Hochschule Darmstadt

Fachbereiche Mathematik und Naturwissenschaften & Informatik

Analyse von Saliency Map Methoden mittels Adversarial Examples

Abschlussarbeit zur Erlangung des akademischen Grades
Master of Science (M. Sc.)
im Studiengang Data Science

Vorgelegt von:

Nicole Wagner

Matrikelnummer: 769903

Referent:	Prof. Dr. Horst Zisgen
Korreferent:	Prof. Dr. Arnim Malcherek
Ausgabedatum:	29. Januar 2024
Abgabedatum:	15. Juli 2024

Erklärung

Ich versichere hiermit, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die im Literaturverzeichnis angegebenen Quellen benutzt habe. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder noch nicht veröffentlichten Quellen entnommen sind, sind als solche kenntlich gemacht. Die Zeichnungen oder Abbildungen in dieser Arbeit sind von mir selbst erstellt worden oder mit einem entsprechenden Quellennachweis versehen. Diese Arbeit ist in gleicher oder ähnlicher Form noch bei keiner anderen Prüfungsbehörde eingereicht worden.

Frankfurt, 15. Juli 2024

_____

Nicole Wagner

Zusammenfassung

Für die Bildverarbeitung und -klassifikation sind Convolutional Neural Networks (CNNs) zu einer Schlüsseltechnologie geworden, die beeindruckende Erfolge in verschiedensten Anwendungen erzielt hat. Saliency Maps sind ein bekannter Ansatz, um Einblicke in die Entscheidungsprozesse von CNNs zu gewinnen, indem sie darstellen, welche Pixel eines Bildes für die Vorhersage eines Modells am relevantesten sind.

Diese Masterarbeit untersucht die Qualität der drei Saliency Map-Methoden Vanilla Gradient, Grad-CAM und Layer-wise Relevance Propagation (LRP) mit den beiden Zerlegungsregeln $\alpha\beta$-Regel und ε-Regel mit den Parametern $\alpha = 1$, $\alpha = 2$ und $\varepsilon = 1$.

Um die Qualität zu messen, werden Saliency Maps von Bildern und deren Adversarial Examples erzeugt und miteinander verglichen. Adversarial Examples sind minimal veränderte Versionen von Bildern, die ein Modell zu fehlerhaften Vorhersagen verleiten. Erzeugt ein Adversarial Example eine ähnliche Saliency Map wie sein Originalbild, so hat das Erklärmodell nicht auf das veränderte Bild und dessen Klassifikation reagiert, was auf eine geringe Aussagekraft hinweist. Zur Messung der Ähnlichkeit von Saliency Maps zwischen den Originalbildern und den Adversarial Examples wurde in dieser Arbeit die Wasserstein-Metrik als neue Kennzahl eingeführt. Darüberhinaus wurde das bekannte Relevance Ranking und der Spearman-Rangkorrelationskoeffizient verwendet.

Bei den Wasserstein-Metriken hat sich das Grad-CAM-Verfahren als beste Methode herausgestellt. Der Spearman-Rangkorrelationskoeffizient hingegen bewertet das LRP-Verfahren mit $\varepsilon = 1$ am besten. Das LRP-Verfahren mit $\alpha = 1$ schneidet beim Vergleich am schlechtesten ab. Ein weiteres Ergebnis dieser Masterarbeit ist, dass verschiedene Metriken zur Messung der Ähnlichkeit zwischen Original- und Adversarial Saliency Maps die Qualität der Saliency Map-Methoden unterschiedlich bewertet.

Abstract

For image processing and classification, Convolutional Neural Networks (CNNs) have become a key technology which has achieved impressive success in various applications. Saliency maps are a well-known approach to gain insights into the decision-making processes of CNNs by visualising which pixels of an image are most relevant for the prediction of the model.

This master's thesis analyses the quality of the three saliency map methods Vanilla Gradient, Grad-CAM and Layer-wise Relevance Propagation (LRP) with the two decomposition rules $\alpha\beta$-rule and ε-rule with the parameters $\alpha = 1$, $\alpha = 2$ and $\varepsilon = 1$.

To measure the quality, saliency maps of images and their adversarial examples are generated and compared with each other. Adversarial examples are minimally modified versions of images that lead a model to make incorrect predictions. If an adversarial example generates a similar saliency map to its original image, the explanatory model has not reacted to the altered image and its classification, which indicates a low informative value. To measure the similarity of saliency maps between the original images and the adversarial examples, the Wasserstein metric was introduced as a new key figure in this work. In addition, the well-known relevance ranking and the Spearman rank correlation coefficient were used.

For the Wasserstein metrics, the Grad-CAM method proved to be the best method. The Spearman rank correlation coefficient, on the other hand, rates the LRP method with $\varepsilon = 1$ as the best. The LRP method with $\alpha = 1$ performs worst in the comparison. Another result of this master's thesis is that different metrics for measuring the similarity between original and adversarial saliency maps evaluate the quality of the saliency map methods differently.

Inhaltsverzeichnis

Abkürzungsverzeichnis

Liste der Symbole

$\mathbf{1}_n$	Zeilenvektor mit n Einsen
α	Parameter für das LRP-Verfahren mit $\alpha\beta$-Regel
α_k^c	Gewichte für das Grad-CAM Verfahren
ε	Parameter für das LRP-Verfahren mit ε-Regel
θ	Einheintsvektor
ω_k	Gewichte des letzten Fully Connected Layers für das CAM-Verfahren
A	Matrix der Nebenbedingungen eines linearen Optimierungsproblems
A^k	Feature Map des letzten Convolutional Layers
A_{ij}^k	Fature der Reature Map A^k
C	Menge der Klassen
E^c	Erklärmodell für Klasse c
I	Bild
$\mathbb{I}_n$	n-dimensionale Einheitsmatrix
L^c	Saliency Map für Klasse c
S_c	Score-Funktion des CNN für die Klasse c
S^{d-1}	d-dimensionale Einheitssphäre
c	eine beliebige Klasse
c'	Zielklasse des Adversarial Examples
d	Vektor mit allen Kosten für den Transport zwischen zwei Verteilungen
d_{ij}	Kosten für den Transport von Masse von Pixel i zu Pixel j
f	Vektor mit allen Flüssen zwischen zwei Verteilungen
f_{ij}	Fluss von Masse von Pixel i zu Pixel j
r_i	Rang des Relevance Scores des Pixel i im Originalbild
s_i	Rang des Relevance Scores des Pixel i im Adversarial Example
v_i	Relevance Scores des Pixel i im Originalbild
v_i'	skalierter Relevance Scores des Pixel i im Originalbild
w_i	Relevance Scores des Pixel i im Adversarial Example
w_i'	Relevance Scores des Pixel i im Adversarial Example
x_i	Koordinatenvektor des Pixel i im Originalbild
y_i	Koordinatenvektor des Pixel i im Adversarial Example

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einleitung

Künstliche Intelligenz (KI) ist aus dem Alltag der Menschen nicht mehr wegzudenken. Es gibt diverse Machine Learning Algorithmen, die aus großen Datenmengen Erkenntnisse erlangen, soziale Netzwerke analysieren, Daten clustern, Vorhersagen treffen oder auch Bilder erkennen. Doch wie genau kommt eine KI eigentlich auf ihre Einschätzung? Welche Datenpunkte in einem Datensatz, Zeichen und Wörter in einem Text oder Pixel in einem Bild waren für die KI ausschlaggebend, die jeweilige Aussage zu treffen? Genau diese Fragen lassen sich unter dem Stichwort "explainable AI", zu deutsch "erklärbare KI", zusammenfassen [Gun+19]. Auch Regierungen haben bereits die Notwendigkeit erkannt, Erklärbarkeit von KI-Systeme zu verlangen. So formuliert die EU in ihrem AI Act, "dass KI-Systeme so entwickelt und verwendet werden, dass sie angemessen nachvollziehbar und erklärbar sind" [Eur24].

Für die Bildverarbeitung und -klassifikation sind Convolutional Neural Networks (CNNs) dabei zu einer Schlüsseltechnologie geworden, die beeindruckende Erfolge in verschiedensten Anwendungen erzielt hat [Agg18].

