den unterschiedlichen Layern erhalten bleibt. Die verschiedenen Versionen unterscheiden sich hauptsächlich in der Art, wie die Relevanz-Werte für die einzelnen Knoten berechnet werden. Zur Berechnung gibt es verschiedene Formeln, die im Folgenden näher erläutert werden.

Die ursprüngliche Version von LRP, wie sie in [2] beschrieben ist und die Grundlage für die in Zusammenhang mit dieser Arbeit durchgeführten Untersuchungen ist, erläutert zwei verschiedene Formeln, um die Relevanz von einem Layer zu dem vorherigen Layer zu berechnen. Sie werden als die  $\epsilon$ -Formel und die  $\beta$ -Formel bezeichnet.

Die  $\epsilon$ -Regel ist folgendermaßen definiert:

$$R_{i \leftarrow j}^{(k,k+1)} = \frac{z_{ij}}{z_j + \epsilon \cdot \text{sign}(z_j)} R_j^{(k+1)} \quad (3.2)$$

Die Variable  $z_{ij}$  bildet ab, in welchem Ausmaß das Neuron  $i$  die Relevanz von Neuron  $j$  gesteigert hat. [27, S.195] Berechnet wird dies wie folgt:

$$z_{ij} = (w_{ij}x_i)^p \quad (3.3)$$

$$z_j = \sum_{k:w_{kj} \neq 0} z_{kj} \quad (3.4)$$

Bei dieser Berechnung wird die Variable  $\epsilon$  als Stabilisator eingesetzt, wenn  $z_j$  einen sehr kleinen Wert in der Nähe von Null einnimmt.

Die zweite Art der Berechnung erfolgt über die  $\beta$ -Regel:

$$R_{i \leftarrow j}^{(k,k+1)} = \left( (1 + \beta) \frac{z_{ij}^+}{z_j^+} - \beta \frac{z_{ij}^-}{z_j^-} \right) R_j^{(k+1)} \quad (3.5)$$

Bei dieser Regel werden die negativ- und postiv-gewichteten Aktivierungen getrennt betrachtet. Die Variable  $z_{ij}^{+/-}$  beinhaltet den positiven/negativen Beitrag, den ein Knoten  $i$  zum Knoten  $j$  beigetragen hat. Diese einzelnen Beiträge werden durch die Summe aller positiven/negativen Beiträge der Knoten eines Layers  $l$  geteilt:

$$z_j^{+/-} = \sum_i z_{ij}^{+/-} \quad (3.6)$$

Dadurch ist es möglich, dass die Relevanz von Layer  $l+1$  zu Layer  $l$  erhalten bleibt. In Formel 3.5 kann über  $\beta$  der Einfluss positiver Beiträge reguliert werden. Bei einem  $\beta$ -Wert von Null werden nur positive Beiträge in der Heatmap angezeigt, während  $\beta$ -Werte, die ungleich Null sind, die negativen Beiträge, die gegen eine bestimmte Klassifizierung sprechen, korrigieren. [5, S.4]

Der  $\beta$ -Wert beeinflusst auch die Schärfe des Bildes der Heatmap. Ein Wert von  $\beta = 1$  wird als hoch betrachtet. Bei diesem Wert ist eine gute Bildschärfe in der Heatmap gegeben. [4, S.3]

Die hier vorgestellten  $\epsilon$ -Regel und die  $\beta$ -Regel wurden von [4] im Jahr 2016 vorgestellt. In neueren Publikationen der Autoren des LRP-Algorithmus [27] werden weitere Regeln eingeführt, die auch miteinander kombiniert werden können, um bessere Erklärungen für die Ergebnisse eines Neuronalen Netzes erzeugen zu können. Im Folgenden werden die drei grundlegenden Regeln dieser Veröffentlichung näher erläutert.

**Basic-Regel (LRP-0)** ist die grundlegende Regel, aus der die später vorgestellten Regeln abgeleitet werden:

$$R_j = \sum_k \frac{a_j w_{jk}}{\sum_{0,j} a_j w_{jk}} R_k \quad (3.7)$$

Dies Regel erfüllt die Eigenschaft, dass im Fall von einem fehlenden Gewicht  $w_{jk} = 0$  oder einem deaktivierten Knoten  $a_j = 0$  auch die Relevanz  $R_j = 0$  ist. Die mit dieser Regel erzeugte Erklärung ist jedoch oft zu komplex und ist nicht auf das eigentliche Objekt konzentriert. In der mit dieser Regel erzeugten Heatmap werden viele Bereiche angezeigt, die außerhalb des eigentlichen Objekts liegen. [27, S.196, 200]

**Epsilon-Regel (LRP- $\epsilon$ )** Diese Erweiterung der Basic-Regel wurde bereits oben vorgestellt und wird hier noch einmal in einer leicht veränderten Form präsentiert. Der einzige Unterschied zur Basic-Regel (LRP-0) ist der zusätzlich positive  $\epsilon$ -Wert im Nenner des Bruchs.

$$R_j = \sum_k \frac{a_j w_{jk}}{\epsilon + \sum_{0,j} a_j w_{jk}} R_k \quad (3.8)$$

Durch den  $\epsilon$ -Wert werden Elemente, die Rauschen erzeugen, in der Erklärung entfernt. Die Erklärung fokussiert sich auf eine begrenzte Anzahl von Merkmalen, die zum tatsächlichen Objekt gehören. Je größer der  $\epsilon$ -Wert gewählt wird, desto blasser wird die Darstellung der relevanten Werte in der mit dieser Regel erzeugten Heatmap. Die Gefahr besteht, dass zu wenige Merkmale dargestellt werden und dadurch die erzeugte Erklärung nicht mehr verständlich ist. [27, S.196-197, 200]

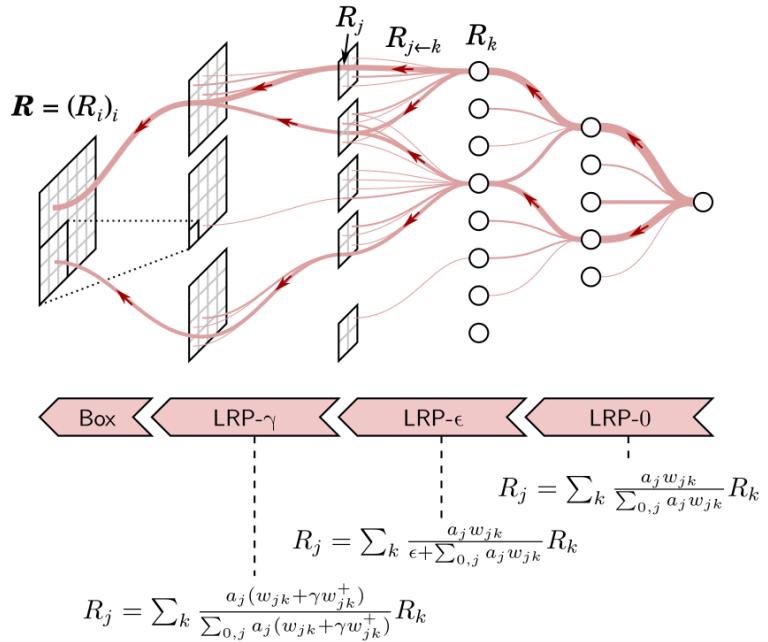
**Gamma-Regel (LRP- $\gamma$ )** Bei dieser Regel werden positive Beiträge höher gewichtet als negative, was zu einer weiteren Verbesserung der Ergebnisse führt.

$$R_j = \sum_k \frac{a_j \cdot (w_{jk} + \gamma w_{jk}^+)}{\sum_{0,j} (w_{jk} + \gamma w_{jk}^+)} R_k$$

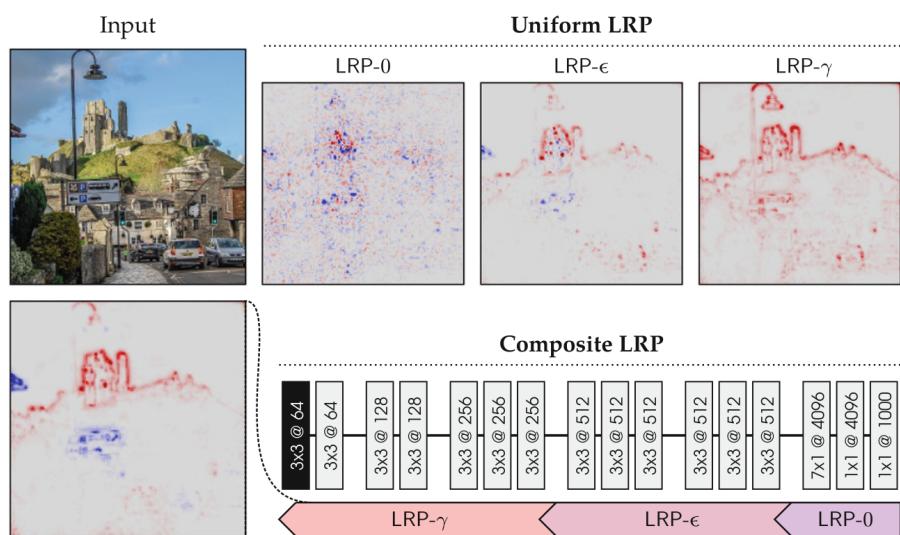
Diese Regel verteilt die Beiträge auf der unteren Schicht so um, dass positive Beiträge gegenüber negativen Beiträgen bevorzugt werden. Wenn positive Beiträge überwiegen, wird die Ausprägung von positiven und negativen Beiträgen in der Propagierungsphase begrenzt. Dies macht die Erklärungen robuster. Die so erzeugte Heatmap ist für den Menschen leichter zu verstehen, da relevante Merkmale stärker hervorgehoben werden. Es werden jedoch auch nicht-relevante Bereiche, die außerhalb des eigentlichen Objekts liegen, wieder stärker sichtbar. [27, S.197, 200]

In Abbildung 3.6 ist zu sehen, wie die Vorhersage im Rückwärts-Durchlauf durch das Neuronale Netz geleitet werden und dabei auf den verschiedenen Layern die erläuterten Regeln angewendet werden. Der Verlauf der Propagation zur Ermittlung der Relevanzwerte ist in der Abbildung rot dargestellt.

Über die LRP-Parameter  $\epsilon$  und  $\beta$  und die Auswahl der Regel, mit der die Relevanzwerte berechnet werden, kann LRP individuell für ein konkretes Projekt justiert werden. Hinsichtlich der Auswahl der passenden Regel bietet sich an, sich die Vorteile der einzelnen Regeln zu Nutzen zu machen und verschiedene Regeln miteinander zu kombinieren. So können unterschiedliche Regeln für unterschiedliche Layer eingesetzt werden. Dieses Vorgehen wird von den Entwicklern von LRP als **Composite LRP** bezeichnet. [27, S. 200]



**Abbildung 3.6:** Der Verlauf der Propagation (rote Pfeile) in einem Neuronalen Netz unter Verwendung verschiedener Propagations-Regeln. [28, S. 252]



**Abbildung 3.7:** Einsatz unterschiedlicher LRP-Regeln auf den verschiedenen Layern. [27, S. 200]

Abbildung 3.7 zeigt, welchen Einfluss die unterschiedlichen LRP-Regeln auf die jeweilige Heatmap haben und wie durch die Kombination von verschiedenen LRP-Regeln auf den unterschiedlichen Layern die für einen Menschen verständlichste Erklärung erzeugt werden kann.

Auf den oberen Layern (rechts in Abbildung 3.7) sind die verschiedenen Klassen noch eng miteinander verwoben und die Neuronen Anzahl ist noch relativ gering. Hier bietet es sich an die LRP-0 Regel einzusetzen, da sie am nächsten an der Vorhersagefunktion und deren Ableitung ist.

Auf den mittleren Layer kann es besonders durch den Einsatz von Convolutional Layern zu Abweichungen (spurious variations) kommen, die am besten mit der LRP- $\epsilon$  Regel herausgefiltert werden können, damit nur die für eine Erklärung wichtigsten Faktoren ausgewählt werden.

Für die niedrigeren Layer ist die LRP- $\gamma$  Regel am besten geeignet, weil diese Regel dafür sorgt, dass die Relevanz mehr gestreut wird und der Beitrag jedes einzelnen Pixels nicht so stark hervortritt. Dies trägt zu einer verständlicheren Erklärung bei. [27, S. 200]

## 3.3 Theoretischer Vergleich von Anchors und LRP

Beide Verfahren gehören zur Gruppe von Algorithmen, die lokale Interpretierbarkeit herstellen, weil sie die Gründe für eine spezifische Entscheidung oder eine einzelne Vorhersage herausarbeiten.

Anchors erstellt für ein Bild eine Vielzahl von neuen Bildern, die Kombinationen von verschiedenen Superpixel enthalten. Auf Grundlage dieser neuen Bilder ermittelt der Algorithmus, ob die korrekte Klasse vorhergesagt wird und wie sich die verschiedenen Kombinationen von Superpixelen auf die Vorhersagewahrscheinlichkeit auswirken. Dazu muss für jedes neu erstellte Bild das Netz durchlaufen werden. Dies wirkt sich auf die Rechenzeit aus. In der verwendeten Anchors-Implementierung kann der Entwickler über das Setzen der entsprechenden Parameter die Anzahl der zu erzeugenden Bildern und die Anzahl der Superpixel, in die das Bild aufgeteilt wird, bestimmen. Diese Parameter haben Einfluss auf die Genauigkeit der Vorhersage und auf den Rechenaufwand.<sup>3</sup>

Demgegenüber braucht LRP für ein Bild nur einen Vorwärts- und ein Rückwärtsdurchlauf durch das Netz. Dadurch ist die Rechenzeit für ein Bild wesentlich kürzer als bei Anchors. (Im Versuch benötigte Anchors ca. 1 Minute 14 Sekunden für die Erzeugung von Heatmaps für 8 Bilder. LRP benötigte ca. 42 Sekunden für dieselbe Anzahl an Bildern. Anchors kann 8 Bilder parallel verarbeiten, die Berechnung für ein Bild bei Anchors dauerte auch ca. 1 Minute 14 Sekunden.)

---

<sup>3</sup><https://docs.seldon.io/projects/alibi/en/stable/methods/Anchors.html>

Hinsichtlich der visuellen Ausgabe der Methode erzeugt Anchors ein Bild in dem der Bereich, der für die Vorhersage die größte Relevanz hat als Superpixel dargestellt wird. LRP dagegen erzeugt eine Heatmap, in der die Relevanz jedes Pixels hervorgehoben wird. Die Vorhersage ist dadurch wesentlich genauer, als bei Anchors.

Zusätzlich werden in der durch LRP erzeugte Heatmap die Pixel andersfarbig dargestellt, die gegen die getroffene Vorhersage sprechen. Dadurch können zum Beispiel verschiedene Objekte in einem Bild unterschieden werden. Diese Informationen können auch hilfreich im medizinischen Bereich bei der Interpretation von Heatmaps, die gescannte Organe darstellen. Diese Funktionalität ist in Anchors nicht vorgesehen.

## 4 Daten und Methoden

### 4.1 Daten und Vorverarbeitung

Für die im Rahmen dieser Arbeit durchgeführten Untersuchungen wurde der Datensatz verwendet, der unter dem Namen German Traffic Sign Recognition Benchmark (GTSRB) bekannt ist.<sup>1</sup> Der Datensatz besteht aus mehr als 50.000 Farbbildern von deutschen Straßenschildern. Es sind 43 unterschiedliche Schilder in dem Datensatz enthalten. Die verschiedenen Aufnahmen eines Schildes sind jeweils in einer Klasse zusammengestellt. [30, S.1]

Die Daten sind unterteilt in einen Trainingsdatensatz, bestehend aus knapp 35.000 Bildern, einen Testdatensatz mit ca. 12.700 Bildern und einen Validierungsdatensatz mit ca. 4500 Bildern. Die Bilder sind farbig und haben jeweils eine Größe von 32 x 32 Pixeln. In Abbildung 4.1 wird jeweils ein Bild für jede Klasse aus dem GTSRB-Datensatz gezeigt.

### 4.2 Netzwerkarchitektur

Zur Klassifizierung der Daten wurde ist eine Convolutional Neural Network (CNN)-Architektur eingesetzt. CNNs sind Neuronale Netze, die sich besonders für die Berechnung von Matrizen und die Verarbeitung von großen Datenmengen eignen, wie sie in der Bilderkennung häufig vorkommen. Das für die Untersuchung verwendete

---

<sup>1</sup>[https://benchmark.ini.rub.de/gtsrb\\_news.html](https://benchmark.ini.rub.de/gtsrb_news.html)

<sup>2</sup>[https://www.researchgate.net/figure/An-example-of-the-43-traffic-sign-classes-of-GTSRB-dataset\\_fig9\\_311896388](https://www.researchgate.net/figure/An-example-of-the-43-traffic-sign-classes-of-GTSRB-dataset_fig9_311896388)



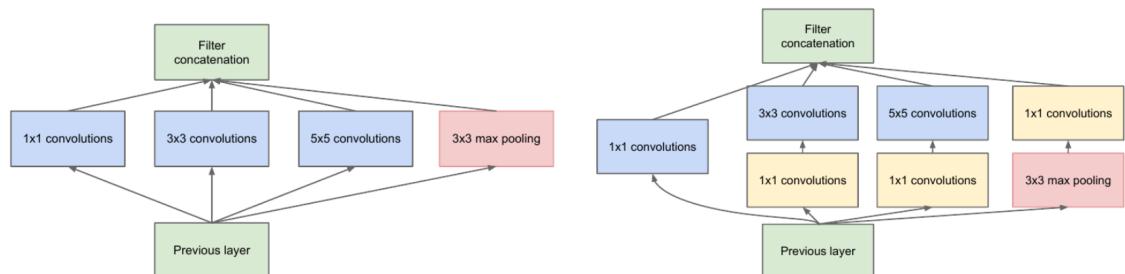
**Abbildung 4.1:** Die verschiedenen Klassen des GTSRB-Datensatzes.<sup>2</sup>

Neuronale Netz wurde vom Bundesamt für Sicherheit in der Informationstechnik (BSI) zur Verfügung gestellt.

Um die Leistung des CNN hinsichtlich Geschwindigkeit und Genauigkeit zu steigern wurde ein Inception Modul in das Netz eingebaut. Ein Inception Modul besteht aus einer Kombination unterschiedlicher Faltungsoperationen. [31, S.2ff] Die Verwendung dieses Moduls hat verschiedene Vorteile:

- Für die Eingabedaten können über verschiedene Faltungsoperationen lokale und globale Features ermittelt werden. Dies geschieht durch unterschiedliche Filtergrößen in parallel angeordneten Schichten.
- Die Anzahl der Faltungen mit größeren Filtern kann durch die Verwendung von  $1 \times 1$  Filtern reduziert werden. Dadurch werden die erlernbaren Gewichte reduziert und das Training beschleunigt.

Zum besseren Verständnis wird in Abbildung 4.2 das Prinzip eines Inception Moduls dargestellt: Das der Untersuchung zugrunde liegende Modell wurde mit einem Netz



**Abbildung 4.2:** Das Prinzip des Inception Moduls, links: die einfache Version, rechts: größere Filter werden durch  $1 \times 1$  Filter reduziert. [31, S.4]

trainiert, das aus folgenden Layern besteht: zu Beginn ist das Inception Modul, dann folgen drei Kombinationen von Max-Pooling Layer mit anschließendem Batch-Convolution-Modul. Den Abschluss bildet wieder ein Max-Pooling Layer.

Das Inception Modul zu Beginn besteht wiederum aus sieben Batch-Convolution Modulen. Die Batch-Convolution Module sind jeweils zusammengesetzt aus einer Convolutional-Schicht, einer Batch-Normalisierungsschicht und einer ReLU-Schicht. Die Batch-Normalisierungsschicht wird zur Standardisierung der Daten eingesetzt. Die ReLU-Schicht ist eine nicht-lineare Aktivierungsfunktion, die alle negativen Werte auf null setzt. Alle Werte größer null bleiben unverändert. [7, S.93]

Das Netz hatte nach dem Training mit dem GTSRB-Trainingsdatensatz eine Genauigkeit (Accuracy) von 98,7%.

Neben dem beschriebenen Inception Netz wurde bei der Qualitätsuntersuchung in 5.3 noch ein einfacheres Netz zusätzlich verwendet, um die Arbeitsweise des LRP-Algorithmus auf verschiedenen Netzen untersuchen zu können. Das einfache Netz besteht aus vier Convolutional Schichten und drei linearen Layern. Das Netz hatte nach dem Training eine Accuracy von 85,3%.

Die genaue Struktur der beiden Netze wird im Anhang in 7.1 und 7.2 dokumentiert.

## 4.3 LRP-Implementierung

Zur Erklärung der Klassifizierungs-Entscheidung des eingesetzten Netzes wurde die LRP-Implementierung aus dem Github-Repository<sup>3</sup> von Moritz Böhle verwendet. Diese LRP-Implementierung basiert auf der PyTorch-Programmbibliothek und wurde für dieses Projekt der Klassifizierung von Verkehrsschildern angepasst.

Bei dieser Implementierung kann die LRP-Methode über die vier folgenden Parameter eingestellt werden: *Methode*, *LRP-Exponent*, *Epsilon-Wert* und *Beta-Wert*. Bei der Methode kann zwischen *basic-rule* und *epsilon-rule* gewählt werden. Diese beiden Regeln wurden bereits in Abschnitt 3.2 näher erläutert. Zur weiteren Einstellung kann für die *epsilon-rule* der *LRP-Exponent* und der *Epsilon-Wert* angepasst werden. Der *LRP-Exponenten* dient der Skalierung der Wichtigkeitswerte pro Knoten. Dies ist der Wert  $p$  in der Formel 3.3.<sup>4</sup> Der *Epsilon-Wert* wird zur Stabilisierung eingesetzt, damit numerische Instabilität vermieden werden kann. Diese kann auftreten, wenn der Nenner in Formel 3.2 einen Wert nahe null einnimmt.

Bei der *basic-rule* kann nur der *Beta-Wert* skaliert werden. Ein hoher *Beta-Wert* verstärkt die Aktivierung von Knoten, die einen positiven Einfluss auf die Aktivierung des Knotens der Nachfolge-Schicht haben. Ist dieser Wert niedrig, werden die hemmenden Neuronen einer Schicht mehr betont.<sup>5</sup> Bei einem *Beta-Wert* von null werden nur die positiven Beiträge in der Heatmap angezeigt. [5, S.4]

---

<sup>3</sup><https://github.com/moboehle/Pytorch-LRP>

<sup>4</sup>[https://github.com/moboehle/Pytorch-LRP/blob/master/inverter\\_util.py](https://github.com/moboehle/Pytorch-LRP/blob/master/inverter_util.py), Zeile 51 und 390/391

<sup>5</sup><https://github.com/moboehle/Pytorch-LRP/blob/master/investigator.py>, Zeile 33ff

## 5 Untersuchungen zu LRP

Der naheliegendsten Aspekte um ein Erklär-Model zu beurteilen, ist der Nutzen, der mit seinem Einsatz verbunden ist. Um den Nutzen der durch den Einsatz von LRP beurteilen zu können, kann die Methoden anhand verschiedener Fragestellungen untersucht werden:

- **Menschliche Interpretierbarkeit:** kann der Mensch aus der erzeugten Erklärung die Entscheidungsstrategie des Netzes nachvollziehen?
- Wie hoch ist die **Wiedergabetreue** der Methode? Führt eine Neutralisierung relevanter Pixel zu einem starken Rückgang der Netz-Vorhersagbarkeit?
- **Stabilität:** Wie robust ist die Erklärung, wenn das Bild in eine beliebige Richtung verschoben wird oder wenn es gespiegelt oder rotiert wird?
- **Parameter-Einstellungen:** Welchen Einfluss haben verschiedene Parameter-Einstellungen auf die erzeugten Erklärungen?
- **Implementierungen:** Wie unterscheiden sich die unterschiedlichen Implementierungen von LRP? Welchen Unterschied gibt es hinsichtlich der erzeugten Erklärungen und der Laufzeit?
- Gibt es **verschiedene Arten von Erklärungen**? Kann man zwischen verschiedenen Heatmaps wählen und wenn ja, wie unterscheiden sie sich? Sind außer Heatmaps noch andere Arten der Erklärung durch LRP möglich?
- **Anwendbarkeit:** Ist die Methode auf unterschiedliche Modelle anwendbar?
- **Laufzeit:** Wie hoch ist die rechnerische Komplexität und wie lange braucht der Algorithmus um eine Vorhersage zu erstellen?

Aus Zeitgründen konnte nur ein Teil dieser Fragestellung in dieser Arbeit untersucht werden. Im Folgenden werden vier Untersuchungen mit LRP geschildert, die Rahmen dieser Arbeit durchgeführt wurden.

Zu Beginn wurde analysiert, ob und wie sich eine Pixel-Manipulation im Originalbild auf die zur Erklärung eingesetzte Heatmap auswirkt. In der zweiten Untersuchung wurde betrachtet, ob sich die Erklärung durch eine Verschiebung des Originalbildes verändert. Im dritten Teil dieses Kapitels werden für die Aufgabe der Erklärung der Verkehrsschild-Klassifizierung passende LRP-Parameter Ermittlungen ermittelt. Dies wurde für zwei Netze mit unterschiedlicher Architektur durchgeführt. Zuletzt wird untersucht, wie sich das Einbringen eines Aufklebers (Patch) auf das Bild des Verkehrsschildes, auf die Erklärung durch LRP auswirkt.

## 5.1 Auswirkungen von Pixelmanipulationen im Eingangsbild

### Hypothese 1

Wenn in dem Originalbild einzelne Pixel manipuliert werden, wird dadurch die Erklärung der Vorhersage beeinflusst. Die Manipulation kann sowohl visuell von einem Menschen durch Betrachtung der Heatmap, als auch rechnerisch, über die Distanzmetriken erkannt werden. Es wird erwartet, dass die Ergebnisse der visuellen Wahrnehmung und der Berechnung der Distanzmetrik miteinander korrelieren.

Die erste Untersuchung wurde durchgeführt um die Stabilität von LRP zu analysieren. Auf Grundlage des Verkehsschilderdatensatzes GTSRB sollte ermittelt werden, ob Störungen, die in die zu klassifizierende Bilder eingebracht wurden, Einfluss auf die über LRP erzeugten Erklärungen haben.

Ausgangspunkt ist ein Netz, dass auf Grundlage des Trainings mit dem GTSRB-Datensatz, eine Auswahl von Bildern aus dem Testdatensatz, der jeweils richtigen Klasse zuordnet. Mit Hilfe der Python-Implementierung für LRP wurde für jedes Bild eine Heatmap erzeugt, um zu ermitteln worauf die Klassifizierung des Netzes beruht. Die Heatmap hebt die für die Klassifizierung relevanten Pixel hervor.

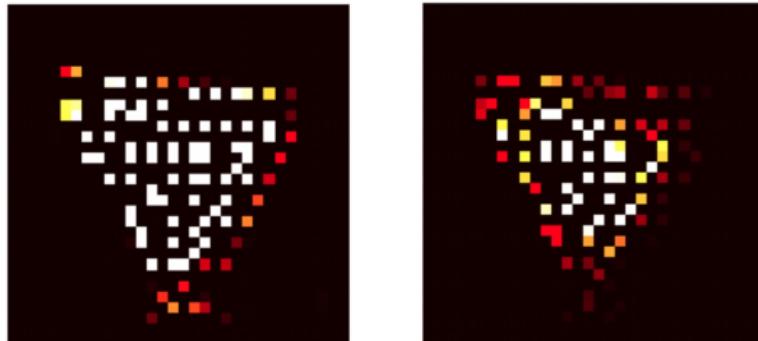
Für jedes verwendete Originalbild aus dem Trainingsdatensatz wurde ein zweites Bild erzeugt, bei dem einige Pixel gestört wurden. Die Anzahl der gestörten Pixel wurde zufällig gewählt. Dazu wurden zwischen 50 und 300 Pixeln zufällig gestört. Die Anzahl der maximal gestörten Pixel wurde auf 300 begrenzt, weil sonst die Gefahr bestanden hätte, dass das Netz das Bild nicht mehr richtig klassifizierte.

Die Störung erfolgte dadurch, dass der Wert von einzelnen Pixeln auf null gesetzt wurde. Die so erzeugten Heatmaps wurden als Bild abgespeichert und zusätzlich auch im csv-Format, um die Relevanzwerte der einzelnen Pixel der Heatmap für die weitere Berechnung zur Verfügung zu haben.

Die Idee war die visuellen Auswirkungen der eingebrachten Störung von mehreren Menschen im Rahmen einer Umfrage beurteilen zu lassen. Zusätzlich sollte die Auswirkung der eingebrachten Störung auch rechnerisch erfasst werden.

Für ein Bild Paar, bestehend aus Originalbild und Bild mit gestörten Pixeln, wurde jeweils eine Heatmap erstellt. Insgesamt wurden auf diese Weise 21 Paare von Heatmaps erzeugt. Aus diesen 21 Paaren von Heatmaps wurde ein Fragebogen für eine Umfrage erstellt. In dieser hatten die Teilnehmer der Umfrage die Aufgabe, die Ähnlichkeit der jeweiligen Heatmap-Paare zu beurteilen. Die Umfrage-Teilnehmer sollten die Ähnlichkeit der beider Heatmap mit einer Zahl zwischen 1 (=niedrig) und einer 5 (=hoch) bewerten. Abbildung 5.1 zeigt einen Ausschnitt aus der Umfrage, in dem eins von 21 Bildpaaren und die Bewertungsmöglichkeit dargestellt ist.

**9. Bild (24-25)**



Bewertung der Ähnlichkeit (1 = niedrig, 5 = hoch)

zur Bewertung bitte eine Zahl aus dem Dropdown-Menu auswählen

**Abbildung 5.1:** Ein Bild Paar aus der Umfrage.

17 Personen haben den Fragebogen erhalten, von 11 Personen wurde der Fragebogen ausgefüllt zurückgegeben. Die zurückgegebenen Fragebögen wurden ausgewertet.

Die erstellten csv-Dateien mit den Relevanzwerten der einzelnen Pixel der Heatmaps wurden zur rechnerischen Ermittlung der Ähnlichkeit verwendet. Die dazu eingesetzten Methoden waren die Metriken Manhattan-Distanz (L1) und euklidische Distanz (L2). Aus der Summe der Mittelwerte der Umfrageergebnisse pro Heatmap-Paar und der Summe der jeweiligen Distanzmetriken wurde der Pearson-Korrelationskoeffizient berechnet. Dieser lag für die L1-Metrik bei -0,581 und für die L2-Metrik bei -0,611. Also in beiden Fällen eine negative Korrelation.

Die Abbildungen 5.2 zeigen die Korrelationsgerade und die Streuung der Werte für die 21 Heatmap-Paare. Ermittelt wurden auch die Formel für die beiden Regressionsgeraden für die L1-Distanz und die L2-Distanz:

$$L1 : f(x) = -3,0698x + 4,2735 \quad (5.1)$$

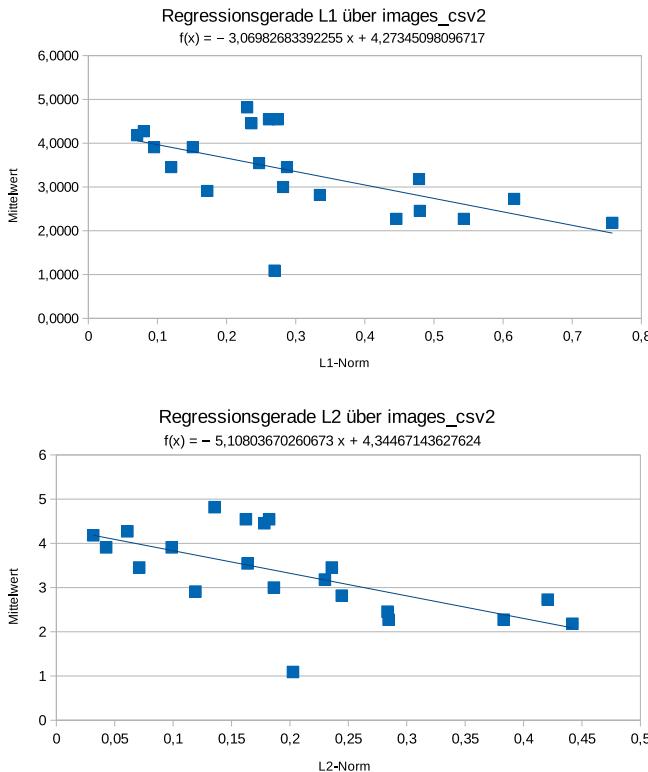
$$L2 : f(x) = -5,1080x + 4,3447 \quad (5.2)$$

Zwei Variable korrelieren dann negativ miteinander, wenn das Ansteigen einer Variablen das Absteigen der anderen Variablen zur Folge hat.<sup>1</sup> Bezogen auf diese Untersuchung bedeutet dies, dass ein höherer Mittelwert bei der Bewertung der Ähnlichkeit zwischen zwei Heatmaps zu einem niedrigen Wert der Distanz-Metrik führt. Dies würde in dieser Untersuchung Sinn machen. Wenn der Mittelwert hoch ist, bedeutet dies, dass die Umfrage-Teilnehmer eine höhere Ähnlichkeit zwischen den beiden Heatmaps wahrnehmen. In diesem Fall müsste die Distanz, hier repräsentiert durch die L1- und L2-Metrik, gering sein.

<sup>1</sup><https://de.statista.com/statistik/lexikon/definition/77/korrelation/#:text=Eine%20Korrelation%20misst%20die%20St%C3%A4rke,Korrelation%20E2%80%9Eje%20mehr%20Variable%20A%E2%80%A6>

## 5. Untersuchungen zu LRP

---



**Abbildung 5.2:** oben: Manhattan-Distanz (L1)  
unten: euklidische Distanz (L2)

Allerdings sprechen die errechneten Werte von -0,581 und -0,611 nicht für eine hohe Korrelation, sondern nur für eine mäßige Korrelation, wenn man davon ausgeht, dass Werte zwischen 0,5 und 1 auf eine „mäßige bis starke“ Korrelation hinweisen und die errechneten Werte im unteren Teil dieser Spanne liegen. [16, S.119f]

Es kann vermutet werden, dass es einen Zusammenhang zwischen der visuellen Beurteilung der Ähnlichkeit zwischen zwei durch LRP erzeugte Heatmaps und der mathematisch berechneten Distanz gibt. Unabhängig davon, hat die Untersuchung gezeigt, dass die in das Bild eingebrachten Störungen zu veränderten Heatmaps geführt haben. Dies konnte sowohl rechnerisch, als auch visuell wahrgenommen werden.

Denkbar ist, dass die eingebrachte Manipulation des Bildes sich zwar auf die Relevanz einzelner Pixel auswirkt, es aber nicht genügt, sich bei der Interpretation der Erklärung nur auf die Relevanz einzelner Pixel zu konzentrieren. Es könnte der Fall sein, dass nicht nur die Relevanz einzelner Pixel eine Aussagekraft hat, sondern auch die Kombination verschiedener relevanter Pixel für die Erklärung bedeutsam wäre.

In den weiteren Untersuchungen wurde zu Ermittlung der Ähnlichkeit von Matrizen nur die Distanz-Metriken verwendet und auf weitere Umfragen verzichtet, da Umfragen mit einem höheren Aufwand verbunden sind. Die Berechnung von Distanz-Metriken kann schnell durchgeführt werden und ohne großen Aufwand leicht wie-

derholt werden, wenn man die Untersuchungsparameter im Nachhinein verändern muss.