Saliency Maps sind ein bekannter Ansatz, um Einblicke in die Entscheidungsprozesse von CNNs zu gewinnen, indem sie darstellen, welche Pixel eines Bildes für die Vorhersage eines Modells am relevantesten sind [Ade+18; Bac+15; Sel+16a; SVZ13].

Mit dieser Arbeit wird die Anwendung von Adversarial Examples weiterverfolgt, um die Qualität von Saliency Map-Methoden zu messen. Dafür wird auf der Arbeit von Dieter und Zisgen [DZ23a] aufgebaut. Adversarial Examples sind minimal veränderte Versionen von Bildern, die das Modell zu fehlerhaften Vorhersagen verleiten [GSS14]. Von Dieter und Zisgen [DZ23a] wurden die Adversarial Examples genutzt, um die Reaktion des Erklärmodells LRP auf die Manipulation zu evaluieren. Erzeugt ein Adversarial Example eine ähnliche Saliency Map, wie sein Originalbild, so hat das Erklärmodell nicht auf das veränderte Bild und dessen Klassifikation reagiert, was auf eine geringe Aussagekraft hinweist.

Die Grundlage für die Analyse bildet das von Dieter und Zisgen [DZ23a] definierte CNN, welches auf dem CIFAR10 Datensatz [Kri09] trainiert wurde und die bereits erstellten Adversarial Examples von Dieter und Zisgen [DZ23a].

Diese Arbeit wendet die Analyse der Adversarial Examples nun zusätzlich auf die beiden Saliency Map-Methoden Vanilla Gradient [SVZ13] und Grad-CAM [Sel+16a] an. Darüber hinaus werden neue Metriken zur Messung der Unterschiede zwischen den Originalbildern und den Adversarial Examples etabliert. Neben dem häufig ge-

nutzten Spearman-Rangkorrelationkoeffizienten [Fah+23; Ade+18] und dem Relevance Ranking [DZ23a] wird die Wasserstein-Metrik [RTG98] genutzt. Sie wurde eingeführt, um die Ähnlichkeit von zwei Bildern zu messen und als Methode für die Bildersuche in Datenbanken anhand von Referenzbildern [RTG98; RTG00].

Ziel der Arbeit ist es, die Qualität der drei Erklärmodelle Vanilla Gradient, Grad-CAM und LRP mit dem Einsatz von Adversarial Examples und der Wasserstein-Metrik zu evaluieren und miteinander zu vergleichen. Durch eine systematische Analyse und Vergleich verschiedener Saliency Map-Methoden soll diese Arbeit einen Beitrag zur Verbesserung der Erklärbarkeit von CNNs leisten und Wege aufzeigen, wie die Qualität von Erklärmodellen besser bewertet werden kann.

Diese Masterarbeit gliedert sich in die folgenden Abschnitte. In Kapitel 2 wird eine Übersicht über die aktuelle Forschung im Bereich der Saliency Maps und der Metriken zur Messung der Qualität gegeben. Kapitel 3 gibt einen Überblick über die Grundlagen der Bilderkennung. Dabei werden die Themen Convolutional Neural Networks, Adversarial Examples und die genutzten Methoden der Saliency Maps vorgestellt. Es folgt in Kapitel 4 eine Einführung der statistischen Methoden und Kennzahlen zur Messung der Qualität der Saliency Maps mithilfe der Adversarial Examples. Insbesondere werden die neuen Wasserstein-Metriken eingeführt. Kapitel 5 stellt die tatsächliche Implementierung mit den genutzten Algorithmen und Technologien vor. Im Anschluss werden die Ergebnisse in Kapitel 6 zusammengefasst und in Kapitel 7 diskutiert. Die Arbeit wird mit einem Fazit und dem Ausblick auf weitere Forschungsansätze abgeschlossen.

2 Literaturrecherche

Mit der Fragestellung, wie der Mensch in die Vorgehensweisen von Blackbox-KI-Modellen schauen kann, haben sich zahlreiche Autoren beschäftigt. Darunter auch Chakraborty u. a. [Cha+17]. Sie definieren zunächst Erklärbarkeit damit, dass ein Anwender in der Lage sein soll, die Ergebnisse eines KI-Modells zu verstehen und zu interpretieren. Die Erklärbarkeit teilen sie dafür in zwei Dimensionen auf. Zum einen geht es um Modelltransparenz, also die Nachvollziehbarkeit von Berechnungen und Modellparametern, um das allgemeine Verständnis des Algorithmus. Gerade bei neuronalen Netzen ist das allerdings schwer. Daher gibt es die zweite Dimension: Modellfunktionalität. Sie bewertet, wie gut ein Modell das Ergebnis erklären kann. Das kann in Form von Texten, Visualisierungen oder lokalen Erklärungen geschehen. Letztere beschreiben dabei, wie sich das Ergebnis verändert, sobald lokale Änderungen am Input vorgenommen werden. Die in dieser Arbeit betrachteten Saliency Map-Methoden fallen daher in den Bereich der Modellfunktionalität.

Die Weiterentwicklung und Erforschung neuer Methoden zur Erstellung von Saliency Maps schreitet kontinuierlich voran. So wurde beispielsweise von Ahmed Asif Fuad u. a. [Ahm+20] die sogenannte Feature Explanation Method vorgestellt. Diese Methode berechnet Saliency Maps unabhängig von einer gewählten Klasse. Dabei wird, wie beim Grad-CAM-Verfahren, der letzte Convolution Layer betrachtet. Die Feature Maps werden in binäre Matrizen umgewandelt, wobei nur die relevantesten Features eine 1 erhalten. Alle Feature Maps werden mit einer gewichteten Summe zusammengefasst. Der Mittelwert der Feature Maps stellt jeweils das Gewicht dar. Diese Methode wurde von Bourroux u. a. [Bou+22] erweitert. Hierbei werden für jedes Convolution Layer in einem CNN diese Saliency Maps berechnet, welche wiederum mit verschiedenen Methoden zu einer finalen Saliency Map kombiniert werden.

Auch Wang u. a. [Wan+20] stellen eine neue Saliency Map-Methode vor, welche auf dem letzten Convolution Layer beruht. Ihr Score-CAM-Verfahren berechnet im Unterschied zum Grad-CAM-Verfahren allerdings keine Gradienten. Sie berechnen einen Score für jede Feature Map mittels eines Forwardpass durch das CNN zur Zielklasse. Die Linearkombination aus den Feature Maps und den berechneten Scores erzeugt die Saliency Map.

Allerdings reicht es nicht aus, Saliency Map-Methoden zu entwickeln. Es ist auch notwendig, ihre Qualität evaluieren zu können. Ghorbani, Abid und Zou [GAZ19] haben untersucht, ob Bilder so manipuliert werden können, dass sie keine erkenn-

baren Unterschiede zum Originalbild aufweisen und die gleiche Klasse bestimmen, jedoch eine abweichende Saliency Map erzeugen. Erklärmodelle gelten als fragil, wenn entsprechende Bilder existieren. Die Ähnlichkeit von zwei Saliency Maps wurde dafür mit dem Spearman-Rangkorrelationskoeffizienten und der Schnittmenge der Top-Pixel gemessen. Für alle untersuchten Methoden, unter welchen auch das hier betrachtete Vanilla Gradient-Verfahren war, war die Erstellung solcher manipulierten Bilder möglich.

Ein umfangreicher Vergleich von Saliency Map-Methoden wurde von Adebayo u. a. [Ade+18] durchgeführt. Ihre Zuverlässigkeitsprüfung basiert auf zwei Verfahren. Zum einen randomisieren sie die Modellparameter des genutzten CNN und vergleichen die Saliency Maps mit dem korrekt trainierten Modell. Unterscheiden sich die Saliency Maps nicht voneinander, so sind die Methoden nicht abhängig vom Modell und fallen im Test durch. Zum anderen randomisieren sie die Trainingsdaten. Dafür trainieren sie ein weiteres Modell mit den gleichen Daten, aber zufällig verteilten Labeln. Auch hier müssen sich die Saliency Maps vom Originalmodell unterscheiden. Für den Vergleich der Saliency Maps wurde unter anderem wieder der Spearman-Rangkorrelationskoeffizient verwendet. Die Verfahren Vanilla Gradient und Grad-CAM konnten die Tests bestehen.