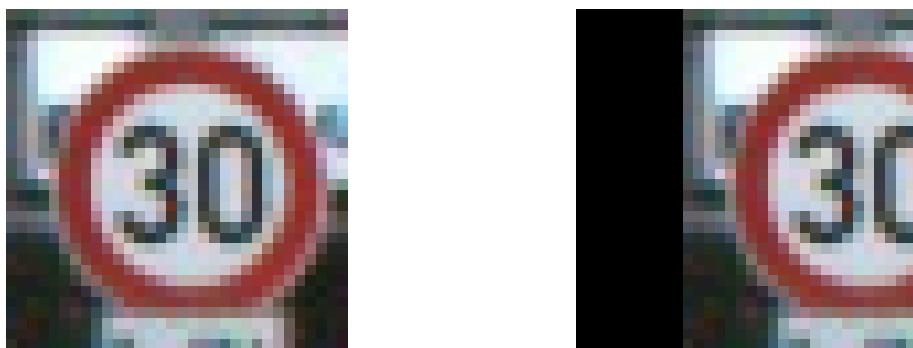
## 5.2 Robustheit von LRP

### Hypothese 2

Wenn das Bild des Verkehrsschildes um  $n$  Pixel verschoben wird, werden auch die für die Vorhersage relevanten Pixel gleichermaßen verschoben. Die Erklärung der Vorhersage des Neuronalen Netzes ändert sich dadurch nicht.

Ziel dieser Untersuchung ist es, zu ermitteln, wie robust die LRP-Heatmap ist, wenn sich die Position des Bildes ändert. Es soll analysiert werden, ob die relevanten Pixeln in der Heatmap des Originalbildes, mit den relevanten Pixeln der Heatmap übereinstimmen, die nach einer Transformation des Bildes von LRP ermittelt wurden.

Im Detail wurde wie folgt vorgegangen. Zunächst wurde über das LRP-Verfahren die Heatmap für das Originalbild berechnet. Das Originalbild wurde anschließend transformiert, indem es um einige Pixel nach rechts verschoben wurde. In der durchgeführten Untersuchung wurde das Bild um 10 Pixel nach rechts verschoben. Abbildung 5.3 zeigt das Originalbild und das verschobene Bild.



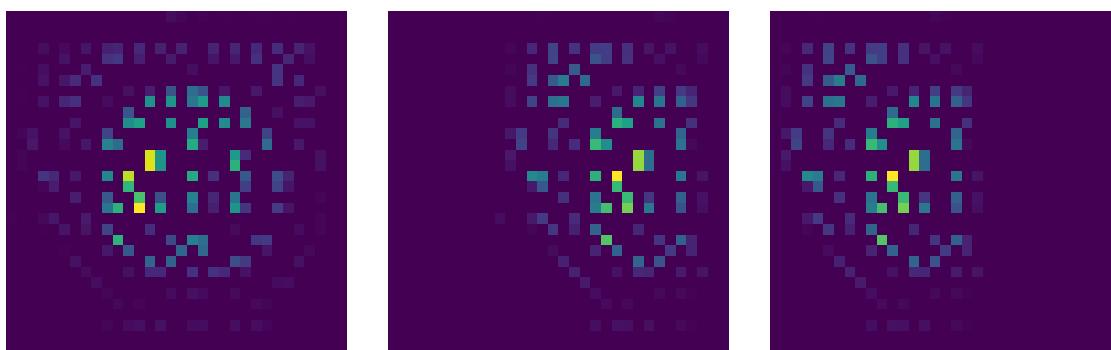
**Abbildung 5.3:** links: das Originalbild; rechts: das transformiert Bild

Die durch die Verschiebung zustande gekommene leere linke Spalte wurde mit Nullen aufgefüllt. In Abbildung 5.3 ist dieser Bereich schwarz dargestellt. Für das so transformierte Bild wurde eine weitere Heatmap erstellt. Um eine Vergleichbarkeit mit der Heatmap des Originalbildes zu ermöglichen wurde die Heatmap des transformierten Bildes zurücktransformiert, so dass sie mit der Ausgangsposition des ursprünglichen Bildes übereinstimmt.

Da nun die Heatmap des Ausgangsbildes und die zuletzt durch die Verschiebung erzeugte Heatmap verglichen werden sollten, war es zuvor notwendig die rechte Spalte, die durch den Transformationsprozess verändert wurden, bei beiden Bildern abzu-

schneiden, damit sichergestellt ist, dass sich die Relevanz-Werte der verschobenen Heatmap wieder auf derselben Position befinden, wie im Ausgangszustand, bei der Heatmap für das ursprüngliche Bild.

In der Abbildung 5.4 sind die drei Heatmaps zu sehen, rechts die Heatmap des Originalbildes, in der Mitte die Heatmap des um 10 Pixel nach rechts verschobenen Originalbildes und auf dem dritten Bild wurde die mittlere Heatmap wieder zurücktransformiert. Um die Heatmap des Originalbildes und des zurücktransformierten Bildes vergleichbar zu machen, wurde die im Transformationsprozess veränderten Ränder bei beiden Heatmaps herausgerechnet.



**Abbildung 5.4:** links: Heatmap des Originalbildes, Mitte: Heatmap des transformierten Bildes rechts: die zurücktransformierte Heatmap

Bemerkenswert ist hier der Vergleich zwischen der Heatmap des Originalbildes (linkes Bild) und der Heatmap des zurücktransformierten Bildes (rechtes Bild). Die berechnete euklidische Distanz zwischen diesen beiden Bildern beträgt: 0.008840068. Dieser Wert ist gering und stimmt mit dem visuellen Vergleich der beiden Abbildungen überein. Bei diesem Beispiel wurde folgende Parameter-Einstellung für LRP verwendet: *lrp-exponent* = 2, *beta* = 1, *epsilon* = 1e-6, *method*='e-rule'.

Bei der in dieser Untersuchung verwendeten Implementierung für LRP kann zwischen der  $\epsilon$  – rule und die  $\beta$  – rule gewählt werden. Zur  $\epsilon$  – rule gehört die beiden Parameter *lrp-exponent* und der *epsilon*-Wert, zur  $\beta$  – rule gehört nur der *beta*-Wert. Wenn die  $\epsilon$  – rule verwendet wird, haben nur der *lrp-exponent* und der *epsilon*-Wert einen Einfluss, eine Veränderung des *beta*-Wertes würde in diesem Fall keine Auswirkungen haben. Entsprechendes gilt für den *beta*-Wert, hier haben Veränderungen der *epsilon*-Parameter keine Auswirkungen. Mit diesem Wissen wurden die untersuchten Parameter in der Tabelle 5.2 gewählt.

Mit dem *LRP-Exponenten* können die Relevanzwerte der Knoten in jeder Schicht neu skaliert werden. Hier handelt es sich um den Wert  $p$ , der in der Formel 3.3 im Kapitel zur Erläuterung von Layerwise Relevance Propagation 3.2 erwähnt wird. Mit diesem Wert kann die Berechnung der Beiträge der Übergänge von einem Knoten auf einen Nachfolgeknoten beeinflusst werden.

Der  $\epsilon$ -Wert hat die Aufgabe die Relevanz-Werte zu reduzieren, wenn die Beiträge zur Aktivierung eines Neurons gering oder widersprüchlich sind. Eine Erhöhung des  $\epsilon$ -Wertes reduziert die Erklärungsfaktoren, die eine geringerer Bedeutung haben. Dies führt in der Regel zu Erklärungen, die in Bezug auf Eingabemerkmale sparsamer und weniger verrauscht sind. [27, S.199]

Bei einem höheren  $\epsilon$ -Wert werden nur Neuronen von einer hohen Relevanz in der Heatmap berücksichtigt. Neuronen von geringerer Relevanz werden nicht angezeigt. Dies führt dazu, dass die Heatmap karger und weniger verrauscht ist. Ist der  $\epsilon$ -Wert geringer, dann ist die Heatmap in der Regel stärker verrauscht.

Über den  $\beta$ -Wert kann das Gewicht von Faktoren, die gegen eine bestimmte Klassifizierung sprechen variiert werden. Man spricht hier von hemmenden Faktoren. Ist der  $\beta$ -Wert nahe Null dann werden nur die eine Klassifizierung unterstützenden Faktoren angezeigt. Durch einen  $\beta$ -Wert ungleich von Null kann der hemmende Effekt von Neuronen-Aktivierungen korrigiert werden. [5, S.4]

Eine Erhöhung des  $\beta$ -Wertes steigert die hemmende Wirkung und verringert das Ausmaß an positiver Evidenz. Dies führt dazu, dass nur die stärksten Regionen beibehalten werden. Der optimale Wert für  $\beta$  hängt vom Klassifikator ab. [3, S.4]

Um zu überprüfen, in wie weit die ermittelte Distanz der beiden Heatmaps von den gewählten Parameter-Einstellungen von LRP abhängen, wurde das Distanzmaß für verschiedene LRP-Parameter-Einstellung ermittelt. Die Berechnung der Distanz erfolgte hier über die euklidische Norm. Es wurde für alle Bilder aus dem Trainingsdatensatz die jeweilige Distanz zwischen den beiden Heatmaps berechnet und daraus der Mittelwert erzeugt. Dabei wurde das Bild jeweils um 10 Pixel nach rechts verschoben. Da das Bild nur eine Größe von 32 x 32 Pixeln hat, war dies eine Verschiebung von ca. 31% der Kantenlänge. Bilder, die der Algorithmus falsch klassifiziert hat, wurden nicht in die Berechnung einbezogen. Die folgende Tabelle 5.1 gibt einen Überblick über den Einfluss, den verschiedene Parameter-Kombinationen von LRP auf die mittlere Distanz haben.

lrp-exponent	beta	epsilon	method	distance
1	0,5	0,1	e-rule	0,1220211
1	1	0,01	e-rule	3,0030327
1	1	1e-6	e-rule	8456,1694
2	1	1e-6	e-rule	0,0029356
1	0	1e-6	b-rule	0,0039013
1	0,5	1e-6	b-rule	0,0043239
1	1	1e-6	b-rule	0,0056651

**Tabelle 5.1:** Die gewählten LRP-Parameter beeinflussen die berechnete mittlere Distanz zwischen den Heatmaps

In Tabelle 5.1 unterscheiden sich die Distanz-Werte von den ersten drei LRP-Parameter-Kombination von den übrigen Werten. Wenn man die Heatmaps betrachtet, die mit diesen Einstellungen vorgenommen wurden, sind sie wenig aussagekräftig. Bei den

übrigen Einstellungen ist die Distanz zwischen den beiden erzeugten Heatmaps gering. Man kann davon ausgehen, dass das LRP-Verfahren stabiler ist, wenn die mittlere Distanz gering ist.

Die Robustheit des Algorithmus ist anscheinend von der getroffenen Parameter-Wahl bei LRP abhängig. Wenn eine stabile Interpretation der Ergebnisse angestrebt wird, sollte die LRP-Einstellungen so vorgenommen werden, dass die mittlere Distanz zwischen der jeweiligen Original-Heatmap und der zurück-transformierten Heatmap gering ist. In der obigen Tabelle 5.1 ist dies bei den unteren vier Einstellungen der Fall.

Die Robustheit des LRP-Algorithmus gegenüber einer horizontalen Translation des Ausgangsbildes ist gegeben, wenn die passende LRP-Parameter-Einstellung gewählt wurde. Ist dies der Fall, dann sind die Unterschiede zwischen der Heatmap des Originalbildes und der Heatmap des transformierten Bildes gering. Die Pixel, die beim Ausgangsbild relevant für die Klassifizierung sind, sind es auch bei dem transformierten Bild und auch die Höhe der Relevanz ändert sich nicht oder nur geringfügig, wenn die passenden LRP-Parameter verwendet werden.

### 5.3 Qualität der Erklärung durch LRP

#### Hypothese 3

Die Wahl unterschiedlicher LRP-Parameter wirkt sich auf die Heatmap aus. Es gibt LRP-Parameter-Werte, die für eine Anwendung geeigneter ist als andere Einstellungen. In dieser Untersuchung soll die dem gewählten Netz und verwendeten Datensatz entsprechenden geeigneten LRP-Parameter gefunden werden.

Das Thema der objektiven Bewertung der Qualität von Heatmaps wird noch erforscht, zurzeit gibt es noch keine allgemein anerkannten Richtlinien, wie die Qualität ermittelt werden kann. Derzeit existieren verschiedene Ansätze und Methoden, über die die Qualität von Heatmaps ermittelt werden kann.

In dieser Arbeit wird die Ausgabe des Neuronalen Netzes und das Einbringen von Störungen verwendet, um die Qualität zu bewerten. Ein entscheidender Einflussfaktor für die Qualität der Heatmap ist der zur Berechnung verwendete Algorithmus. Aber auch die Performanz des Neuronalen Netzes ist in diesem Zusammenhang ein wichtiger Aspekt. Die Effizienz der Klassifizierung durch das Neuronale Netz ist abhängig von der verwendeten Architektur des Netzes und dem Umfang und der Qualität der Trainingsdaten. [25, S.5]

Es sollte auch berücksichtigt werden, dass die Heatmap von der Struktur des klassifizierenden Algorithmus geprägt ist. Daher ist eine Übereinstimmung mit der mensch-

lichen Intuition oder eine Fokussierung auf das gewünschte Objekt nicht zwangsläufig gegeben. Die Autoren von [25, S.6] bevorzugen Heatmaps, die eine geringe Komplexität aufweisen, da diese für Menschen leichter zu verstehen sind. Die Heatmap sollte sich auf wenige Informationen konzentrieren und nur die relevanten Bereiche des Bildes hervorheben.

Die Komplexität der Heatmap kann als Hilfsmittel zur Beurteilung der Qualität herangezogen werden. Sie kann über die Entropie des Heatmap-Bildes ermittelt werden. Die Entropie eines Bildes ist ein theoretisches Maß für den Informationsgehalt eines Bildes. Gute Heatmaps fokussieren ihre Darstellung auf die relevanten Bereiche und vernachlässigen die unwichtigen. Demgegenüber zeigen weniger gute Heatmaps viele Informationen und Rauschen. Die Menge an Informationen in einem Bild wirkt sich auch auf die Dateigröße des komprimierten Bildes aus. Bilder mit vielen Informationen benötigen mehr Speicherplatz, als Bilder, die weniger Informationsgehalt haben. Insofern können über den Vergleich von Dateigröße von komprimierten Heatmaps, die mit verschiedenen Methoden erstellt wurden, Rückschlüsse bezüglich der Qualität der Heatmap gezogen werden. [25, S.8]

In dieser Arbeit wird die Qualität der Heatmap über das Einbringen von Störungen ermittelt. Bei der sogenannten Störungs-Analyse (Perturbation Analysis) werden einzelne Pixel gestört. Diese Methode beruht auf der Idee, dass die Störung von Pixeln, die für die Vorhersage des Netzes eine hohe Relevanz haben, zu einem stärkeren Rückgang des Vorhersagewertes für die richtige Klasse führen, als die Störung von Pixeln, die eine geringere Bedeutung haben. Auf diese Weise kann über den durchschnittlichen Rückgang des Vorhersagewertes, nachdem in mehreren Durchgängen Störungen der relevanten Pixel vorgenommen wurden, das Maß der Qualität der Heatmap ermittelt werden. Werden die relevantesten Pixel von der Methode, die zur Erklärung eingesetzt werden, identifiziert und anschließend gestört, dann sollte ein starker Rückgang beim Vorhersagewert beobachtet werden können. [27, S.15]

In dem Buch von Wojciech Samek [27, S.15] wird auf weitere Methoden hingewiesen, die zur Beurteilung der Qualität von Heatmaps eingesetzt werden können, wie zum Beispiel die „pointing game“-Methode, der Einsatz von aufgabenspezifischen Bewertungsschemata, die Nutzung von Ground Truth-Informationen oder die Erklärung auf der Basis von der Erfüllung von bestimmten Axiomen. Diese beiden Methoden sind für den Kontext der Untersuchung der Klassifizierung von Verkehrsschildern nicht oder nur bedingt geeignet. Die Ground Truth-Methode beruht darauf, dass es eine Grundwahrheit gibt, in Abschnitt 5.4 wird eine Untersuchung geschildert, die als Annäherung zu dieser Methode betrachtet werden kann. Die Erklärungen über Axiome sind eher bei der Untersuchung von Texten anwendbar.

In dieser Untersuchung wurde die störungs-basierte Analyse eingesetzt, um die Qualität der durch das LRP-Verfahren erzeugten Heatmap zu beurteilen. Die Idee ist, aus der mittels LRP erzeugten Heatmap die relevanten Pixel zu ermitteln. Die Position dieser Pixel wird gespeichert. Anschließend werden diese relevanten Pixel abhängig von der Höhe ihrer Relevanz absteigend sortiert. Im weiteren Verlauf werden schrittweise immer mehr relevante Pixel „gestört“, so dass sie für zur richtigen

Klassifizierung nicht mehr beitragen können. Unmittelbar nach dem Einbringen jeder zusätzlichen Störung wird ermittelt, welchen Einfluss dieses Vorgehen für die Klassifizierungsentscheidung hat. Zum Schluss, nachdem eine gewisse Anzahl von relevanten Pixeln gestört worden sind, wird die Höhe der Abnahme der Vorhersagefunktion berechnet.

### **5.3.1 Untersuchung mit dem Inception Netz**

Zur Umsetzung der Untersuchung wurde eine Python-Datei erstellt, mit der die Heatmap für ein Bild bestimmt werden kann. Aus der Heatmap werden die relevantesten Pixel ermittelt und die Koordinaten der Pixel gespeichert. Die Pixel werden nach Relevanzwert absteigend sortiert. In die relevanten Pixel werden Störungen eingebracht. Dabei wird schrittweise von den relevantesten Pixeln bis hin zu den weniger relevanten Pixeln vorgegangen. Es wird die Anzahl der eingebrachten Störungen ermittelt, ab der die Vorhersage der richtigen Klasse nicht mehr gegeben ist. Für jedes Bild wird das Verhältnis von Anzahl an Pixeländerungen zur korrekt vorhergesagten Klasse berechnet. Über diese errechneten Werte wird ein Mittelwert für die Gesamtzahl aller Bilder aus dem Trainingsdatensatz gebildet. In der Untersuchung wurden die 100 relevantesten Pixel schrittweise gestört.

Es existieren verschiedene Möglichkeiten einzelne Pixel zu stören. Die Anforderungen an ein optimales Verfahren sind, dass relevante Bildinformationen entfernen werden, ohne unbeabsichtigte neue Strukturen einzuführen. [25, S.12]

Um die Störung der relevanten Pixel der Heatmap vorzunehmen, wurden drei verschiedenen Verfahren untersucht:

**Listing 5.1:** Drei verschiedene Verfahren des Pixel-Flippings

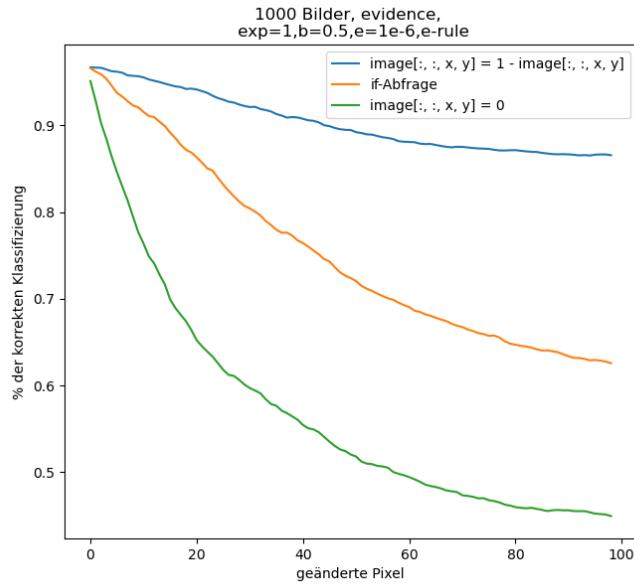
```
(1)     image[:, :, x, y] = 0
(2)     image[:, :, x, y] = 1 - image[:, :, x, y]
(3)     if float(image[:, 0, x, y]) <= 0.5:
            image[:, 0, x, y] = 1
        else:
            image[:, 0, x, y] = 0
        if float(image[:, 1, x, y]) <= 0.5:
            image[:, 1, x, y] = 1
        else:
            image[:, 1, x, y] = 0
        if float(image[:, 2, x, y]) <= 0.5:
            image[:, 2, x, y] = 1
        else:
            image[:, 2, x, y] = 0
```

Im ersten Pixel-Flipping-Verfahren wurde der Wert des Pixels an der zu manipulierender Stelle auf null gesetzt, im zweiten Verfahren wurde der Wert an dieser Stelle von Eins subtrahiert und im dritten Verfahren wurde untersucht, ob der Wert kleiner oder gleich 0,5 ist, in diesem Fall wurde er auf 1 gesetzt, ansonsten wurde er auf null gesetzt. In allen drei Methoden ist die Idee, die Pixel an den Stellen von hoher Relevanz einen anderen Wert zu geben, so dass sie für die Vorhersage nicht mehr relevant sind. Die Pixel haben einen Wert zwischen null und eins. In der *if-Abfrage* werden sie genau in die entgegengesetzte Richtung ihres bisherigen Wertes gedreht. Wenn sie kleiner oder gleich 0,5 sind erhalten sie den Wert 1 und wenn sie größer 0,5 sind erhalten sie den Wert 0. Es gibt drei *if-Abfragen*, weil dieses Flipping in allen drei Ebenen des dreidimensionalen Arrays eines Bildes erfolgt. Die Auswirkungen dieser drei Methoden kann der Abbildung 5.5 entnommen werden.

Das erste Verfahren des Pixel-Flippings, indem die einzelnen Pixel auf null gesetzt werden, erzeugt die besten Ergebnisse. Wie der Abbildung 5.5 zu entnehmen ist, fällt die Kurve dieses Verfahrens steil ab, je mehr relevante Pixel gestört werden. Die Kurve flacht ab, wenn schon der Einfluss eines großen Teils der relevanten Pixel gestört wurden. Die weniger relevanten Pixel haben nur noch einen geringeren Einfluss auf die Vorhersage.

Der Kurvenverlauf der beiden anderen Kurven ist nahezu linear. Hier ist die Steigung weniger abhängig von der Relevanz der Pixel. Deshalb sind diese beiden Verfahren weniger geeignet als das erste Verfahren. Im weiteren Untersuchungsverlauf wurde daher mit dem ersten Verfahren weitergearbeitet.

In einem zweiten Schritt wurde die geschilderte Versuchsanordnung variiert. Es wurde nicht mehr untersucht, ob die Klasse nach den schrittweise gesteigerten eingebrachten Störungen noch richtig vorhergesagt wird oder nicht. Hier wurde stattdessen die Wahrscheinlichkeit, dafür, dass ein Bild der korrekten Klasse zugeordnet wird, betrachtet. In jeder Runde wurde berechnet wie hoch die Wahrscheinlichkeit



**Abbildung 5.5:** Drei Arten des Pixel-Flippings.

in Bezug auf die korrekte Klasse nach jeder injizierten Störung ist. Aus diesen Wahrscheinlichkeiten wird am Ende des Programms der Durchschnittswert berechnet.

Die beiden geschilderten Versuchsanordnungen wurden jeweils für verschiedene LRP-Parameter-Einstellungen durchgeführt, um die Auswirkungen der Parameter-Wahl zu analysieren. Der Tabelle 5.2 sind die untersuchten LRP-Einstellungen zu entnehmen.

Es wurden vier verschiedene LRP-Einstellungen für die *e-rule* untersucht und zwei unterschiedliche für die *b-rule*. Für die *e-rule* ist der LRP-Exponent und der epsilon-Wert relevant. Daher wurde für die Regel zunächst mit der Grundeinstellung der LRP-Implementierung die Untersuchungen durchgeführt, dann die Auswirkungen zweier höherer Epsilon-Werte ( $\epsilon = 0.01, 0.1$ ) betrachtet. Für die letzte Untersuchung mit dieser Regel, wurde der Epsilon-Wert auf den ursprünglichen Wert ( $\epsilon = 1e - 6$ ) gesetzt und der LRP-Exponent auf 2.

Auf die *b-rule* hat nur der *beta*-Wert Einfluss, die anderen Parameter wurden auf die Grundeinstellung gesetzt. In Bezug auf den *beta*-Wert wurden die Untersuchungen mit der Grundeinstellung ( $\beta = 1$ ) und einem reduzierten Wert ( $\beta = 0.5$ ) durchgeführt.

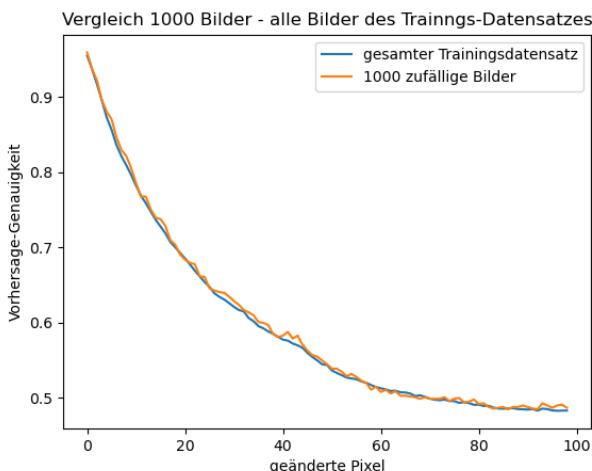
Zur Durchführung dieser Untersuchung wurden für die geschilderten LRP-Parameter-Kombinationen aus Tabelle 5.2 jeweils ein Durchgang mit jeweils 1000 Bildern aus dem Testdatensatz durchgeführt. Ein Durchgang mit der  $\epsilon - rule$  mit 1000 Bildern dauerte ca. 1 Stunden. Da die Implementierung der *beta - rule* weniger performant ist, hat der Rechner unter der Anwendung dieser Regel etwa 4 1/2-mal solange gebraucht, wie bei der  $\epsilon - rule$ . Deshalb wurde die Berechnungen nicht mit dem

	lrp-exponent	beta	epsilon	method
1.	1	1	1e-6	e-rule
2.	1	1	0,01	e-rule
3.	1	1	0,1	e-rule
4.	2	1	1e-6	e-rule
5.	1	0,5	1e-6	b-rule
6.	1	1	1e-6	b-rule

**Tabelle 5.2:** Die in dieser Untersuchung verwendeten LRP-Parameter

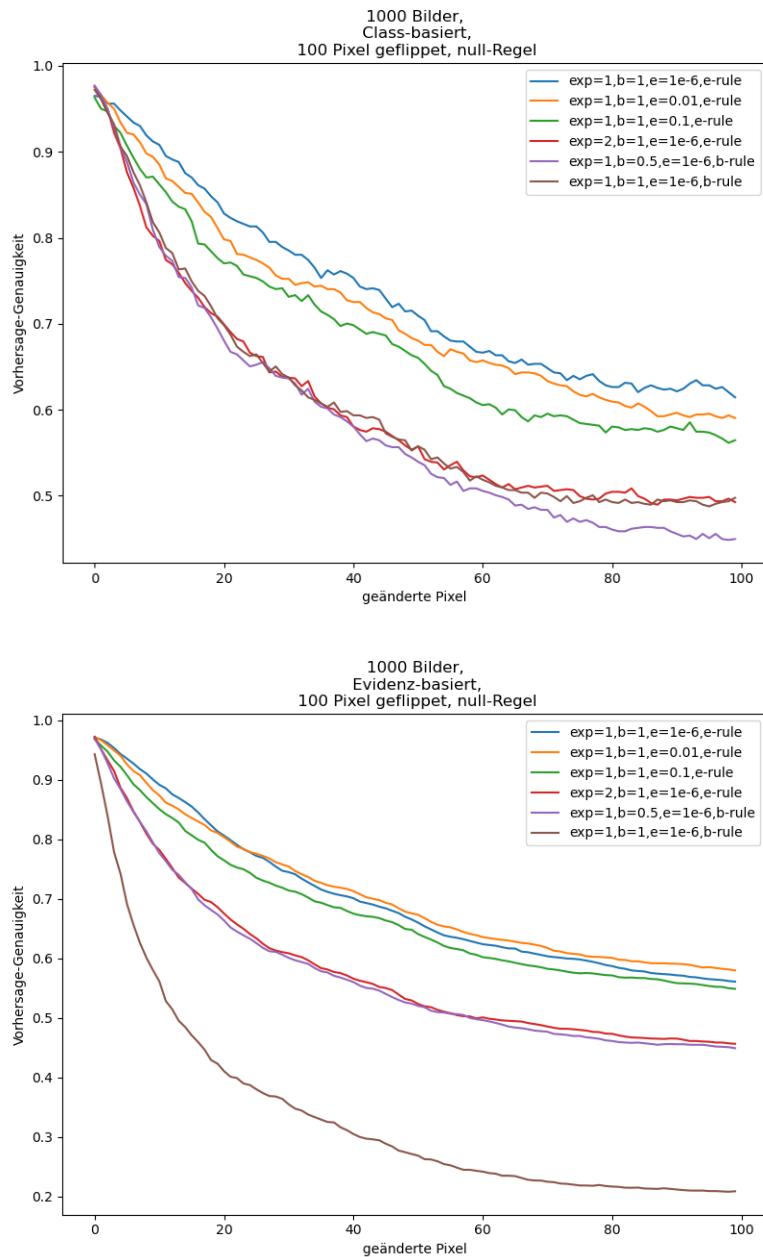
vollen Testdatensatz von über 12.000 Bildern durchgeführt. Die Verwendung von 1000 Bildern in dieser Untersuchung war ein guter Kompromiss zwischen der Repräsentativität der Ergebnisse und der Rechenzeit.

Abbildung 5.6 zeigt das die Unterschiede im Kurvenverlauf gering sind, wenn bei derselben Parameter-Einstellung, einmal die Berechnung auf der Basis von 1000 Bildern erfolgt und im anderen Fall der gesamte Trainingsdatensatz von über 12.000 Bildern verwendet wird. Der Kurvenverlauf bei 1000 Bildern ist nicht so glatt wie bei dem gesamten Datensatz, jedoch ist der grobe Kurvenverlauf nahezu identisch.



**Abbildung 5.6:** Auswirkung der Anzahl der verwendeten Bilder auf den Kurvenverlauf.

Die Untersuchung bezüglich der unterschiedlichen Parameter-Einstellungen wurde also zweimal ausgeführt, zum einen wurde analysiert, welchen Einfluss das Einbringen der Störung in die relevanten Pixel der Heatmap auf die Entscheidung hat, ob die richtige Klasse noch vorhergesagt wird oder nicht. Im zweiten Teil der Untersuchung wurde betrachtet, wie hoch die Wahrscheinlichkeit, dafür, dass ein Bild der korrekten Klasse zugeordnet wird, nach dem Einbringen der schrittweisen Störung ist. Beide Teile der Untersuchung wurden für die unterschiedlichen Parameter-Einstellung aus Tabelle 5.2 durchgeführt. Die Ergebnisse sind in den beiden Abbildungen in 5.7 dargestellt.



**Abbildung 5.7:** Auswirkungen verschiedener LRP-Parameter:  
 oben: klassen-basierte Entscheidung;  
 unten: wahrscheinlichkeits-basierte Entscheidung

Obwohl bei beiden Untersuchungen dieselbe Anzahl an Bildern verwendet wurde und auch die LRP-Parameter gleich waren, fällt auf, dass die unteren Kurven einen glatteren Verlauf haben. Dies liegt wahrscheinlich daran, dass die Kurven in der oberen Abbildung auf der Entscheidung richtige oder falsche Klasse, also auf diskreten Werten beruht und die Kurven des unteren Diagramms auf Wahrscheinlichkeiten, also stetigen Werten, ermittelt wurden.

In den beiden Diagrammen ist der beste Kurvenverlauf dort zu finden, wo die Kurve zu Beginn stark abfällt und mit zunehmenden gestörten Pixeln (dargestellt auf der x-Achse) abflacht. Je steiler die Kurve abfällt, desto genauer ist der Erklärung für die Entscheidung des Neuronalen Netzes. [28, S.254]

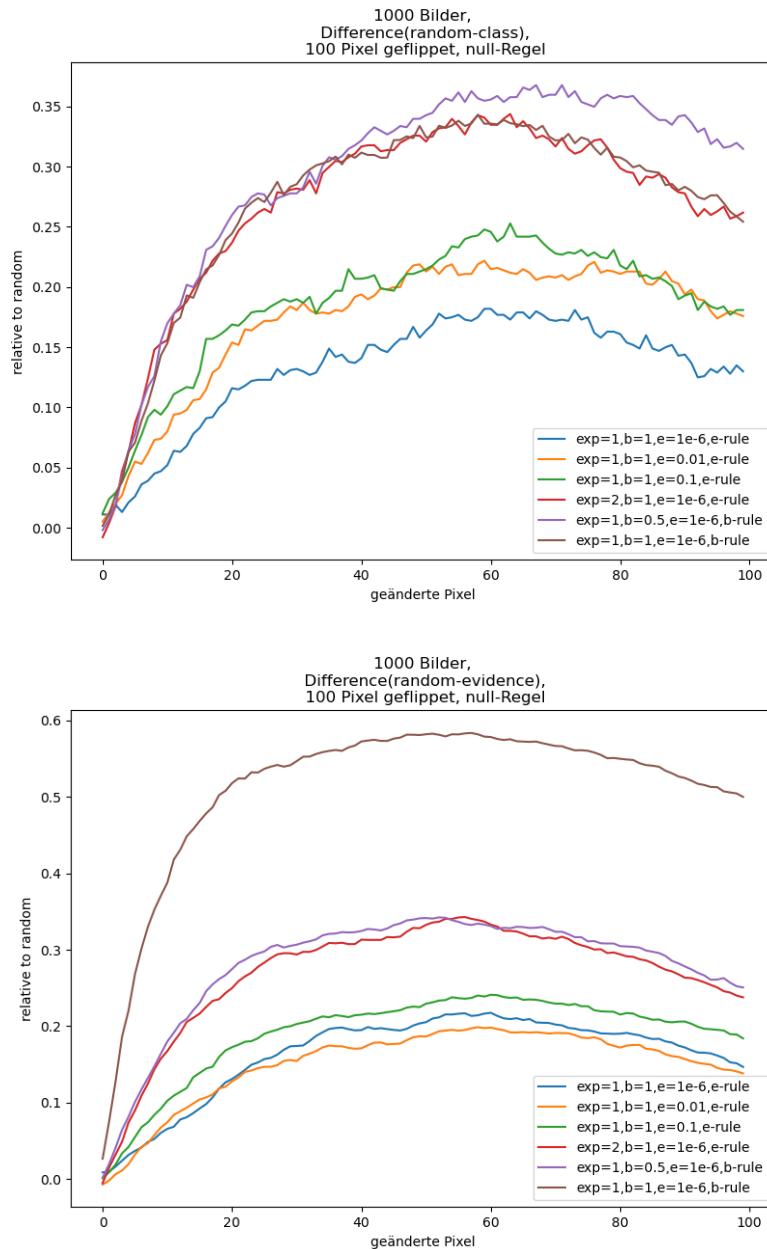
In beiden Diagrammen liefern die drei unteren LRP-Parameter-Einstellung die besten Ergebnisse. Im wahrscheinlichkeits-basierten unteren Diagramm zeigt die unterste Kurve den besten Verlauf. Die Kurve fällt durch das Auslöschen der relevantesten Pixel stark ab und geht in einen flacheren Verlauf über, nachdem die relevantesten Pixel entfernt wurden.

Für die beiden Untersuchungen, die auf klassen-basierte Entscheidungen und wahrscheinlichkeitsbasierte Entscheidungen beruhen, wurden der Bezug zu relativen Werten betrachtet, um die Ergebnisse zu normieren. Die Resultate sind den beiden Diagrammen in Abbildung 5.8 zu entnehmen.

In Abbildung 5.8 werden die Ergebnisse, die in Abbildung 5.7 dargestellt sind, in Bezug zu Zufallswerten gesetzt. Dieses Vorgehen wurde gewählt, um die Qualität der erzeugten Heatmap beurteilen zu können. Es ist zu erwarten, dass eine gute Heatmap besser Ergebnisse erzeugt, als eine Heatmap, die auf Zufallswerten beruht. Eine zufällige Heatmap kann als untere Schranke dafür betrachtet werden, was eine Heatmap leisten kann.