Heo, Joo und Moon [HJM19] stellen eine weitere Variante zur Qualitätsüberprüfung vor. Auch sie manipulieren das zugrundeliegende CNN. Dabei passen sie die Modellparameter gezielt an, um die Ergebnisse eines Erklärmodells drastisch zu ändern, aber die Accuracy des Modells beizubehalten. Ein Ergebnis ihrer Untersuchungen war, dass nicht nur das Erklärmodell, welches zur Anpassung der Parameter verwendet wurde, abweichende Saliency Maps erzeugt hat, sondern dass auch andere Erklärmodelle davon beeinflusst wurden. Darüber hinaus wurde in einem zweiten Ansatz überprüft, wie die Erklärmodelle auf Bilder reagieren, die mehrere Klassen beinhalten. Sie haben gezeigt, dass ein Datensatz von zwei CNNs, die sich nur in den Entscheidungsgrenzen unterscheiden, mit der gleichen Accuracy klassifiziert werden kann, aber die zugehörigen Saliency Maps fundamentale Unterschiede aufweisen.

Die Autoren Tomsett u. a. [Tom+20] erweitern den Gedanken der Qualitätskontrolle von Saliency Maps. Anstatt Metriken zu definieren, um die Qualität zu messen, wollen sie die Qualität der Metriken überprüfen. Dafür stellen sie drei Prüfansätze vor. Erstens sollen Metriken überprüft werden, ob sie die Saliency Maps einer Methode ähnlich bewerten. Zweitens soll eine Metrik daraufhin überprüft werden, ob sie die Saliency Maps verschiedener Methoden für das gleiche Bild ähnlich bewertet. Drittens wird die Korrelation verschiedener Metriken für eine Methode überprüft. Bei Untersuchung verschiedener Qualitätsmetriken für mehrere Saliency Map-Methoden konnten sie Inkonsistenzen erkennen.

Dieter und Zisgen [DZ23a] nutzen Adversarial Examples, um falsche Klassifikationen zu erzeugen und haben die Saliency Maps des LRP-Verfahrens überprüft. Dabei wurden die Saliency Maps der Originalbilder mit den Saliency Maps der Adversarial Examples verglichen. Um die Ähnlichkeit zu quantifizieren, wurden unter anderem die Verschiebungen der Ränge ausgewertet. Das Ergebnis war, dass keine nennens-

werten Unterschiede zwischen den Adversarial Examples und den Originalbildern festgestellt werden konnten. Die Saliency Maps des LRP-Verfahrens scheinen in jedem Bild hauptsächlich Kanten zu markieren.

Einen Vorschlag, um bessere Saliency Maps zu erzeugen machen Serrurier u. a. [Ser+22]. Sie beobachten, dass Saliency Maps oft ein hohes Maß an Rauschen aufweisen, was für menschliche Anwender oft nicht aussagekräftig ist. Daher nutzen sie zusätzliche Bedingungen während des Trainings eines neuronalen Netzes und erhalten ein sogenanntes 1-Lipschitz neuronales Netz. Dies stellt ebenfalls ein robustes Netz dar und erzeugt Saliency Maps, welche weniger Rauschen enthalten und deutlich auf die wesentlichen Bereiche des Bildes konzentriert sind. Die neuen Saliency Maps wurden mit den von Menschen als relevant identifizierten Bildregionen verglichen. Diese Vergleiche zeigten eine hohe Übereinstimmung.

Die Literaturrecherche hat ergeben, dass es sowohl eine Vielzahl von Saliency Map-Methoden gibt als auch unterschiedliche Ansätze, diese zu evaluieren. Hierbei spielen unterschiedliche Faktoren eine Rolle. Zum einen ist die Verlässlichkeit fundamental. Manipulierte Bilder oder CNNs sollten unterschiedliche Saliency Maps erzeugen. Gleichzeitig müssen Saliency Maps auch für Menschen nützlich sein. Diese Masterarbeit untersucht die Verlässlichkeit verschiedener Methoden. Die Grundlage dafür bildet die Arbeit von Dieter und Zisgen [DZ23a].

3 Grundlagen der Bilderkennung

In diesem Kapitel werden die Grundlagen der Bilderkennung zusammengefasst. Dabei erfolgt eine kurze Einführung in Convolutional Neural Networks. Anschließend wird erklärt, was Adversarial Examples sind und wie sie generiert werden können. Abschließend werden die in dieser Arbeit verwendeten Erklärmodelle Vanilla Gradient, Grad-CAM und Layer-wise Relevance Propagation vorgestellt.

3.1 Convolutional Neural Networks

Neuronale Netze sind die wesentlichen Deep Learning-Modelle [GBC16]. In verschiedenen Schichten, auch Layer genannt, kombinieren sie lineare und nichtlineare Funktionen. Die Daten gelangen dabei über einen Inputlayer in das Modell und der Outputlayer liefert das berechnete Ergebnis. Die Layer dazwischen werden Hidden Layer genannt. Die Modellparameter werden vom Modell anhand von klassifizierten Trainingsdaten gelernt [GBC16; SPV18]. In den Layern werden die Input-Informationen in der Regel in mehreren Knoten mit einer linearen Funktion zusammengerechnet und durch eine nichtlineare Aktivierungsfunktion an den nächsten Layer weitergegeben.

Convolutional Neural Networks (CNNs) sind eine spezielle Form von neuronalen Netzen, die zur Verarbeitung von Daten in einem Rasterformat, wie beispielsweise Bildern, verwendet werden. Der Unterschied zu normalen neuronalen Netzen besteht in der Anwendung der mathematischen Operation Convolution, im Deutschen auch Faltung genannt, in mindestens einem Layer [GBC16].

Da diese Arbeit direkt auf der Masterarbeit [Die20] aufbaut und der Fokus auf der Auswertung weiterer Saliency Map-Methoden mit neuen Metriken liegt, werden die Grundlagen zu CNNs nur rudimentär zusammengefasst. Für detailliertere Informationen sei auf die Lehrbücher [Bis09; GBC16; Agg18] verwiesen.

Fully Connected Layer Fully Connected Layer werden manchmal auch Dense Layer genannt [Agg18; Aba+15]. Dabei werden alle Inputfeatures mit allen Knoten verbunden. In einem CNN wird dieser Layer in der Regel als Output Layer verwendet. Er dient der Bestimmung der Scores für jede Klasse. Die Anzahl der Knoten eines Fully Connected Layers entspricht daher auch der Anzahl an Klassen. Um für

jede Klasse Wahrscheinlichkeiten zu bestimmen, wird das Fully Connected Layer beispielsweise mit einer Softmax-Aktivierungsfunktion verbunden [GBC16].

Aktivierungsfunktionen Bevor die Ergebnisse einer linearen Funktion in einem neuronalen Netz an das nächste Layer weitergegeben werden, durchlaufen sie eine nichtlineare Aktivierungsfunktion [GBC16]. Eine häufige Aktivierungsfunktion ist Rectified Linear Unit (ReLU). Sie wird elementweise angewendet und behält nur die positiven Zahlen.

$$\mathrm{ReLU}(z) = \max\{0, z\}$$

Für Klassifikationen dient Softmax der Ermittlung der Wahrscheinlichkeiten für alle Klassen. Seien dafür $z_0, \dots z_{N-1}$ die Outputs des vorherigen Layers. Dann wird der Softmax-Wert für z_i berechnet durch

$$\mathrm{softmax}(z_i) = \frac{\exp(z_i)}{\sum_j \exp(z_j)}.$$

Convolution Layer Convolution Layer sind die namensgebenden Layer eines CNN. Sie kombinieren einen Input, beispielsweise ein Bild, mit einem Kernel und erzeugen dadurch gemeinsame Werte für Bildregionen. Diese neuen Werte werden auch Feature genannt. Alle Werte zusammen ergeben eine Feature Map. Durch dieses Verfahren können beispielsweise Strukturen in Bildern identifiziert werden. Die Ergebnisse werden in der Regel durch eine Aktivierungsfunktion an ein sogenanntes Pooling Layer übergeben. Pooling Layer dienen der Reduktion von Komplexität, indem sie für Regionen wieder gemeinsame Werte berechnen [GBC16]. Abbildung 1 zeigt eine schematische Darstellung dieses Vorgehens.

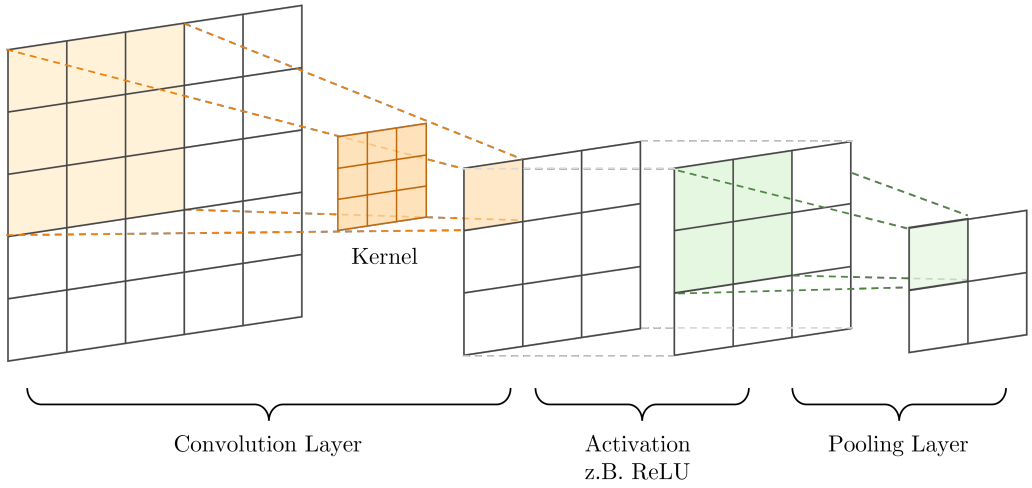


Abbildung 1.: Schematische Darstellung eines Abschnitts mit Convolution, Activation und Pooling nach [Die20]

Übliche Pooling-Verfahren sind zum Beispiel das Max-Pooling, wobei für eine Region der Maximalwert behalten wird, oder das Global Average Pooling, wobei der Durchschnitt berechnet wird.

Convolution sowie Pooling Layer reduzieren die Komplexität des CNN, da sie weniger Features an das nächste Layer übergeben, als sie selbst erhalten haben [GBC16].

Forwardpass Wird für ein gegebenes Bild I das Ergebnis des CNN berechnet, so wird dieser Prozess auch Forwardpass genannt.

Backpropagation Backpropagation beschreibt einen Algorithmus, den Gradienten zu einem Ergebnis in neuronalen Netzen zu berechnen. Während des Trainings eines neuronalen Netzes wird der Gradient einer Loss-Funktion berechnet. Die Loss-Funktion quantifiziert, wie groß die Abweichung des berechneten Ergebnisses zum tatsächlichen Label der Trainingsdaten ist. Um die Abweichung zu verringern, werden die Parameter des neuronalen Netzes schrittweise mit Stochatsic Gradient Descent optimiert. Backpropagation ist allerdings ein Algorithmus, der für beliebige Funktionen einen Gradienten bestimmen kann. Dafür verwendet Backpropagation die Kettenregel, wobei eine effiziente Reihenfolge für die Berechnung verwendet wird [GBC16]. Für Details zur Backpropagation sei wieder auf die Lehrbücher [Bis09; GBC16; Agg18] verwiesen.

3.2 Adversarial Examples

Neuronale Netze erreichen mittlerweile herausragende Ergebnisse bei der Bildklassifikation. Allerdings gibt es Möglichkeiten, das neuronale Netz zu überlisten [GSS14; SPV18]. Eine Methode dafür sind sogenannte Adversarial Examples [Sze+14]. Für Adversarial Examples werden Bilder, die vom CNN klassifiziert werden können, minimal verändert, um ein anderes Klassifikationsergebnis zu erhalten. Für Menschen können die Änderungen zwar sichtbar sein, doch die Klasse des Bildes ist weiterhin eindeutig zu erkennen. Daher eignen sich diese Bilder sehr gut, um CNNs weiter zu untersuchen [SPV18]. Adversarial Examples werden mittlerweile auch während des Trainingsprozesses eingesetzt, um CNNs zu verbessern [GSS14]. Allgemein lässt sich ein Adversarial Example als Ergebnis eines Optimierungsproblems beschreiben [Sze+14; SPV18]. Sei hierfür $I \in \mathbb{R}^{n \times m \times 3}$ ein Bild mit $n \times m$ Pixeln und 3 Farbwerten, c seine Klasse aus der Menge der Klassen C und $g : \mathbb{R}^{n \times m \times 3} \to C$ die trainierte Funktion, die einem Bild I eine Klasse c zuordnet. Dann ergibt sich ein minimal verändertes Bild I' für die Zielklasse c' durch die Lösung des folgenden linearen

Optimierungsproblems:

$$\begin{aligned} &\min_{\eta} \|I - I'\| \\ &\text{s.t. } g(I') = c' \\ &\quad g(I) = c \\ &\quad c \neq c' \\ &\quad L \leq I' \leq U. \end{aligned}$$

Da es sich bei dem Adversarial Example I' um ein Bild handelt, beschreiben die Vektoren L und U die untere- und obere Grenze der Pixelwerte. Das Ergebnis η beschreibt den Unterschied zwischen dem Originalbild und dem gesuchten Adversarial Example $\eta = I - I'$.

Da die Lösung des Optimierungsproblems rechenaufwändig ist, existieren verschiedene Approximationsverfahren um Adversarial Examples zu erzeugen [SPV18]. Für die Auswertungen dieser Masterarbeit wurden die zur Verfügung gestellten Adversarial Examples der Masterarbeit [Die20] bzw. des Papers [DZ23a] genutzt. Diese wurden mittels L-BFGS Attack erstellt. Für detaillierte Informationen zur Methode wird auf [Sze+14] verwiesen.

3.3 Saliency Maps

Saliency Maps sind eine Methode um die Modellfunktionalität von CNN zur Bildklassifikation zu erklären [GAZ19]. Eine Saliency Map zeigt in der Regel, wie wichtig die einzelnen Pixel eines Bildes für die Klassifikation eines Bildes sind. Es gibt auch Verfahren, welche die Wichtigkeit für Features einer Feature Map aus einer Zwischenschicht berechnen. Die Wichtigkeiten, auch Relevance Scores genannt, werden häufig als Heatmap visualisiert. In der weiteren Arbeit wird aus Gründen der Vereinfachung in der Regel nur der Begriff Pixel verwendet. Für die Berechnung der Relevance Scores gibt es eine Vielzahl von Methoden, die sich deutlich voneinander unterscheiden [Anc+17]. In dieser Arbeit werden die Methoden Grad-CAM [Sel+16b], Vanilla Gradient [SVZ13] und Layer-wise Relevance Propagation [Bac+15] untersucht. Hierfür sei zunächst allgemein ein Erklärmodell für Bilder mit Saliency Maps definiert.

Definition 3.1 (Erklärmodell)**.** Gegeben sei ein CNN zur Bilderkennung von N Klassen. Sei $I \in \mathbb{R}^{3\times n\times m}$ ein Bild mit n mal m Pixeln und drei Farbdimensionen und $c \in 0, \ldots, N-1$ eine Klasse des CNN. Ein Erklärmodell ist eine Abbildung

$$\begin{aligned} E^c : \mathbb{R}^{3\times n\times m} &\to \mathbb{R}^{n'\times m'} \\ I &\mapsto L^c \end{aligned}$$

welche einem Bild I für eine Klasse c eine Saliency Map L^c zuordnet. Dabei erhält jeder Pixel i des Bildes einen Relevance Score $w_i \in \mathbb{R}$. Die Dimension der Saliency Map ist kleiner oder gleich der Dimension der Pixel im Bild.

In allen drei hier untersuchten Erklärmodellen werden für die Ermittlung der Relevance Scores die Ergebnisse der Score-Funktion für eine Klasse genutzt.