Zur Berechnung der Zufallswerte wurde dieselben Verfahren angewendet wie bei den Ausgangsverfahren. Einmal wurden Zufallswerte klassen-basiert erzeugt und für das andere Verfahren wahrscheinlichkeits-basiert. In beiden Verfahren wurden statt der schrittweisen Entfernung der relevantesten Werte, schrittweise zufällige Werte negiert. Die Kurven in Abbildung 5.8 wurden dann aus der jeweiligen Differenz der Zufallswerte und der klassen-basierten beziehungsweise der wahrscheinlichkeits-basierten Kurven berechnet. Ein steiler Anstieg der Kurven bei Erhöhung der Störungen ist ein Hinweis darauf, dass die relevanten Pixel der Heatmap einen höheren Einfluss auf die Entscheidung des Modells haben, als wenn der Anstieg weniger steil ist. [25, S.7] Bei beiden Untersuchungen liefern die Berechnungen über LRP mit der Einstellung *b-rule* die besseren Ergebnisse, gegenüber der Verwendung der *e-rule*.

Bei der Untersuchung der klassen-basierten Entscheidung liefern die LRP-Einstellungen, die auf der *b-rule* beruhen und die LRP-Einstellung, mit der *e-rule* und dem LRP-Exponenten 2 nahezu identische Ergebnisse. In der Untersuchung der klassen-basierten Entscheidung sind die Resultate unter Verwendung der *b-rule* und dem Beta-Wert von 1 am besten.

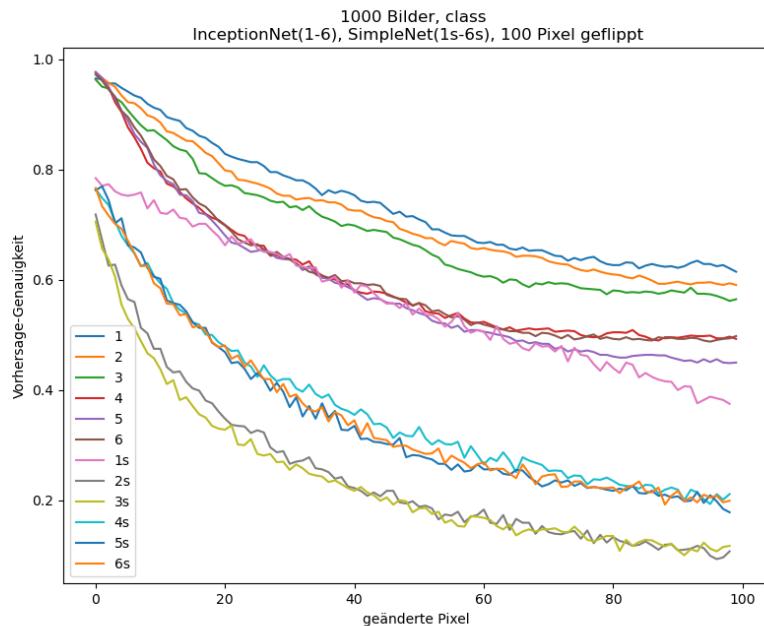


**Abbildung 5.8:** Verschiedene LRP-Parameter in Bezug zu Kurven aus Zufallswerten  
 oben: klassenbasierte Entscheidung;  
 unten: wahrscheinlichkeitsbasierte Entscheidung

### 5.3.2 Untersuchung mit dem einfachen Netz

Die bisherigen Untersuchungen wurde auf Grundlage des Inception Netzes, wie es in 4.2 beschrieben wurde, durchgeführt. Zur Überprüfung, welchen Einfluss das verwendete Netz auf die Ergebnisse hat, wurden die klassen-basierten Untersuchungen und die wahrscheinlichkeits-basierten Untersuchungen mit einem einfacheren Netz wiederholt. Die Struktur dieses Netzes wird in Abschnitt 4.2 erläutert.

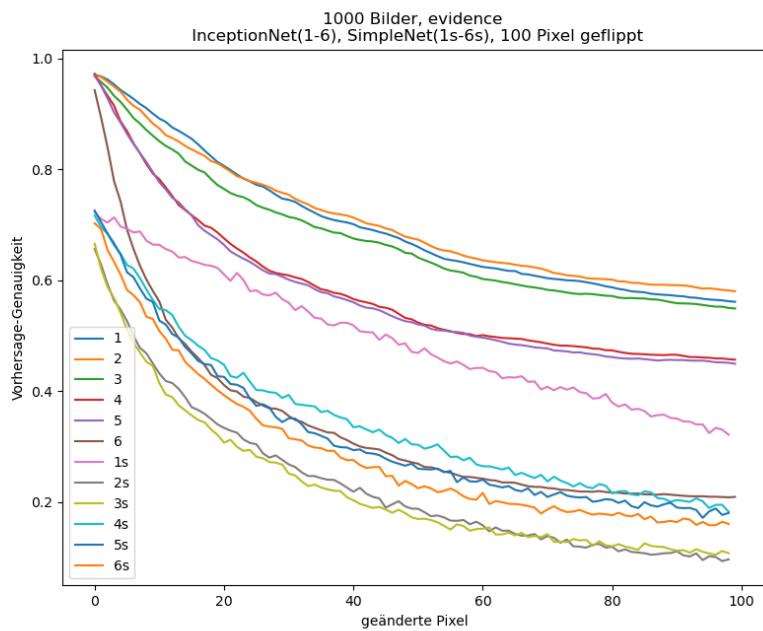
Die Ergebnisse des Inception Netzes und des einfachen Netzes werden in den beiden Diagrammen in Abbildung 5.9 und 5.10 dargestellt. Das erste Diagramm zeigt den Verlauf der klassen-basierten Untersuchung, das zweite Diagramm den Verlauf der wahrscheinlichkeits-basierten Untersuchung. Die Untersuchung für jedes der beiden Netze wurde mit den sechs in Tabelle 5.2 aufgeführten LRP-Parameter-Kombinationen durchgeführt. Daher sind in jedem der beiden Diagramme 12 Kurvenverläufe zu sehen.



**Abbildung 5.9:** Vergleich des Ergebnisses zweier Netze mit sechs verschiedenen LRP-Parameter Kombinationen, Klassen-basiert, Parameter-Einstellungen (1 - 6 und 1s - 6s) entsprechend Tabelle 5.2.

In den beiden Diagrammen 5.9 und 5.10 fällt zunächst auf, dass die Kurvenverläufe des einfachen Netzes (1s bis 6s) bei ca. 80% Genauigkeit beginnen und bei dem Inception Netz (1 bis 6) bei fast 100% Genauigkeit. Dies liegt daran, dass das Inception Netz nach dem Training eine Genauigkeit von 98,7% erreicht hat und das einfache Netz eine Genauigkeit von 85,2%.

Beim Einsatz des Inception Netzes führten die vierte, fünfte und sechste Parameter-Einstellungen aus der Tabelle 5.2 zu den besten Ergebnissen, weil der Kurvenver-



**Abbildung 5.10:** Vergleich des Ergebnisses zweier Netze mit sechs verschiedenen LRP-Parameter Kombinationen, wahrscheinlichkeits-basiert, Parameter-Einstellungen (1 - 6 und 1s - 6s) entsprechend Tabelle 5.2.

lauf nach Störung der relevantesten Pixel zunächst stark abfällt und bei den weniger relevanten Pixeln abflacht. Unter dem einfachen Netz liefern die Parameter-Kombinationen zwei und drei die besten Ergebnisse. Hier sind die Kurvenverläufe jedoch nahezu parallel. Bei der wahrscheinlichkeits-basierten Untersuchung unter Verwendung des Inception Netzes erzeugt die sechste LRP-Einstellung, das mit Abstand beste Ergebnis von allen Untersuchungen.

Diese Untersuchung hat gezeigt, dass die Qualität des Ergebnisses von LRP von den gewählten Parameter-Einstellungen abhängt. Diese müssen dem jeweiligen Anwendungsgebiet angepasst werden. In Abhängigkeit vom eingesetzten Neuronalen Netz können unterschiedliche LRP-Einstellungen notwendig sein. Einstellungen, die für ein Netz gute Ergebnisse liefern, sind nicht unbedingt bei einem anderen Netz auch die beste Wahl. Für die Untersuchung des GTSRB-Datensatzes mit Hilfe des hier verwendeten Inception-Netzes hat die sechste Parameter-Einstellung aus Tabelle 5.2 die besten Ergebnisse erzeugt. Für andere Untersuchungen mit anderen Datensätzen und anderen Netzen sollte das Ergebnis verschiedener LRP-Parameter getestet.

## 5.4 Manipulation über einen Aufkleber

### Hypothese 4

Wenn ein Angreifer das Bild eines Verkehrsschildes so manipuliert, dass das Netz nicht das eigentliche Verkehrsschild vorhersagt, sondern ein Verkehrs-schild das der Angreifer ausgewählt hat, dann kann diese Manipulation über die durch LRP erzeugte Heatmap erkannt werden.

Um den Wahrheitsgehalt dieser Hypothese zu untersuchen, wurde diese Untersuchung in Anlehnung an die wissenschaftliche Arbeit zum „Adversarial Patch“ [6] durchgeführt. Die Autoren schildern in ihrer Arbeit eine Methode über die ein Aufkleber (englisch Patch) erstellt werden kann, über den ein Bild manipuliert werden kann. Der Aufkleber wird vor der Verwendung so trainiert, dass er eine bestimmte Klasse vorhersagt. In der realen Welt kann der Aufkleber direkt auf das Objekt, zum Beispiel ein Verkehrsschild, geklebt werden. In dieser Untersuchung wird der Aufkleber in das Bild des Verkehrsschildes eingebracht.

Der Aufkleber zeigt ein Farbmuster und ist für einen gewöhnlichen Betrachter unverdächtig. Für einen möglichen Angreifer, der die Erkennung des Bildes manipulieren möchte, erfüllt der Patch den Zweck, dass ein Neuronales Netz sich bei der Klassifi-zierung des Bildes auf den Patch stützt und nicht auf das eigentliche Objekt (hier das Verkehrszeichen), das klassifiziert werden soll. [6, S.1 und 5]

In dieser Untersuchung geht es darum zu ermitteln, ob über eine Methode des Erklärens wie LRP, so eine Manipulation aufgedeckt werden kann. Es soll überprüft werden, ob in der durch die LRP erzeugten Heatmap erkennbar ist, dass die nicht korrekt vorhergesagte Klasse, durch den Patch verursacht wurde. Dies wäre dann der Fall, wenn in der Heatmap im Bereich des Patches, sich die höchsten Relevanz-Werte für die Vorhersage befinden würden. Vom methodischen Verständnis ist dieses Vorgehen eine Annäherung an die „Ground Truth“-Methode, die wie bereits in Abschnitt 5.3 erwähnt wurde, eine Möglichkeit ist, um die Qualität von Heatmaps zu beurteilen. Die Grundwahrheit ist in dieser Untersuchung die richtige Klasse des Verkehrsschildes.

Zur Durchführung der Untersuchung wurde ein Bild mit einem Neuronalen Netz darauf trainiert, ein bestimmtes Verkehrsschild vorherzusagen. Dieses Bild wurde als Patch in das ursprünglich zu klassifizierende Bild eines Verkehrsschildes eingebracht. In Abbildung 5.11 ist das Beispiel eines mit einem Patch versehenen Bildes zu sehen.

In der Untersuchung wurden alle Bilder des Trainingsdatensatzes mit so einem Patch versehen. Der Patch wurde auf die Klasse 2 der Verkehrsschilder (Geschwindigkeitsbegrenzung 50 Stundenkilometer) des Trainingsdatensatzes trainiert. Bilder, bei denen die Manipulation nicht funktionierte, weil das Neuronale Netz nicht die Klasse des Patches vorhergesagt hat, wurden in der Untersuchung nicht berücksichtigt.



**Abbildung 5.11:** Das Neuronale Netz klassifiziert den Patch und nicht das Verkehrszeichen.

Bei der Untersuchung wurde weniger auf die visuelle Beurteilung der Heatmap Wert gelegt, weil diese in einem gewissen Maß subjektiv ist. Stattdessen wurde die Relevanzwerte der Heatmap betrachtet.

Die Untersuchung wurde so gestaltet, dass für allen Bildern des Trainingsdatensatzes, bei denen die Manipulation erfolgreich war, ermittelt wurde, wie hoch der Prozentsatz der Relevanzwerte ist, die innerhalb des Patch-Bereichs liegen und wie viel Prozent außerhalb dieses Bereiches liegen. Diese Analyse wurde zum Vergleich sowohl für das mit einem Patch versehene Bild, als auch für das Originalbild (ohne Patch) vorgenommen. Für die Gesamtzahl aller Bilder des Trainingsdatensatzes wurde ein Mittelwert über die Relevanzwerte errechnet.

Erwartet wurde, wenn man die Gesamtzahl aller so manipulierten Bilder betrachtet, dass bei den Bildern mit Patch ein hoher mittlerer Prozentsatz des durch das Heatmappingverfahren bestimmten Relevanzbereichs innerhalb des Patches liegt, da der Patch darauf trainiert wurde diese spezielle Klasse vorherzusagen. Das Ergebnis war jedoch nicht so eindeutig, wie es erwartet wurde. Tabelle 5.3 zeigt die Ergebnisse im Einzelnen.

Bild mit Patch	
mittlere Relevanzwerte innerhalb des Patches:	53,59
mittlere Relevanzwerte außerhalb des Patches:	46,41
Originalbild (ohne Patch)	
mittlere Relevanzwerte innerhalb des Patch-Bereichs:	28,01
mittlere Relevanzwerte außerhalb des Patch-Bereichs:	71,99

**Tabelle 5.3:** Vergleich der mittleren Relevanzwerte innerhalb und außerhalb des Patches

Die mittleren Relevanzwerte innerhalb des Patches sind bei dem Bild mit Patch zwar wesentlich höher, als bei dem Bild ohne Patch, betrachtet man aber alleine das Bild mit Patch, dann liegen dort immer noch 46,41% der relevanten Pixel außerhalb des Patches. Dies ist erstaunlich, weil erwartet wurde, dass ein höherer Prozentsatz innerhalb des Patches liegt, weil die Klassifizierung hauptsächlich durch den Patch verursacht wurde.

Eine der möglichen Ursachen könnte in der Platzierung des Patches liegen. Da der Patch in der ersten Untersuchung zu diesem Thema nur leicht versetzt von der Mitte des Bildes platziert wurde und somit ein großer Teils des Verkehrsschildes vom Patch überdeckt war, wurde in einer zweiten Untersuchung der Patch mehr an den Rand des Bildes verlagert. (Abbildung 5.11 zeigt die Position des Patches in dieser zweiten Untersuchung) Erwartet wurde, dass durch die Verschiebung des Patches die mittleren Relevanzwerte, die innerhalb des Patches liegen höher sein werden.

Bild mit Patch	1. Untersuchung	2. Untersuchung
mittlere Relevanzwerte innerhalb des Patches:	53,59	56,11
mittlere Relevanzwerte außerhalb des Patches:	46,41	43,89
<b>Originalbild (ohne Patch)</b>		
mittlere Relevanzwerte innerhalb des Patches:	28,01	30,81
mittlere Relevanzwerte außerhalb des Patches:	71,99	69,19

**Tabelle 5.4:** Vergleich der mittleren Relevanzwerte (unterschiedliche Patch-Positionen)

Wie der Tabelle 5.4 zu entnehmen ist, haben sich die mittleren Relevanzwerte durch die Verschiebung des Patches an den äußeren Rand ein paar Prozent gegenüber der ersten Untersuchung erhöht. Bei dem Bild mit Patch sind sie im Bereich des Patches mit 56,11% immer noch geringer als erwartet.

Es fällt außerdem auf, dass in dieser zweiten Untersuchung auch die mittleren Relevanzwerte für den Bereich, den der Patch abdecken würde, im Originalbild um ca. 3% gestiegen sind. Wenn nicht der gesamte Testdatensatz für diese Untersuchung eingesetzt worden wäre, könnte vermutet werden, dass in der zweiten Untersuchung zufällig andere Verkehrsschild ausgewählt worden sind, bei denen eine höhere Relevanz in diesem Bereich vorhanden ist. Die Untersuchung wurde variiert, um eine Erklärung für die offenen Fragen zu bekommen.

Es bestand die Vermutung, dass die Auswahl der Verkehrsschildklasse, auf die der Patch trainiert wurde, einen Einfluss auf das Ergebnis der mittleren Relevanzwerte haben könnte, die innerhalb des Patches liegen.

Da die Untersuchungsergebnisse nicht ganz den Erwartungen entsprechen schien es sinnvoll zu verstehen, wie die Ergebnisse zustande gekommen sind. Die Aussagekraft des Mittelwertes liefert keine Informationen über die Verteilung der Werte über den ganzen Testdatensatz. Daher sollte nun untersucht werden, wie viel gepatchte Bilder jeweils mit wie viel Relevanz innerhalb des Patches verteilt sind. Also bei wie vielen Bildern 10% Relevanz innerhalb des Patches liegen, bei wie vielen 20% usw., bis zu den Bildern, bei denen 100% der Werte innerhalb des Patches liegen.

Weiterhin wurde das Untersuchungs-Design dahingehend verändert, dass in der Heatmap nur noch die zehn relevantesten Pixel betrachtet werden sollten. Um dies zu erreichen, wurden alle Pixel, mit Ausnahme den zehn relevantesten Pixel der Heatmap auf null gesetzt. Für die so geänderten Heatmaps wurde wiederum die

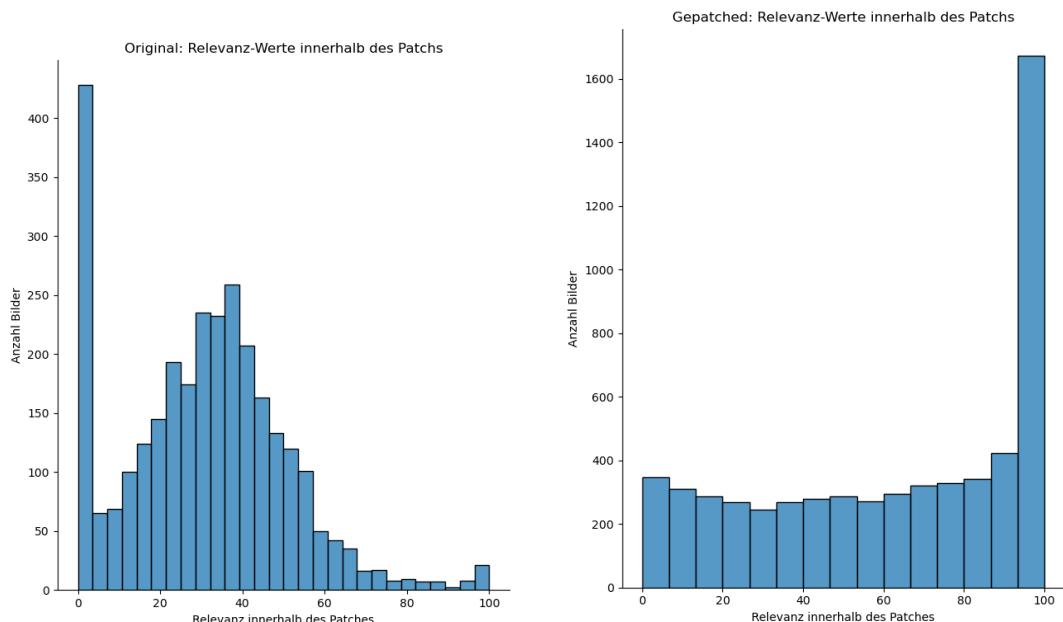
## 5. Untersuchungen zu LRP

---

Verteilung der Relevanzwerte, so wie im vorherigen Abschnitt geschildert, betrachtet.

Da die gewählten LRP-Parameter einen Einfluss auf die Heatmap haben, wurden die oben beschriebenen zwei Variationen der Heatmap (die vollständige Heatmap und die Heatmap mit den zehn relevantesten Pixeln) mit verschiedenen LRP-Parameter-Einstellungen durchgeführt. Zu diesem Zweck erfolgte der Untersuchungsdurchlauf für jeweils sechs verschiedene Parameter-Kombinationen. Für diese Berechnung wurden jeweils 6000 Bilder des Testdatensatzes verwendet. Aus Zeitgründen wurde nur ein Teil der Bilder aus dem Testdatensatz verwendet, da die Unterschiede der Ergebnisse zwischen vollem und halben Datensatz nur marginal waren.

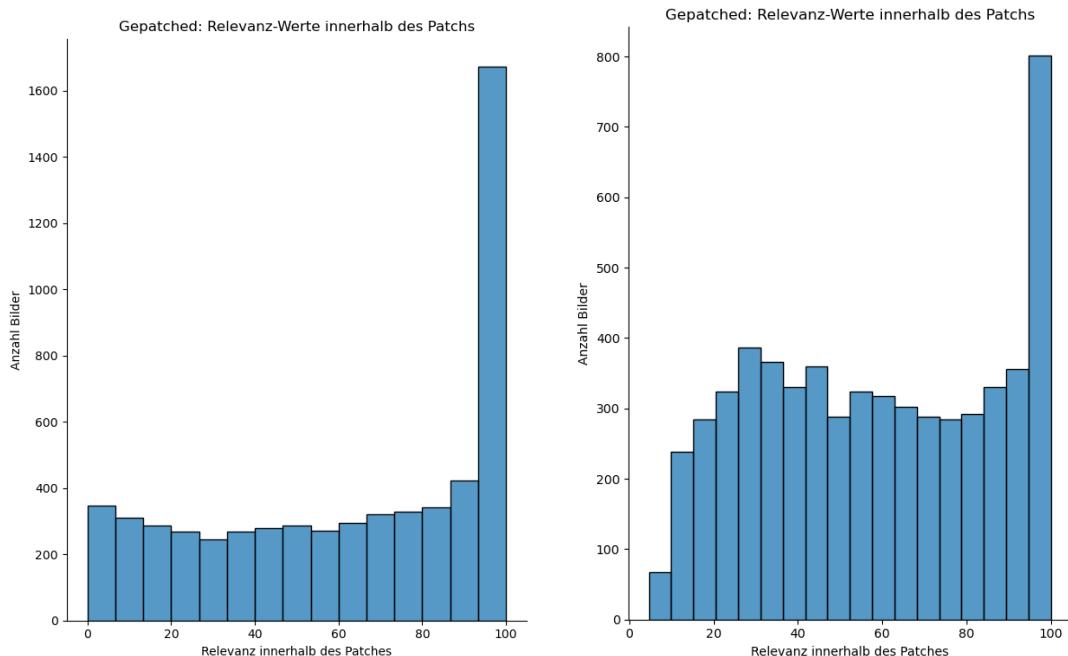
In Abbildung 5.12 wird die Verteilung des Prozentsatzes der Pixel, die sich im Bereich des Patches befinden, dargestellt. Links ist die Verteilung für das Originalbild zu sehen rechts für das Bild mit Patch. Da das Originalbild kein Patch hat, wurde, um eine Vergleichbarkeit herzustellen, hier der Bereich untersucht, in dem sich der Patch befinden würde, wenn er dieselbe Position einnehmen würde, wie es bei dem gepatchten Bild der Fall ist. Die Untersuchung, dieser Abbildung vorausgeht, wurde mit folgenden LRP-Parametern erstellt: LRP-Exponent = 1, beta = 1, epsilon = 0,1, e-rule, unter Verwendung der zehn relevantesten Pixel der Heatmap. Es wurden auch Plots für die sechs andere LRP-Einstellungen erzeugt, jeweils für die gesamte Heatmap und für die Heatmap, die nur die 10 relevantesten Pixel enthält. Diese Diagramme sind im Anhang 7.3 zu finden. Im Anhang 7.4 ist auch der beispielhafte Vergleich der Heatmaps für ein ausgewähltes Bild, jeweils mit Patch und ohne Patch, bei verschiedenen LRP-Parametern abgebildet.



**Abbildung 5.12:** Verteilung der Top 10 Relevanzwerte, die sich innerhalb des Patch-Bereiches befinden: links im Bild ohne Patch, rechts im Bild mit Patch

Auf der linken Seite der Abbildung 5.12 befindet sich die Darstellung der Untersuchung für das Originalbild ohne Patch. In den Heatmaps zu diesen Bildern haben über 400 Bilder keine oder sehr wenige Relevanz-Werte im Bereich des Patches. Der errechnete Mittelwert der Relevanz-Werte in diesem Bereich liegt bei 30,6%. Die meisten Werte liegen in dem Bereich zwischen 0 und 65%.

Anders sieht es bei dem Bild mit Patch aus. Hier hat das LRP-Heatmappingverfahren ergeben, dass bei ca. 27% der Bilder der Patch ausschließlich die Ursache für die Klassifizierung war. Man könnte erwarten, dass es einen kontinuierlichen Anstieg der Anzahl der Bilder mit den Relevanzwerten innerhalb des Patches in dem Bereich von 0% zu 100% gibt. Die Verteilung im Bereich zwischen 0% und 90% liegt nahezu gleich, bei jeweils ca. 370 Bildern. Wenn man sich die Verteilung als Kurve vorstellt, kann zwischen 90% und 100% der Relevanzwerte innerhalb des Patches, ein hoher Anstieg bei der Anzahl der Bilder festgestellt werden, bei denen diese Relevanzwerte innerhalb des Patches liegen.



**Abbildung 5.13:** Bild mit Patch: Relevanzwerte innerhalb des Patches: links Heatmaps mit den 10 relevantesten Pixeln, rechts die gesamten Heatmaps

In Abbildung 5.13 werden die Heatmaps, die die gesamten Relevanzwerte enthalten mit den Heatmaps verglichen, die nur die 10 relevantesten Pixeln enthalten, beide Diagramme beziehen sich auf die Heatmaps, die über für das Verkehrsschild mit Patch erzeugt wurden.

Auf der linken Seite der Abbildung 5.13 ist das rechte Bild von Abbildung 5.12 (Heatmap mit den 10 relevantesten Pixeln) erneut zu sehen. Diesmal im Vergleich mit der Verteilung der Relevanzwerte bei der gesamten Heatmap. Im rechten Bild liegen bei ca. 800 Bildern nur die Hälfte der Bilder der 100% Relevanz im Bereich des

Patches. Die Werte der restlichen Relevanzen, bei denen die Relevanzwerte zwischen 20% und 90% liegen, ist ähnlich wie in der linken Darstellung, gleichmäßig verteilt. Dies ist ein Hinweis darauf, dass die 10 relevantesten Pixel einen hohen Einfluss auf das Ergebnis haben. Alle Pixel mit weniger Relevanz, sind weniger bedeutsam für die Klassifizierung.

Tabelle 5.5 zeigt wie sich die LRP-Einstellungen auf die mittleren Relevanzwerte auswirken, die sich innerhalb und außerhalb des Patches befinden. Für diese Untersuchung von Bedeutung, sind die Werte, die sich im gepatchten Bild innerhalb des Patches befinden, weil sie Auskunft darüber geben, ob die Manipulation des Bildes mit dem Verkehrsschild durch LRP erkannt werden kann. Hier liefern die LRP-Versionen, die die *b-rule* einsetzen bessere Ergebnisse, als die, die die *e-rule* verwenden.

					gepatchtes Bild		Originalbild	
	LRP-Exp.	beta	epsilon	rule	im Patch	außerhalb	im Patch	außerhalb
1.	1	1	1e-6	e-rule	56,83	43,17	28,86	71,14
2.	1	1	0,01	e-rule	59,98	40,02	29,20	70,80
3.	1	1	0,1	e-rule	62,72	37,28	30,57	69,43
4.	2	1	1e-6	e-rule	57,31	42,69	30,91	69,09
5.	1	0,5	1e-6	b-rule	64,82	35,18	30,07	69,93
6.	1	1	1e-6	b-rule	71,5	28,5	31,53	68,47

**Tabelle 5.5:** Vergleich der mittleren Relevanz-Wert bei verschiedenen LRP-Einstellungen  
(Grundlage Heatmap mit den 10 relevantesten Pixeln)

Die Ergebnisse aus Tabelle 5.5 zeigen, dass die LRP-Einstellungen für das Originalbild (ohne Patch), kaum einen Einfluss auf den Anteil der Relevanzwerte haben, die sich innerhalb des Patches befinden. Bei dem Bild mit Patch, variieren die Relevanzwerte innerhalb des Patches je nach LRP-Einstellung zwischen ca. 57% und 71,5%. Hier können gut gewählte Parameter dazu beitragen, dass die Manipulation über einen Patch besser erkannt werden kann.

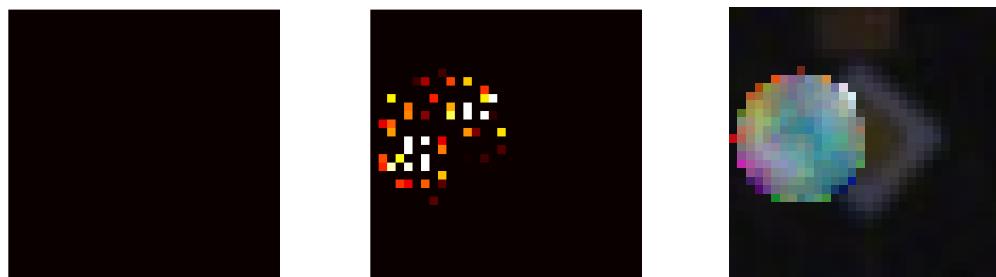
Was bedeutet dieses Ergebnis für die untersuchte Hypothese? Wenn man die Relevanzwerte der Heatmap betrachtet, kann man einen Unterschied zwischen den Bildern mit Patch und den Bildern ohne Patch feststellen. Mehr als ein Viertel der Bilder die manipuliert wurden, stützen sich bei ihrer Vorhersage vollständig auf den Patch (nach Abbildung 5.13, linkes Bild). Bei den Bildern ohne Patch sind es nur sehr wenige, die einen hohen Relevanzbereich im Patch haben. Daher ist der Einfluss des Patch für die Vorhersage in der Heatmap erkennbar. Jedoch ist das Ergebnis nicht so eindeutig, wie es zu erwarten gewesen wäre.

Es gibt verschiedene Aspekte, die das Ergebnis der Vorhersage beeinflussen könnten. Zum einen wurde der Patch auf das Verkehrsschild mit der Geschwindigkeitsbegrenzung 50 km/h trainiert. Im verwendeten Datensatz gibt es insgesamt acht Schilder, die eine Geschwindigkeitsbegrenzung zum Inhalt haben. Diese unterscheiden sich nur durch die Höhe der Geschwindigkeitsbeschränkung, erkennbar, durch die Zahl

in der Mitte des Verkehrszeichens. Die äußere Form dieser Bilder ist gleich. Darüber hinaus haben auch das „Durchfahrt verboten“-Schild und die beiden Überholverbot-Schilder (für PKW und LKW) eine ähnliche Form. Dies bedeutet, dass 11 von 43 Verkehrsschildern eine große Ähnlichkeit hinsichtlich Form und Farbe haben.

Der Patch verdeckt in der Regel nur einen Teil der Zahl, nicht aber die Form des Bildes. Bei einem Bild mit Patch könnte das ursprüngliche Verkehrsschild, was nicht durch den Patch verdeckt wird, auch zu der manipulierten Vorhersage beitragen, wenn nur die Form des Schildes betrachtet wird und nicht der Inhalt in der Mitte des Bildes. In diesem Fall könnte dem Netz die Klassifizierung als Geschwindigkeitsbegrenzungsschild leichter fallen, als wenn dort Schilder von anderer Form oder/und Farbe dargestellt werden. Bei einer ähnlichen Form würde das ursprüngliche Bild also die falsche, durch den Patch verursachte Vorhersage, stützen.

Zukünftige Untersuchungen sollten sich der Frage widmen, welchen Einfluss der Typ des Verkehrsschildes, auf den der Patch trainiert wurde, auf den Erfolg der Manipulation hat. Sind Schilder leichter zu manipulieren, die eine große Ähnlichkeit zu dem Schild haben, auf das der Patch trainiert wurde? Es könnte sein, dass der Patch bei einigen Klassen von Verkehrszeichen überhaupt nicht funktioniert und bei anderen Klassen überdurchschnittlich gut funktioniert. Dies würde einen Einfluss auf die Auswertung haben.



**Abbildung 5.14:** Vergleich Heatmap ohne Patch (links) und Heatmap mit Patch (Mitte), Originalbild mit Patch (rechts).

Es gibt Verkehrszeichen, die für das Neuronale Netz schwer zu erkennen sind, weil das Bild zum Beispiel sehr dunkel ist oder die Aufnahme des Verkehrszeichens undeutlich ist. Bei diesen Bildern besteht die zugehörige Heatmap nur aus einem schwarzen Bild. Wenn auf dieses Bild ein Patch aufgebracht wird, dann hat dieser Patch einen großen Einfluss auf die Heatmap, weil es außerhalb des Patches keine Werte von Relevanz gibt. Abbildung 5.14 zeigt ein entsprechendes Beispiel. Auf der linken Seite befindet sich die Heatmap des Originalbildes und in der Mitte die zugehörige Heatmap des Bildes mit dem Patch. Wenn der Datensatz viele Bilder enthält, bei denen die Heatmap schwarz ist, dann ist der Anteil an Bildern groß, bei dem der Patch 100% der Relevanzwerte enthält.

Ein anderer Einflussfaktor ist die Größe des Patches. Der verwendete Patch hat im Bild ungefähr die Größe des Verkehrsschildes. Die Größe des Patches konnte nicht

verringert werden, weil dadurch die Erfolgsschancen der Manipulation stark gemindert würden. Durch die Größe des Patches wird ein nicht unerheblicher Teil des eigentlichen Verkehrsschildes vom Patch überdeckt. Wenn wesentlich Teile des Verkehrszeichens abgedeckt werden, ist es für das Netz nicht möglich das eigentliche Verkehrsschild zu erkennen. Zum Beispiel wäre es schwierig bei dem Schild in Abbildung 5.11 zu sagen, ob das Ursprungsbild eine Geschwindigkeitsbegrenzung von 30 oder von 80 Stundenkilometer ist.

Die Heatmaps der Originalbilder unterscheiden sich von den Heatmaps der Bilder, auf die ein vor-trainierter Patch eingebracht wurde. In den Heatmaps zu den manipulierten Bildern liegen mehr Relevanzwerte innerhalb des Bereichs des Patches, als bei den nicht manipulierten Bildern. Ob die Position des Patches in der entsprechenden Heatmap klar identifiziert werden kann, ist auch von den gewählten LRP-Parametern abhängig. Das Erkennen der Patch-Position in der Heatmap ist nicht immer eindeutig und ist neben der Wahl der LRP-Parameter abhängig von der Qualität der Aufnahme des Verkehrsschildes. Dort wo die Aufnahmefähigkeit schlecht ist, wird auch das Verkehrsschild ohne Patch schwer vom Neuronalen Netz erkannt.

## 6 Zusammenfassung und Ausblick

In dieser Arbeit wurden die Arbeitsweise und die Leistungsfähigkeit von Methoden, die zur Erklärung von KI-Modellen eingesetzt werden können, untersucht. Konkret wurde das Anchors-Verfahren und die Layerwise Relevance Propagation betrachtet. Anchors wurde hauptsächlich aus der theoretischen Perspektive betrachtet, die Implementierung installiert und einige Durchläufe mit dem GSTRB-Datensatz ausgeführt. Aus zeitlichen Gründen konnte Anchors nicht detaillierter untersucht werden.

Das Anchors-Verfahren erzeugt Erklärungen dadurch, dass Bereiche des Ausgangsbilds hervorgehoben werden, die für die Klassifizierung ausschlaggebend sind. Die hervorgehobenen Bereiche sind ein zusammenhängender Teil des Bildes, der aus mehreren Pixeln, sogenannten Superpixeln, besteht. Eine so erzeugte Erklärung ist für den Menschen leicht verständlich, sie ist aber nur grob und zeigt keine Details.

Demgegenüber arbeitet Layerwise Relevance Propagation die Relevanz einzelner Pixel für die Klassifizierung heraus und stellt sie in einer Heatmap dar. Diese Erklärung ist genauer, als die durch Anchors erzeugte Erklärung. Der Rechenaufwand für Layerwise Relevance Propagation ist geringer als für Anchors, da bei LRP im Neuronale Netz für jedes Bild nur einmal ein Vorwärtsdurchlauf und ein Rückwärtsdurchlauf erfolgen muss. Anchors erfordert eine Vielzahl von Netzdurchgängen für jedes Bild, da zur Ermittlung der relevantesten Bildbereiche wiederholt Kombina-

tionen von einzelnen Bereichen ausgeblendet werden, um die optimale Klassifizierungswahrscheinlichkeit zu berechnen.