Definition 3.2 (Score-Funktion)**.** Gegeben sei ein CNN zur Bilderkennung von N Klassen. Sei $I \in \mathbb{R}^{3\times n\times m}$ ein Bild mit n mal m Pixeln und drei Farbdimensionen und $c \in 0, \ldots, N-1$ eine Klasse des CNN. Dann definiert

$$\begin{aligned} S_c : \mathbb{R}^{3\times n\times m} &\to \mathbb{R} \\ I &\mapsto S_c(I) \end{aligned}$$

eine Score-Funktion des Bildes I für die Klasse c im CNN. Dabei ist der Score für das Bild das Ergebnis des letzten Layers vor der Softmax-Berechnung.

3.3.1 Vanilla Gradient

Unter dem Namen *Image-Specific Class Saliency* wurde von [SVZ13] eines der ersten Erklärmodelle für Bilder vorgestellt [Mol22]. Es ist in der Literatur auch als *gradient explanation* [Ade+18] oder *SimpleGrad* [HJM19] bekannt und wird in dieser Arbeit als *Vanilla Gradient* nach [Mol22] benannt. Vanilla Gradient berechnet die Gradienten der Score-Funktion S_c für ein gegebenes Bild I_0 und die gewünschte Klasse c [SVZ13]. Der Gradient soll ein Indikator für die Relevanz eines Pixels sein, da eine kleine Änderung von Pixeln mit einem hohen absoluten Gradienten einen großen Einfluss auf den Klassen-Score hat.

Um die Vanilla Gradient Saliency Map L^c_{VG} zu bestimmen, wird zunächst der Klassen-Score $S_c(I_0)$ für ein gegebenes Bild I_0 und die gewünschte Klasse c mit einem Forwardpass durch das CNN berechnet. Die Gradienten berechnen sich mittels Backpropagation des Klassen-Scores durch das CNN:

$$w = \left.\frac{\partial S_c}{\partial I}\right|_{I_0}$$

Das Ergebnis sind Ableitungen für jeden Farbwert eines jeden Pixels im Bild I_0. Von Simonyan, Vedaldi und Zisserman [SVZ13] werden nur die absoluten Gradienten betrachtet. Um zuletzt einen einzelnen Wert für jeden Pixel zu bestimmen, wählen Simonyan, Vedaldi und Zisserman [SVZ13] den größten absoluten Gradienten aus den Farbwerten je Pixel:

$$w_i = \max\left(\left\{|w_{(i,\text{rot})}|, |w_{(i,\text{grün})}|, |w_{(i,\text{blau})}|\right\}\right).$$

3.3.2 Grad-CAM

Das Gradient-weighted Class Activation Map (Grad-CAM)-Verfahren wurde von Selvaraju u. a. [Sel+16a] entwickelt. Ziel des Verfahrens ist es, diejenigen Regionen im Bild zu identifizieren, die im letzten Layer des CNN aktiviert werden. Dafür werden die Feature Maps des letzten Convolution Layers betrachtet [Mol22].

Grad-CAM ist eine Weiterentwicklung des Class Activation Map (CAM)-Verfahrens [Zho+15]. Das Class Activation Mapping setzt eine bestimmte CNN-Architektur voraus. Dieses besteht aus einem oder mehreren Convolutional Layern. Nach dem letzten Convolutional Layer werden die Feature Maps mittels Global Average Pooling zusammengefasst und in ein letztes Fully Connected Layer übergeben. Dieses ermittelt aus den Inputwerten und den gelernten Gewichten die Scores für die jeweiligen Klassen. Die Class Activation Map ergibt sich nun als gewichtete Summe der Feature Maps aus dem letzten Convolutional Layer mit den jeweils gelernten Gewichten für die gewünschte Klasse.

Seien A^k mit $k = 1, \ldots, K$ die Feature Maps des letzten Convolutional Layers und ω_k^c die gelernten Gewichte des letzten Layers zwischen Knoten k und Klasse c. Dann definiert

$$L_{CAM}^c = \sum_k \omega_k^c A^k$$

die Class Activation Map für die Klasse c.

Da das CAM-Verfahren nur für diese besondere CNN-Architektur verwendet werden kann, wurde es von [Sel+16a; Sel+16b] verallgemeinert. Dabei ersetzen die Autoren die gelernten Gewichte durch Gradienten.

Sei S_c der Score der Klasse c und seien A_{ij}^k die Features der Feature Maps A^k des letzten Convolutional Layers. Die Gradienten der Features für die Score-Funktion im letzten Convolutional Layer sind $\frac{\partial S_c}{\partial A_{ij}^k}$. Dann ergibt sich das Gewicht für die gesamte Feature Map durch ein Global Average Pooling der Gradienten

$$\alpha_k^c = \frac{1}{K} \sum_i \sum_j \frac{\partial S_c}{\partial A_{ij}^k}$$

Für die finale Gradient-weighted Class Activation Map für eine Klasse c wird nun die gewichtete Summe der Feature Maps berechnet und die negativen Werte werden mittels ReLU auf Null gesetzt.

$$L_{Grad-CAM}^c = \text{ReLU}\left(\sum_k \alpha_k^c A^k\right)$$

Die Autoren Selvaraju u. a. [Sel+16a] haben sich für die Verwendung der ReLU-Funktion entschieden, um nur Pixel mit einem positiven Einfluss auf die Klasse zu

erhalten. Wird der Wert dieser Pixel erhöht, so erhöht sich auch der Score für die gewünschte Klasse. Es ist allerdings nicht ausgeschlossen, dass $\sum_k \alpha_k^c A^k$ nur Werte kleiner oder gleich Null ergibt. In diesem Fall sorgt die ReLU-Funktion dafür, dass alle Werte der Saliency Map die Zahl 0 annehmen.

Da die Feature Maps des letzten Convolutional Layers eine geringere Dimension haben als das Inputbild, gibt es zwei Varianten, das Ergebnis auf das Bild zu projizieren:

1. Upsampling der Saliency Map auf die Dimension des Inputbildes.
2. Punktweise Multiplikation mit Guided Backpropagation. Dies beschreibt ein weiteres Verfahren und wird in dieser Arbeit nicht weiter betrachtet.

Da die Saliency Maps in dieser Arbeit mit statistischen Methoden und verschiedenen Metriken ausgewertet werden, wird die Saliency Map des Grad-CAM-Verfahrens in geringerer Dimension beibehalten und kein Upsampling durchgeführt.

3.3.3 Layer-wise Relevance Propagation

Layer-wise Relevance Propagation (LRP) ist ein weiterer Algorithmus zur Bestimmung einer Saliency Map. Dabei werden die Relevance Scores jedes Pixels für eine Klasse c mit einem sogenannten Backwardpass berechnet. Dafür wird das Ergebnis rückwärts durch das CNN geschickt. In diesem Kontext wird das Ergebnis für eine Klasse des CNN auch als initialer Relevance Score für ein Bild bezeichnet. Bei jeder Übergabe des initialen Relevance Scores von einer Schicht zur vorherigen wird der Relevance Scores in mehrere neue Werte zerlegt. Dieses Vorgehen wird wiederholt, bis man bei der Zerlegung im Input Layer angekommen ist und die Relevance Scores für alle Pixel, genauer gesagt für alle Farbwerte aller Pixel, erhält [Bac+15; Anc+17; DZ23a]. Abbildung 2 aus dem Paper [DZ23a] zeigt eine schematische Darstellung dieses Vorgehens.

Es ist notwendig zu definieren, wie der Relevance Score eines Layers aus den Relevance Scores des nachfolgenden Layers berechnet wird oder umgekehrt formuliert, wie ein Relevance Score in einen vorherigen Layer zerlegt wird. Hierfür wurden von Bach u. a. [Bac+15] mehrere Varianten definiert. Diese Arbeit baut auf den Auswertungen von Dieter und Zisgen [DZ23a; Die20] und verwendet daher die sogenannte ε-Regel (LRP-ε) und die $\alpha\beta$-Regel (LRP-$\alpha\beta$). Exakte Definitionen der Zerlegungsregeln sind in [Bac+15; DZ23a] zu finden. Es sei an dieser Stelle erwähnt, dass für LRP-ε ein Parameter $\varepsilon > 0$ gewählt wird und für LRP-$\alpha\beta$ der Parameter $\alpha \geq 1$. Für die Berechnung einer Saliency Map wird immer eine Regel mit einem festen Parameter für die Zerlegung in allen Schritten verwendet.