Da LRP aufgrund der detaillierten Erklärungen und der kürzeren Laufzeit die besseren Ergebnisse liefert, wurde die meisten Untersuchungen mit diesem Verfahren durchgeführt. Zur Bewertung des Verfahrens wurden vier Hypothesen aufgestellt und LRP diesbezüglich einer Untersuchung unterzogen.

Zunächst wurde die Stabilität des Verfahrens ermittelt. Dazu wurden einige Pixel der einzelnen Bilder gestört. Die Auswirkungen dieser eingebrachten Störung wurden über einen Vergleich der Heatmaps des gestörten und des nicht gestörten Bildes ermittelt. Dies geschah durch menschliche Beurteilung und rechnerischer Ermittlung der Distanz zwischen den beiden Heatmaps. Die in das Bild eingebrachten Störungen haben zu Veränderungen in der zur Erklärung erzeugten Heatmap geführt. Dies konnte sowohl visuell, als auch rechnerisch durch die Berechnung der Distanz festgestellt werden. Zwischen dem mathematischen Ergebnis und der visuellen Beurteilung konnte nur eine mäßige Korrelation festgestellt werden.

Im nächsten Schritt wurde untersucht, wie robust LRP gegenüber einer Verschiebung des Bildes ist. Im Experiment konnte gezeigt werden, dass bei richtig gewählten LRP-Parametern, durch die Transformation des Bildes die Auswahl der relevanten Pixel in der Heatmap nicht beeinträchtigt wird. Lediglich die Stärke der Relevanz einzelner Pixel hat sich durch die Transformation teilweise geringfügig geändert.

In einer weiteren Untersuchung wurden LRP-Parameter ermittelt, die für das konkrete Einsatzszenario der Klassifizierung von Bildern von Verkehrsschildern die besten Ergebnisse liefern. Dabei wurde zwei unterschiedliche Neuronale Netze eingesetzt, ein einfaches Netz und ein komplexeres Inception Netz. Es wurde festgestellt, dass für die beiden eingesetzten Netze jeweils unterschiedliche LRP-Parameter die besten Ergebnisse geliefert haben.

Thema der vierten Untersuchung war die Manipulation der Vorhersage über einen Patch im Bild. Der Patch wurde auf eine bestimmte Verkehrsschild-Klasse trainiert. Das Einbringen des Patches in das Bild hat die Vorhersage des Netzes dahingehend beeinflusst, dass durch den Patch die Auswahl der vorhergesagte Klasse forciert wurde. Durch die LRP-Heatmap konnte festgestellt werden, dass die relevanten Bereiche für die Vorhersage im Bereich des Patchs lagen. Die Ergebnisse waren jedoch nicht immer so eindeutig, wie es zu erwarten gewesen wäre. Wie gut die Klassifikationsentscheidung, die auf dem Patch basierte durch LRP erklärt werden konnte, wurde von den gewählten LRP-Parametern beeinflusst.

In diesem Zusammenhang wäre es lohnenswert zu untersuchen, welche Einflussfaktoren für die Erkennung der Manipulation durch den Patch noch von Bedeutung sind. So könnte zum Beispiel die Klasse des Verkehrsschildes, auf die der Patch trainiert wurde, einen Einfluss darauf haben, wie stark die durch LRP erzeugte Erklärung sich auf den Patch stützt. Auch die Ähnlichkeit zwischen ursprünglicher Klasse des Bildes und der Klasse, auf die der Patch trainiert wurde, könnte einen Einfluss dar-

## 6. Zusammenfassung und Ausblick

---

auf haben, wie groß der Anteil der Relevanzwerte ist, der nicht innerhalb des Patches liegt. Ein weiterer Punkt ist die Qualität des Ausgangsbildes. Ist das Bild unscharf, verwackelt oder sind Teile des Verkehrsschildes verdeckt (zum Beispiel durch einen Ast), dann könnte das nicht nur die Klassifizierung erschweren, sondern auch die Erklärung beeinflussen.

Obwohl in den letzten Jahren erhebliche Fortschritte hinsichtlich der Erklärbarkeit von KI gemacht wurden, stehen die Wissenschaftler weiter vor großen Herausforderungen. Sowohl hinsichtlich der theoretischen Grundlage, als auch der methodischen Umsetzung besteht weiterer Forschungsbedarf. Derzeit existiert noch keine allgemein anerkannte Theorie über erklärbare Künstliche Intelligenz. Es gibt keine Definitionen, die von den Forschern auf diesem Gebiet als allgemeingültig gesehen werden und kein Rahmenwerk, für die Beschreibung von neuen Methoden. Somit fehlt auch ein Maß zur Bewertung der Qualität von einzelnen Methoden. Eine Vergleichbarkeit von verschiedenen Methoden herzustellen ist daher erschwert. [27, S. 16f und S.240]

Die Erklärungen, die bisher bekannte Methoden erzeugen, sind meist von visueller Natur und bestehen oft darin, dass sie einzelne Pixel hervorheben und in einer Heatmap darstellen, die für die Klassifizierung relevant sind. Erklärungsmodelle können zwar die Relevanz von einzelnen Merkmalen für eine Vorhersage ermitteln, es bleibt jedoch oft unklar, ob ein einzelnes Merkmal alleine für einen Ausgabewert eines Modells verantwortlich ist oder ob eine Klassifizierungsentscheidung auf einer Kombination von verschiedenen Merkmalen beruht.

Derzeit haben die erzeugten Erklärungen noch einen niedrigen Abstraktionsgrad. Es werden einzelne Pixel oder Bereiche hervorgehoben, die maßgeblich zur Klassifizierung beigetragen haben. Die Interpretation dieser Bereiche obliegt aber noch dem Menschen. Erst durch die Interpretation ist der Mensch in der Lage, die Entscheidung des Modells zu verstehen. Der Interpretationsschritt kann schwierig und fehlerhaft sein. Um die menschliche Komponente im Prozess des Verstehens von Entscheidungen eines Neuronalen Netzes nicht zu vernachlässigen, sollten Experten aus dem Bereich Mensch-Computer-Interaktion in die weitere Forschung zu diesem Gebiet eingebunden werden. [27, S. 16f]

## 7 Literaturverzeichnis

- [1] Arras, Leila u. a. *Evaluating Recurrent Neural Network Explanations*. 2019. URL: <https://arxiv.org/abs/1904.11829v3> (besucht am 31.10.2019).
- [2] Bach, Sebastian u. a. „On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation“. In: *PLoS ONE* 10.7 (2015).
- [3] Binder, Alexander u. a. „Layer-Wise Relevance Propagation for Deep Neural Network Architectures“. In: *Information Science and Applications (ICISA) 2016*. Hrsg. von Kim, Kuinam J. und Joukov, Nikolai. Bd. 376. Lecture Notes in Electrical Engineering. Singapore: Springer Singapore, 2016, S. 913–922. ISBN: 978-981-10-0556-5. DOI: 10.1007/978-981-10-0557-2.
- [4] Binder, Alexander u. a. *Layer-wise Relevance Propagation for Neural Networks with Local Renormalization Layers*. URL: <http://arxiv.org/pdf/1604.00825v1>.
- [5] Böhle, Moritz u. a. „Layer-Wise Relevance Propagation for Explaining Deep Neural Network Decisions in MRI-Based Alzheimer’s Disease Classification“. In: *Frontiers in aging neuroscience* 11 (2019), S. 194. ISSN: 1663-4365. DOI: 10.3389/fnagi.2019.00194. URL: <https://www.frontiersin.org/articles/10.3389/fnagi.2019.00194/full> (besucht am 31.10.2019).
- [6] Brown, Tom B. u. a. *Adversarial Patch*. URL: <http://arxiv.org/pdf/1712.09665v2>.
- [7] Burkov, Andriy und Lorenzen, Knut. *Machine Learning kompakt: Alles, was Sie wissen müssen*. 1. Auflage. Frechen: MITP, 2019. ISBN: 9783958459953.
- [8] David Kriesel. *Ein kleiner Überblick über Neuronale Netze*. 2007. URL: [www.dkriesel.com/science/neural\\_networks](http://www.dkriesel.com/science/neural_networks).
- [9] Deza, Elena. *Dictionary of Distances*. Elsevier, 2006. ISBN: 9780444520876. DOI: 10.1016/B978-0-444-52087-6.X5000-8.
- [10] Fong, Ruth und Vedaldi, Andrea. „Explanations for Attributing Deep Neural Network Predictions“. In: *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. Hrsg. von Samek, Wojciech u. a. Cham: Springer International Publishing, 2019, S. 149–167. ISBN: 978-3-030-28953-9.
- [11] Frochte, Jörg. *Maschinelles Lernen: Grundlagen und Algorithmen in Python*. München: Hanser, 2018. ISBN: 978-3-446-45291-6. DOI: 10.3139/9783446457058.
- [12] Gill, Navdeep und Hall, Patrick. *Introduction to Machine Learning Interpretability*. Sebastopol, CA: O’Reilly Media, Inc, 2018. ISBN: 9781492033158.

- [13] Glander, Shirin. *Künstliche Intelligenz und Erklärbarkeit – Informatik Aktuell*. 2018. URL: <https://www.informatik-aktuell.de/betrieb/kuenstliche-intelligenz/kuenstliche-intelligenz-und-erklaerbarkeit.html> (besucht am 09.11.2019).
- [14] Khaleghi, Bahador. *The How of Explainable AI: Post-modelling Explainability*. 2019. URL: <https://towardsdatascience.com/the-how-of-explainable-ai-post-modelling-explainability-8b4cbc7adf5f> (besucht am 31.10.2019).
- [15] LeCun, Yann; Bengio, Yoshua und Hinton, Geoffrey. „Deep learning“. In: *Nature* 521.7553 (2015), S. 436–444. DOI: [10.1038/nature14539](https://doi.org/10.1038/nature14539).
- [16] Mittag, Hans-Joachim. *Statistik*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014. ISBN: 978-3-642-54386-9. DOI: [10.1007/978-3-642-54387-6](https://doi.org/10.1007/978-3-642-54387-6).
- [17] Molnar, Christoph. *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable*. Leanpub, 2019. URL: <https://christophm.github.io/interpretable-ml-book/> (besucht am 06.11.2019).
- [18] Montavon, Grégoire. „Gradient-Based Vs. Propagation-Based Explanations: An Axiomatic Comparison“. In: *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. Hrsg. von Samek, Wojciech u. a. Cham: Springer International Publishing, 2019, S. 253–265. ISBN: 978-3-030-28953-9.
- [19] Montavon, Grégoire; Samek, Wojciech und Müller, Klaus-Robert. „Methods for interpreting and understanding deep neural networks“. In: *Digital Signal Processing* 73 (2018), S. 1–15. ISSN: 10512004. DOI: [10.1016/j.dsp.2017.10.011](https://doi.org/10.1016/j.dsp.2017.10.011).
- [20] Montavon, Grégoire u. a. „Explaining NonLinear Classification Decisions with Deep Taylor Decomposition“. In: *Pattern Recognition* 65 (2017), S. 211–222. ISSN: 00313203. URL: <https://www.sciencedirect.com/science/article/pii/S0031320316303582?via%3Dihub> (besucht am 31.10.2019).
- [21] Ranjan, Anurag u. a. *Attacking Optical Flow*. URL: <http://arxiv.org/pdf/1910.10053v1.pdf>.
- [22] Ribeiro, Marco Tulio; Singh, Sameer und Carlos Guestrin. *Anchors: High Precision Model-Agnostic Explanations*. Hrsg. von University of Washington. 2018. URL: <https://homes.cs.washington.edu/~marcotcr/aaai18.pdf>.
- [23] Robnik-Šikonja, Marko und Bohanec, Marko. „Perturbation-Based Explanations of Prediction Models“. In: *Human and Machine Learning*. Hrsg. von Zhou, Jianlong und Chen, Fang. Cham: Springer International Publishing, 2018, S. 159–175. ISBN: 978-3-319-90402-3.
- [24] Samek, Wojciech und Müller, Klaus-Robert. „Towards Explainable Artificial Intelligence“. In: *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. Hrsg. von Samek, Wojciech u. a. Cham: Springer International Publishing, 2019, S. 5–22. ISBN: 978-3-030-28953-9.
- [25] Samek, Wojciech u. a. *Evaluating the visualization of what a Deep Neural Network has learned*. URL: <http://arxiv.org/pdf/1509.06321v1.pdf>.

- [26] Samek, Wojciech u. a. „Evaluating the Visualization of What a Deep Neural Network Has Learned“. In: *IEEE transactions on neural networks and learning systems* 28.11 (2017), S. 2660–2673. DOI: 10.1109/TNNLS.2016.2599820.
- [27] Samek, Wojciech u. a., Hrsg. *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. Bd. 11700. Cham: Springer International Publishing, 2019. ISBN: 978-3-030-28953-9. DOI: 10.1007/978-3-030-28954-6.
- [28] Samek, Wojciech u. a. „Explaining Deep Neural Networks and Beyond: A Review of Methods and Applications“. In: *Proceedings of the IEEE* 109.3 (2021), S. 247–278. ISSN: 0018-9219. DOI: 10.1109/JPROC.2021.3060483.
- [29] Shrikumar, Avanti; Greenside, Peyton und Kundaje, Anshul. *Learning Important Features Through Propagating Activation Differences*. 2017. URL: <https://arxiv.org/abs/1704.02685v1> (besucht am 31.10.2019).
- [30] Stallkamp, J. u. a. „Man vs. computer: benchmarking machine learning algorithms for traffic sign recognition“. In: *Neural networks : the official journal of the International Neural Network Society* 32 (2012), S. 323–332. DOI: 10.1016/j.neunet.2012.02.016.
- [31] Szegedy, Christian u. a. *Going Deeper with Convolutions*. URL: <http://arxiv.org/pdf/1409.4842v1>.
- [32] Yang, Yinchong u. a. „Explaining Therapy Predictions with Layer-Wise Relevance Propagation in Neural Networks“. In: *2018 IEEE International Conference on Healthcare Informatics (ICHI)*. IEEE, 62018, S. 152–162. ISBN: 978-1-5386-5377-7. DOI: 10.1109/ICHI.2018.00025.
- [33] Zilke, Jan Ruben; Mencia, Eneldo Loza und Janssen, Frederik. „DeepRED – Rule Extraction from Deep Neural Networks“. In: *Discovery Science*. Cham: Springer International Publishing, 2016, S. 457–473.

## 7.1 Aufbau des verwendeten Inception Netzes

```
InceptionNet3(
    (features): Sequential(
        (0): InceptionA(
            (conv1x1): BatchConv(
                (conv): Conv2d(3, 64, kernel_size=(1, 1), stride=(1, 1))
                (bn): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
                (relu): ReLU()
            )
            (conv5x5_1): BatchConv(
                (conv): Conv2d(3, 48, kernel_size=(1, 1), stride=(1, 1))
                (bn): BatchNorm2d(48, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
                (relu): ReLU()
            )
            (conv5x5_2): BatchConv(
                (conv): Conv2d(48, 64, kernel_size=(5, 5), stride=(1, 1), padding=(2, 2))
                (bn): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
                (relu): ReLU()
            )
            (conv3x3dbl_1): BatchConv(
                (conv): Conv2d(3, 64, kernel_size=(1, 1), stride=(1, 1))
                (bn): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
                (relu): ReLU()
            )
            (conv3x3dbl_2): BatchConv(
                (conv): Conv2d(64, 96, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
                (bn): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
                (relu): ReLU()
            )
            (conv3x3dbl_3): BatchConv(
                (conv): Conv2d(96, 96, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
                (bn): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
                (relu): ReLU()
            )
            (pool1x1): BatchConv(
                (conv): Conv2d(3, 32, kernel_size=(1, 1), stride=(1, 1))
                (bn): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
                (relu): ReLU()
            )
        )
        (1): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
        (2): BatchConv(
            (conv): Conv2d(256, 256, kernel_size=(2, 2), stride=(1, 1))
            (bn): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
            (relu): ReLU()
        )
        (3): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
        (4): BatchConv(
            (conv): Conv2d(256, 256, kernel_size=(2, 2), stride=(1, 1))
            (bn): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
            (relu): ReLU()
        )
        (5): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
        (6): BatchConv(
            (conv): Conv2d(256, 256, kernel_size=(2, 2), stride=(1, 1))
            (bn): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
            (relu): ReLU()
        )
        (7): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    )
    (classifiers): Sequential(
        (0): Linear(in_features=256, out_features=256, bias=True)
        (1): ReLU(inplace=True)
        (2): Dropout(p=0.5, inplace=False)
        (3): Linear(in_features=256, out_features=128, bias=True)
        (4): ReLU(inplace=True)
        (5): Dropout(p=0.5, inplace=False)
        (6): Linear(in_features=128, out_features=43, bias=True) ))
)
```

## 7.2 Aufbau des verwendeten einfachen Netzes

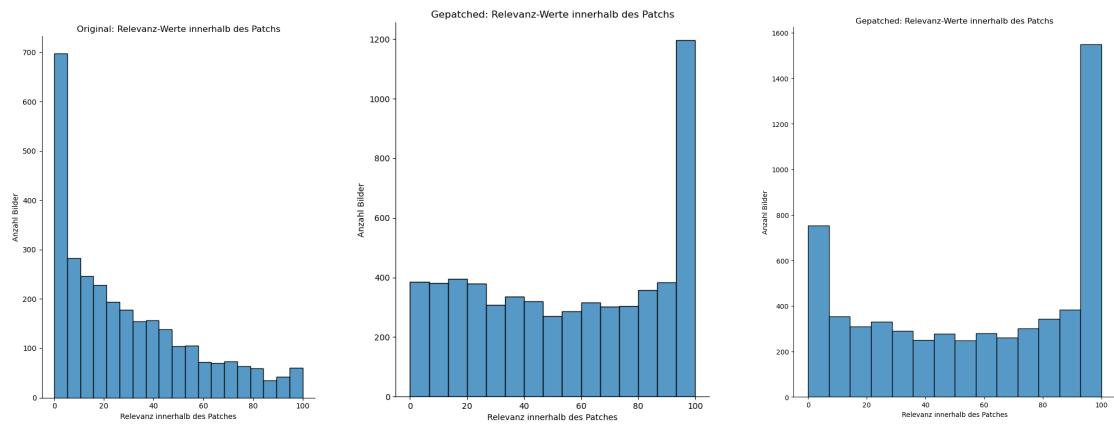
```
simple_Network2(  
    (conv1): Conv2d(3, 12, kernel_size=(5, 5), stride=(1, 1), padding=(2, 2))  
    (relu1): ReLU()  
    (pool1): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)  
    (conv2): Conv2d(12, 32, kernel_size=(5, 5), stride=(1, 1), padding=(2, 2))  
    (relu2): ReLU()  
    (pool2): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)  
    (conv3): Conv2d(32, 64, kernel_size=(5, 5), stride=(1, 1), padding=(2, 2))  
    (relu3): ReLU()  
    (pool3): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)  
    (conv4): Conv2d(64, 128, kernel_size=(5, 5), stride=(1, 1), padding=(2, 2))  
    (relu4): ReLU()  
    (pool4): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)  
    (fc1): Linear(in_features=512, out_features=256, bias=True)  
    (relu_fc1): ReLU()  
    (fc2): Linear(in_features=256, out_features=128, bias=True)  
    (relu_fc2): ReLU()  
    (fc3): Linear(in_features=128, out_features=43, bias=True)  
)
```