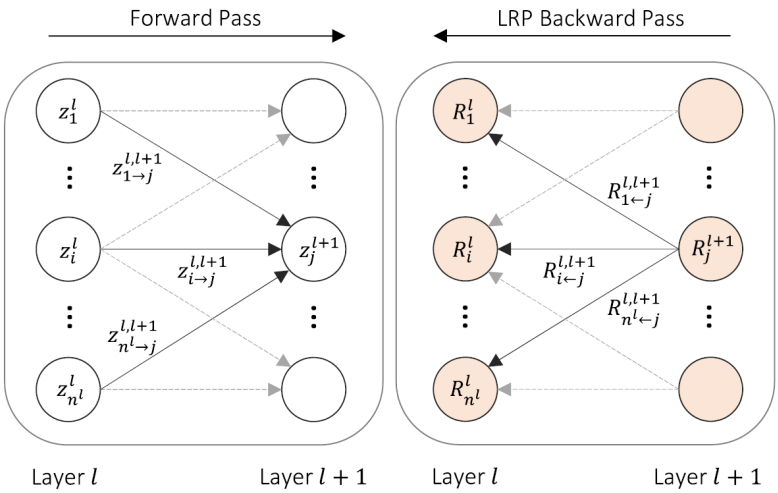


Abbildung 2.: Schematische Darstellung des Forwardpass und Backwardpass beim LRP aus [DZ23a]

4 Grundlagen der statistischen Methoden und Metriken

In dieser Arbeit werden die Saliency Maps der Originalbilder mithilfe verschiedener statistischer Methoden und Metriken mit den Saliency Maps der entsprechenden Adversarial Examples verglichen. Die Unterschiede werden dafür mit dem Spearman-Rangkorrelationskoeffizienten, dem Relevance Ranking und der Wasserstein-Metrik quantifiziert. Weitere statistische Kennzahlen werden für jede Saliency Map einzeln berechnet. Diese werden im Folgenden vorgestellt.

4.1 Spearman-Rangkorrelationskoeffizient

Der Rangkorrelationskoeffizient ist eine Maßzahl für den Grad des Zusammenhangs zweier Stichproben [BT94; Fah+23]. Der Wertebereich liegt zwischen -1 und 1.

Seien $x_1, \ldots, x_n$ und $y_1, \ldots, y_n$ ordinale Stichprobenwerte zweier Zufallszahlen. Für die beiden Stichproben werden die Ränge getrennt gebildet. Seien $r_1, \ldots, r_n$ die Ränge von $x_1, \ldots, x_n$ und $s_1, \ldots, s_n$ die Ränge von $y_1, \ldots, y_n$. Aus den Rängen wird nun der Korrelationskoeffizient nach Pearson berechnet.

$$r_s = \frac{\sum_{i=1}^{n} (r_i - \overline{r})(s_i - \overline{s})}{\sum_{i=1}^{n} (r_i - \overline{r})^2 \sum_{i=1}^{n} (s_i - \overline{s})^2}$$

Für die Ränge gilt nach der Gaußschen Summenformel außerdem

$$\sum_{i=1}^{n} r_i = \sum_{i=1}^{n} i = \frac{n(n+1)}{2} = \sum_{i=1}^{n} s_i$$

und

$$\overline{r} = \frac{1}{n} \sum_{i=1}^{n} r_i = \frac{1}{n} \sum_{i=1}^{n} i = \frac{n+1}{2} = \overline{s}.$$

Des Weiteren sei

$$d_i := r_i - s_i = \left(r_i - \frac{n+1}{2}\right) - \left(s_i - \frac{n+1}{2}\right)$$
$$\Rightarrow \sum_{i=1}^{n} d_i^2 = \frac{(n-1)\,n\,(n+1)}{6} - \frac{(n-1)\,n\,(n+1)}{6} r_s.$$

Daraus ergibt sich für den Korrelationskoeffizienten nach Spearman die Darstellung

$$r_s = 1 - \frac{6 \sum_{i=1}^{n} d_i^2}{(n-1)\,n\,(n+1)}$$

Der Spearman-Rangkorrelationskoeffizient ist ein Maß für den monotonen Zusammenhang zweier Stichproben [Fah+23]. In diesem Fall handelt es sich bei den Stichproben um die Relevance Scores der Pixel. Der Spearman-Rangkorrelationskoeffizienten wird dabei für jedes Originalbild und dessen Adversarial Examples berechnet. Ein positiver Wert deutet auf einen positiven Zusammenhang hin. Das bedeutet, dass Pixel, die im Originalbild einen hohen Relevance Score und somit einen hohen Rang haben, auch im Adversarial Example einen hohen Rang aufweisen. Ein negativer Wert deutet hingegen auf einen negativen Zusammenhang hin. Das heißt, dass Pixel im Adversarial Example, die im Originalbild einen hohen Relevance Score haben, einen geringen Relevance Score aufweisen und umgekehrt. Ein Rangkorrelationskoeffizient nahe Null zeigt, dass es keinen monotonen Zusammenhang zwischen den Saliency Maps gibt. Tabelle 1 gibt einen Überblick über die Einteilung der Effektstärke des Spearman-Rangkorrelationskoeffizienten [Fah+23].

$\lvert r_s \rvert < 0,5$	schwache Korrelation
$0,5 \leq \lvert r_s \rvert < 0,8$	mittlere Korrelation
$0,8 \leq \lvert r_s \rvert$	starke Korrelation

Tabelle 1.: Effektstärke des Spearman-Rangkorrelationskoeffizienten

4.2 Relevance Ranking

Das Relevance Ranking wird analog zu [DZ23a] erstellt. Der erste Schritt des Relevance Rankings ist es, zu bestimmen, welche Pixel im Originalbild am wichtigsten sind. Es wird dann untersucht, in welchem Maße sich der Top-Pixel im Rang des Adversarial Examples verschiebt. Dabei wird wie folgt vorgegangen: Analog zum Spearman-Rangkorrelationskoeffizienten werden für die Relevance Scores die Ränge gebildet. Seien $x_1, \ldots, x_n$ die Relevance Scores eines Originalbildes und $y_1, \ldots, y_n$ die Relevance Scores eines zugehörigen Adversarial Examples. Dabei ist n die Anzahl der Pixel in der Saliency Map. Für beide Beobachtungen werden jeweils die Ränge

gebildet. Dabei werden die Relevance Scores absteigend sortiert. Rang 1 beschreibt jeweils den größten Relevance Score und Rang n den geringsten. Seien $r'_1, \ldots, r'_n$ die absteigenden Ränge von $x_1, \ldots, x_n$ und $s'_1, \ldots, s'_n$ die absteigenden Ränge von $y_1, \ldots, y_n$. Um die Zuordnung eines Relevance Scores zu seinem Rang anzuzeigen, sei folgende Notation definiert: $x_i = x_{(r'_i)}$. Der größte Relevance Score ist somit $x_{(1)}$ im Originalbild und $y_{(1)}$ im Adversarial Example.

Die Relevanzverschiebung für ein Pixel i ergibt sich aus der Differenz der Ränge $d'_i = s'_i - r'_i$. Negative Werte bedeuten somit eine höhere Relevanz des Pixels im Adversarial Example. Positive Werte bedeuten eine niedrigere Relevanz des Pixels im Adversarial Example.

Für die statistische Auswertung wird die absolute Relevanzverschiebung des Pixels mit dem größten Relevance Score im Originalbild $x_{(1)}$ zu jedem Adversarial Example bestimmt. Ein kleiner Wert deutet darauf hin, dass der Pixel im Adversarial Example auch eine hohe Relevanz hat. Es ist ein Indiz für ähnliche Saliency Maps.

Betrachtet werden außerdem die 10 % wichtigsten Pixel des Originalbildes und des Adversarial Examples, also jeweils diejenigen Pixel mit dem größten Relevance Score. Dabei ist die Schnittmenge beider Pixelmengen von Bedeutung, da diese ebenfalls Auskunft darüber gibt, ob die gleichen Pixel jeweils relevant sind.

Seien $X_{10} = \{i | r_i < 0.1 \cdot n\}$ und $Y_{10} = \{j | s_j < 0.1 \cdot n\}$ die Mengen der 10 % größten Relevance Scores des Originalbildes und des Adversarial Examples. Dann ist der Anteil der gleichen Pixel im Originalbild und im Adversarial Example unter den Top-10 %-Pixeln gegeben durch $\frac{|X_{10} \cap Y_{10}|}{|X_{10}|}$. Je höher der Anteil, desto ähnlicher sind die Saliency Maps. Insbesondere wird auch ermittelt, in wie vielen Fällen der Top-Pixel des Originalbildes unter den Top-10 %-Pixeln des Adversarial Examples ist.

4.3 Wasserstein-Metrik

Die Wasserstein-Metrik (WS) beschreibt die Ähnlichkeit zwischen zwei Verteilungen [RTG98]. Sie wird auch Earth Mover's Distance (EMD) genannt und beruht auf dem sogenannten Transportproblem. Die Ursprungsidee für das Transportproblem stammt aus der Verteilung von Waren eines Anbieters aus mehreren Fabriken zu den Konsumenten [Hit41]. Die Lieferung soll dabei möglichst optimal erfolgen. Allgemeiner formuliert soll Masse optimal von verschiedenen Quellen zu verschiedenen Zielen transportiert werden. Minimiert wird der Aufwand, um die Verteilung von Masse bei den Quellen in die Verteilung der Masse der Ziele zu überführen. Der Aufwand ist dabei die Summe aller Massen, die bewegt werden, multipliziert mit der Distanz zu ihrem Zielort. Für das Minimierungsproblem soll so wenig Masse wie möglich über die kürzesten Distanzen bewegt werden. Für die Lösung des Transportproblems muss daher zunächst ein Distanzmaß gewählt werden. Die Distanz wird

auch Kosten des Transports genannt. Formal kann das Transportproblem wie folgt definiert werden [RTG98].

Definition 4.1 (Transportproblem)**.** Seien $\mathcal{I} = \{x_i | i \in I\}$ die Menge an Quellen und $\mathcal{J} = \{y_j | j \in J\}$ die Menge der Ziele. Die Masse einer Quelle $x_i \in \mathcal{I}$ sei v_i und die Masse eines Ziels $y_j \in \mathcal{J}$ sei w_i. Seien d_{ij} die Kosten für den Transport von Masse von x_i nach y_j. Dann wird das Transportproblem definiert durch

$$\begin{aligned} \min \quad & \sum_{i \in I} \sum_{j \in J} d_{ij} f_{ij} & \\ \text{s.t.} \quad & f_{ij} \geq 0 & \forall i \in I, j \in J \\ & \sum_{j \in J} f_{ij} = v_i, & \forall i \in I \\ & \sum_{i \in I} f_{ij} = w_j & \forall j \in J \end{aligned}$$

wobei f_{ij} den Fluss von Masse der Quelle x_i an das Ziel y_j beschreibt.

Dabei kann der Fluss f_{ij} auch nur einen Teil der Masse einer Quelle v_i enthalten und auch nur einen Anteil der Masse in einem Ziel w_j ausmachen. Es sei an dieser Stelle erwähnt, dass die Definition die Annahme beinhaltet, dass die Summe der Masse der Quellen und die Summe der Masse am Ziel gleich groß sind. Es ist allgemein auch möglich, dass die Quelle insgesamt mehr Masse enthält als das Ziel. In diesem Fall lautet die letzte Bedingung $\sum_{j \in J} f_{ij} \leq v_i$. Da im weiteren Verlauf nur Transportprobleme und Wasserstein-Metriken für Quellen und Ziele mit gleichen Massen berechnet werden, wurde hier auf die allgemeine Formulierung verzichtet.

Lineare Optimierungsprobleme werden häufig auch in einer Matrix-Vektor-Schreibweise definiert. Das Transportproblem lässt sich somit formulieren durch

$$\begin{aligned} \min \quad & d^T f \\ \text{s.t.} \quad & f \geq 0 \\ & Af = b. \end{aligned}$$

Dabei stellt f den Vektor mit allen Flüssen dar. Sei dafür $F = (f_{ij})_{i=1,\ldots,|I|,\ j=1,\ldots,|J|}$ die Matrix der Flüsse zwischen Pixel i und j und f_k der k-te Zeilenvektor. Dann hat f die Form $f = \left(f_1, \ldots, f_{|I|}\right)^T$. Analog wird der Distanzvektor $d = \left(d_1, \ldots, d_{|I|}\right)^T$ definiert. Der Vektor $b = \left(v_1, \ldots, v_{|I|}, w_1, \ldots, w_{|J|}\right)^T$ ist der zusammengesetzte Vektor der Massen von Ziel und Quelle. Für die Matrix $A \in \mathbb{R}^{|I|+|J| \times |I| \cdot |J|}$ ergibt sich eine Blockform. Sei dafür $\mathbb{I}_n$ die n-dimensionale Einheitsmatrix und $\mathbf{1}_n = (1, \ldots, 1)$

ein Zeilenvektor mit n Einsen. Dann lässt sich A schreiben als

$$A = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix}$$

mit

$$A_1 = \begin{bmatrix} \mathbf{1}_{|J|} & 0 & \dots & 0 \\ 0 & \mathbf{1}_{|J|} & \dots & 0 \\ \vdots & & \ddots & \\ 0 & 0 & \dots & \mathbf{1}_{|J|} \end{bmatrix} \text{ und } A_2 = \begin{bmatrix} \mathbb{I}_{|I|} & \dots & \mathbb{I}_{|I|} \end{bmatrix}.$$

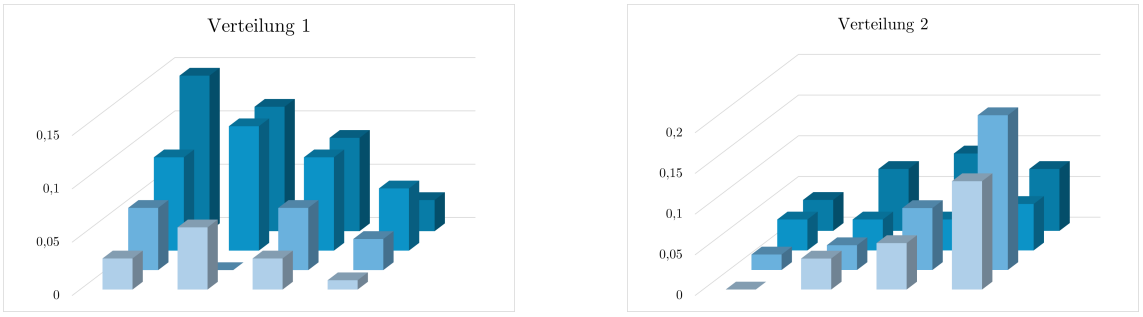
Die Größe der Matrix A hängt somit nur von der Anzahl an Quellen und der Anzahl an Zielen ab.

Die Wasserstein-Metrik berechnet sich nun mithilfe der Lösung des Transportproblems.

Definition 4.2 (Wasserstein-Metrik)**.** Gegeben sei die Lösung des Transportproblems aus Definition 4.1. Dann berechnet sich die Wasserstein-Metrik zwischen zwei Verteilungen von Masse v und w durch

$$\mathrm{WS}(v, w) = \frac{\sum_{i \in I} \sum_{j \in J} c_{ij} f_{ij}}{\sum_{i \in I} \sum_{j \in J} f_{ij}} = \frac{\sum_{i \in I} \sum_{j \in J} c_{ij} f_{ij}}{\sum_{j \in J} w_j}.$$

Es gibt mittlerweile eine verallgemeinerte Darstellung der p-Wassersteinmetrik. Die hier vorgestellte Wasserstein-Metrik entspricht nach dieser Bezeichnung der 1-Wasserstein-Metrik [Kol+17]. Daher wird auf das p im Weiteren verzichtet.