### 7.3 Verteilung der Relevanz-Werte bei den Heatmaps

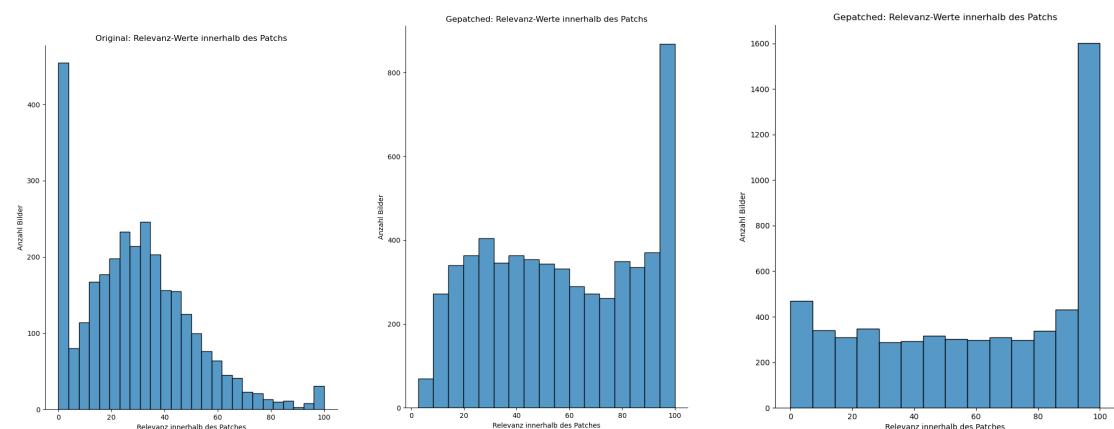
Vollständige Diagramme der Untersuchung, wie sie in Kapitel 5.4 geschildert wird. Nachfolgend werden die Diagramme der Verteilung der Relevanzwert der Heatmaps, die mit verschiedenen LRP-Parametern erstellt worden sind, dargestellt. Links ist immer die Heatmap des Originalbildes zu sehen, in der Mitte die Heatmap des Originalbildes mit Patch und rechts die 10 relevantesten Werte der Heatmap des Bildes mit Patch.

Originalbild - vollständige Heatmap - die 10 relevantesten Pixel der Heatmap

1. LRP-Parameter:  $\text{exp}=1$ ,  $b=1$ ,  $e=1e-6$ , e-rule



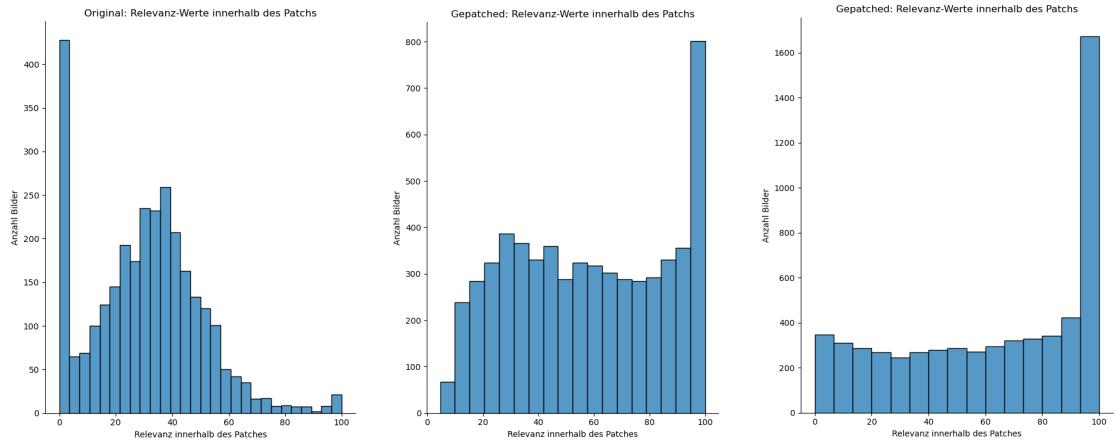
2. LRP-Parameter:  $\text{exp}=1$ ,  $b=1$ ,  $e=0.01$ , e-rule



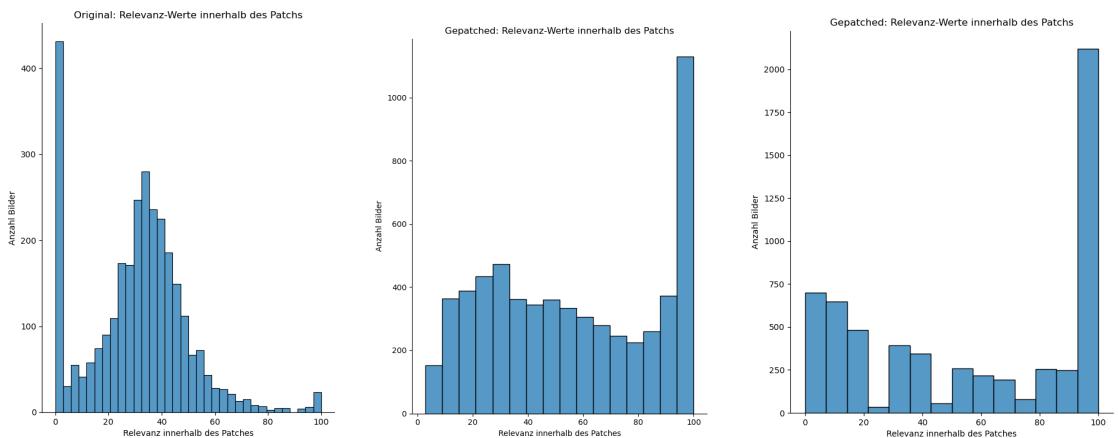
## 7. Anhang

---

### 3. LRP-Parameter: exp=1, b=1, e=0.1, e-rule



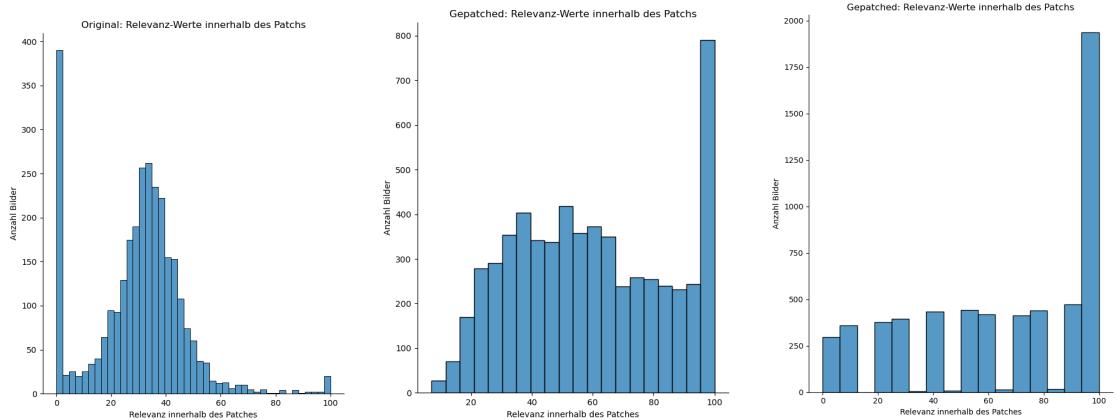
### 4. LRP-Parameter: exp=2, b=1, e=1e-6, e-rule



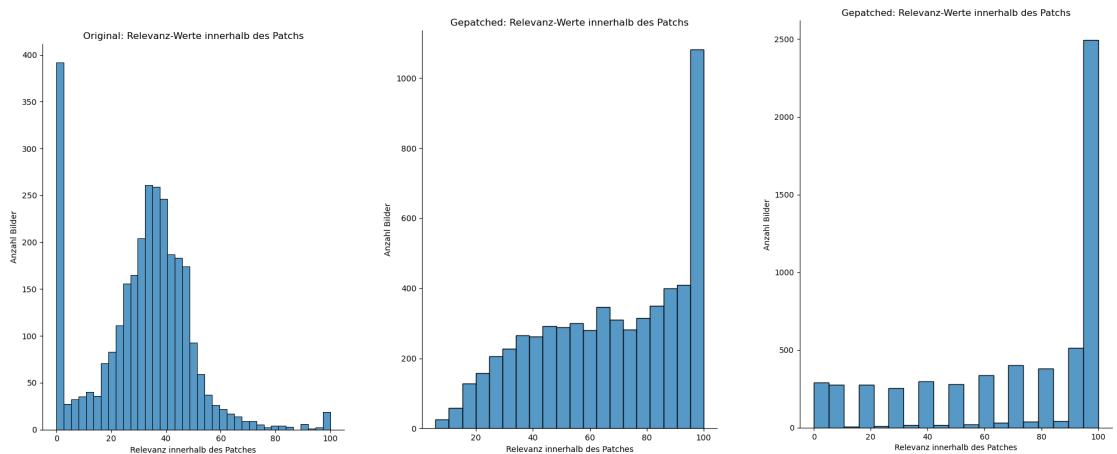
## 7. Anhang

---

### 5. LRP-Parameter: exp=2, b=0,5, e=1e-6, b-rule



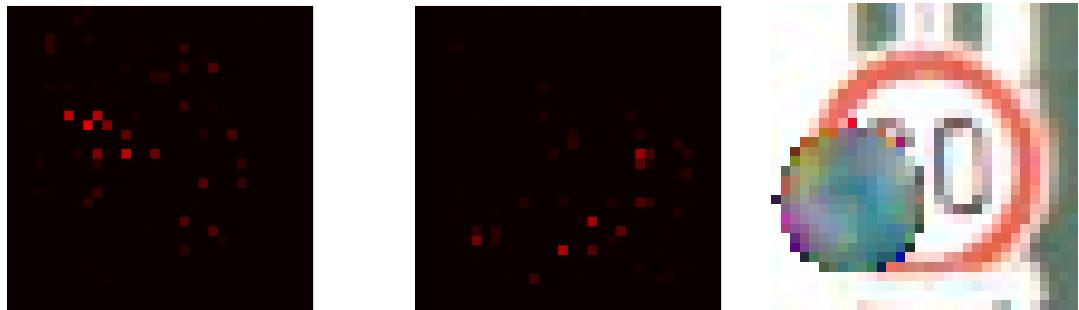
### 6. LRP-Parameter: exp=2, b=1, e=1e-6, b-rule



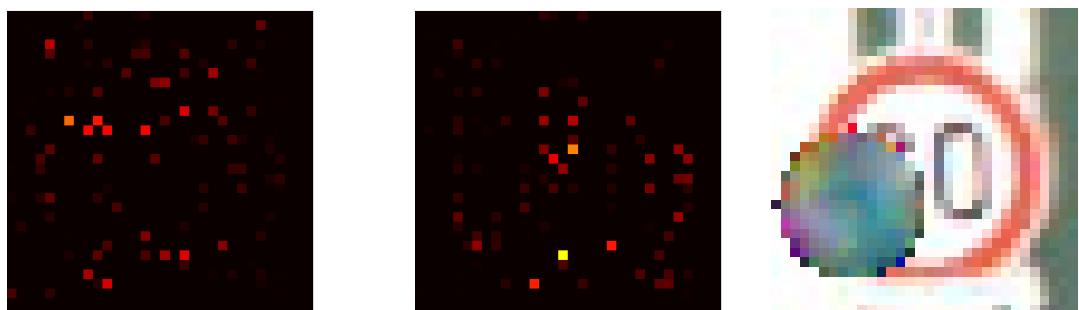
## 7.4 Heatmaps bei verschiedenen LRP-Parametern

Darstellung der Auswirkungen sechs verschiedener LRP-Parameter auf die Heatmap für ein Bild jeweils ohne Patch (links), mit Patch (Mitte) und Bild mit Patch (rechts).

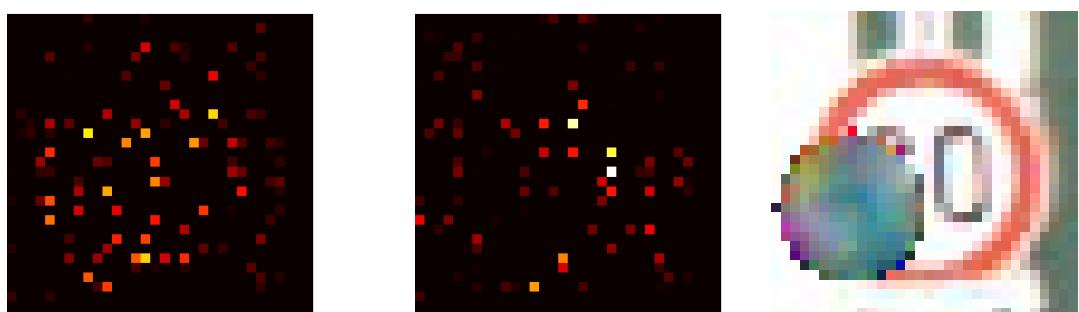
1. LRP-Parameter:  $\text{exp}=1$ ,  $b=1$ ,  $e=1\text{e-}6$ , e-rule



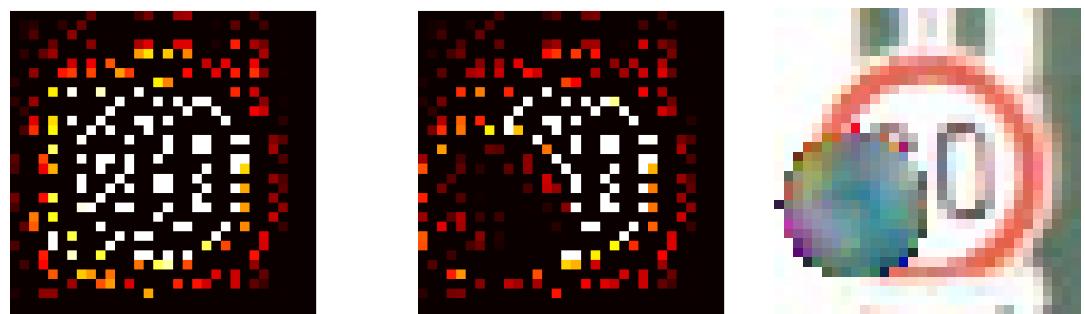
2. LRP-Parameter:  $\text{exp}=1$ ,  $b=1$ ,  $e=0.01$ , e-rule



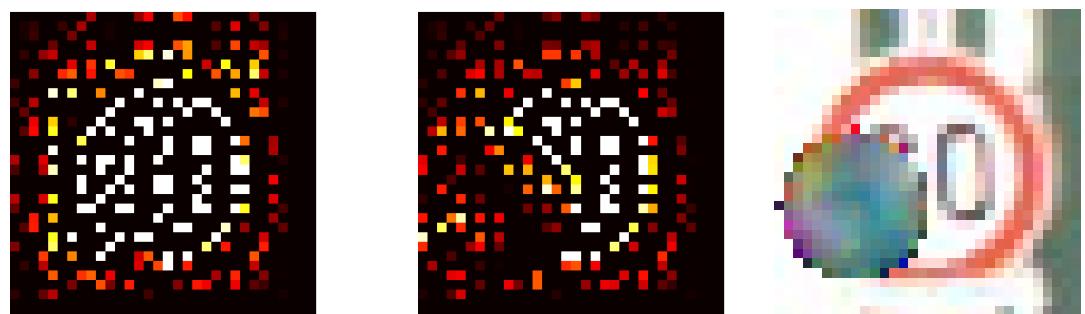
3. LRP-Parameter:  $\text{exp}=1$ ,  $b=1$ ,  $e=0.1$ , e-rule



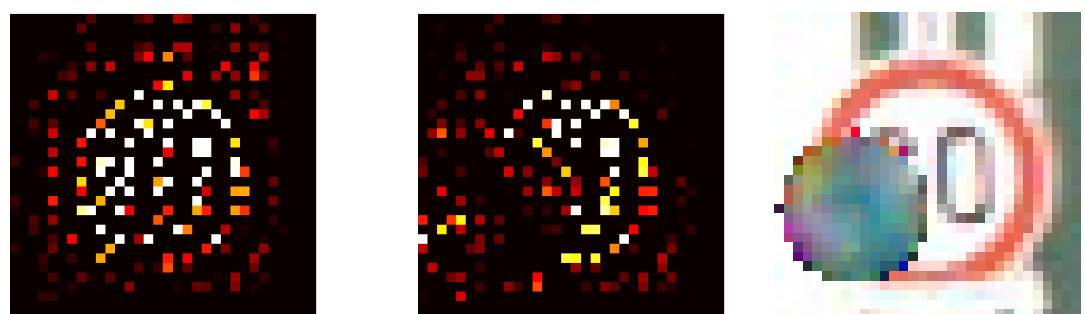
4. LRP-Parameter: exp=2, b=1, e=1e-6, e-rule



5. LRP-Parameter: exp=2, b=0,5, e=1e-6, b-rule



6. LRP-Parameter: exp=2, b=1, e=1e-6, b-rule



# **Eidesstattliche Erklärung**

Hiermit erkläre ich an Eides Statt, dass ich die vorliegende Arbeit selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht. Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

Sankt Augustin, den 18. März 2021

---

Mario Beckel