(a) Verteilung 1 (b) Verteilung 2

Abbildung 3.: Darstellung von zwei diskreten zweidimensionalen Verteilungen

Häufig wird die Wasserstein-Metrik direkt für Verteilungen definiert, wie beispielsweise in [Kol+17]. In diesem Fall handelt es sich bei der Menge der Quellen $\mathcal{I}$ und der Menge der Ziele $\mathcal{J}$ um die jeweiligen Trägermengen der Verteilung der Zufallsvariablen X und Y. Die Massen w_i und v_j stellen die Wahrscheinlichkeiten der Merkmalsausprägung dar. Da die Wasserstein-Metrik für Saliency Maps berechnet wird, welche Relevance Scores sich auf n Pixel verteilen, handelt es sich um einen

diskreten Anwendungsfall. Daher wird im Folgenden nur die Definition für diskrete Verteilungen betrachtet. Zur Veranschaulichung der Masse zeigt Abbildung 3 die Darstellung von zwei zweidimensionalen Verteilungen. Für die Wahrscheinlichkeit p_i diskreter Zufallsvariablen gilt $0 \leq p_i \leq 1$ und $\sum_{i \in I} p_i = 1$. Daher folgt für die Berechnung der Wasserstein-Metrik für Verteilungen auch

$$\sum_{i \in I} v_i = \sum_{j \in J} w_j = 1$$

und somit

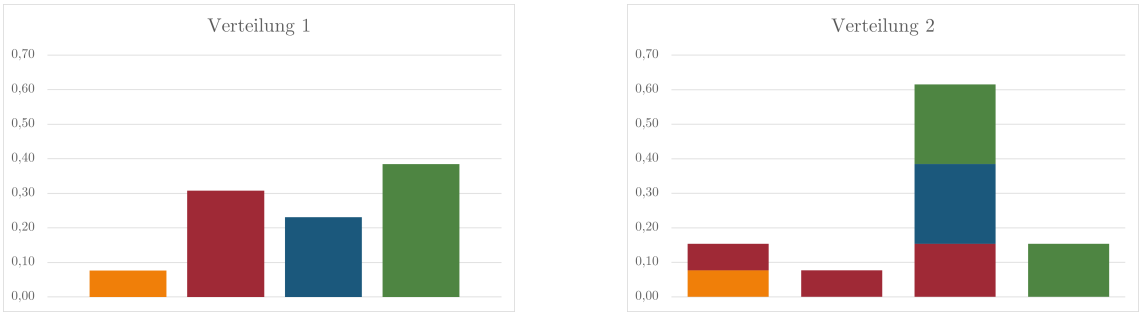
$$\mathrm{WS}(v, w) = \sum_{i \in I} \sum_{j \in J} d_{ij} f_{ij}.$$

Für univariate Verteilungen kann die Wasserstein-Metrik in einer geschlossenen Form berechnet werden [RGC15].

Theorem 4.3 (univariate Wasserstein-Metrik)**.** *Seien $\mathcal{I} \subset \mathbb{R}$ und $\mathcal{J} \subset \mathbb{R}$ endliche Trägermengen und $v_1, \ldots, v_n$ und $w_1, \ldots, w_n$ die Wahrscheinlichkeiten. Die Distanz zwischen $i \in \mathcal{I}$ und $j \in \mathcal{J}$ sei $c_{ij} = |i - j|$. Seien V und W die jeweiligen Verteilungsfunktionen. Die Quantilfunktionen ergeben sich aus der inversen Verteilungsfunktion V^{-1} und W^{-1}. Die Wasserstein-Metrik kann dann berechnet werden als*

$$WS(V, W) = \int_0^1 \left| V^{-1}(t) - W^{-1}(t) \right| \mathrm{d}t.$$

Für den Beweis sei an dieser Stelle auf das Paper [RGC15] verwiesen. Der Vorteil der geschlossenen Darstellung der Wasserstein-Metrik liegt in der schnellen Berechnung, da die Lösung des Optimierungsproblems nicht nötig ist.



(a) Verteilung 1 (b) Verteilung 2

Abbildung 4.: Darstellung des Transports von einer univariaten Verteilung in eine andere

Abbildung 4 zeigt beispielhaft, wie sich die Wahrscheinlichkeiten der univariaten Verteilung 1 auf die Wahrscheinlichkeiten der univariaten Verteilung 2 aufteilen, wenn das Transportproblem gelöst wird. Dabei zeigen die Farben in Verteilung 2 den Ursprung der Masse aus Verteilung 1 an.

Mit der Wasserstein-Metrik werden die Saliency Maps des Originalbildes mit den Saliency Maps seiner Adversarial Examples verglichen. Die Wasserstein-Metrik beschreibt den minimalen Aufwand, um eine Saliency Map in die andere zu transformieren. Eine kleine Metrik bedeutet, dass wenig Masse über geringe Distanzen bewegt werden muss, während eine große Metrik darauf hinweist, dass viel Masse über weite Distanzen bewegt werden muss. Im Folgenden werden zwei verschiedene Varianten für ein Distanzmaß zwischen den Pixeln betrachtet.

Wasserstein-Metrik für Saliency Maps mit euklidischen Distanzen
Seien $v_1, \ldots, v_n \in \mathbb{R}$ die Relevance Scores des Originalbildes und $w_1, \ldots, w_n \in \mathbb{R}$ die Relevance Scores eines zugehörigen Adversarial Examples. Für jeden Relevance Score sei die Position auf der Ebene anhand der Koordinaten der Pixel definiert. Seien $x_1 \ldots x_n \in \mathbb{R}^2$ die ganzzahligen Koordinaten der Pixel des Originalbildes und $y_1 \ldots y_n \in \mathbb{R}^2$ die ganzzahligen Koordinaten im Adversarial Example. Die Pixelkoordinaten bilden somit die Trägermenge für die Verteilung der Relevance Scores. Als Distanzmaß zwischen zwei Pixeln wird die euklidische Distanz der Koordinaten gewählt $d_{ij} = \|x_i - y_j\|_2$.

Im nächsten Schritt müssen die Relevance Scores in eine Verteilung umgewandelt werden. Für die LRP Saliency Maps ist es möglich, dass negative Relevance Scores auftreten. Negative Relevance Scores werden so interpretiert, dass diese Pixel einen Einfluss auf andere Klassen haben [Bac+15]. Daher werden sie für die Auswertung auf Null gesetzt. Nun werden die Relevance Scores beider Saliency Maps skaliert, sodass die Summe der neuen Gewichte jeweils 1 ergibt. Seien $v_i, \ldots, v_n$ die Relevance Scores der Saliency Map des Originalbildes. Dann ergeben $v_i' = \frac{v_i}{\sum_{i \in I} v_i}$ mit $\sum_{i \in I} v_i' = 1$ eine Verteilung der Relevance Scores auf der Saliency Map. Analog werden die Relevance Scores für das Adversarial Example skaliert $w_j' = \frac{w_j}{\sum_{j \in J} w_j}$. Nun kann die zweidimensionale Wasserstein-Metrik für die beiden skalierten Saliency Maps berechnet werden.

Die zweidimensionale Wasserstein-Metrik für zwei Saliency Maps quantifiziert den Aufwand, der erforderlich ist, um die Masse der einen Verteilung so über die Fläche der Saliency Map zu verschieben, dass sie die andere Verteilung annimmt. Dabei wird berücksichtigt, wie weit und wie viel Masse bewegt werden muss. Verschiebt sich jeweils wenig Masse zwischen nahegelegenen Pixeln, so sind die Saliency Maps ähnlich. Verschiebt sich viel Masse über große Distanzen, so sind die Saliency Maps unterschiedlich.

Die Wasserstein-Metrik kann auch als gewichtetes Mittel aller paarweisen Distanzen zwischen Pixeln interpretiert werden, da sie für die skalierten Saliency Maps berechnet wird. Daher kann die Wasserstein-Metrik nur Werte zwischen 0 und dem größtmöglichen Abstand zweier Pixel annehmen. Diese Aussage lässt sich wie folgt beweisen:

Beweis. Die untere Schranke ergibt sich aus der Nichtnegativitätsbedingung des Transportproblems und der, wie oben definierten, skalierten Saliency Maps. Zur Bestimmung der oberen Schranke kann folgende Überlegung herangezogen werden: Die Formel für die Wasserstein-Metrik lässt sich mit vereinfachten Indizes schreiben als $\mathrm{WS}(v, w) = \sum_{k=1}^{K} d_k f_k$, wobei K die Anzahl der Flüsse und Distanzen darstellt und $\sum_{k=1}^{K} f_k = 1$ gilt. Sei nun c_1 die größtmögliche Distanz. Es gilt also $d_1 > d_k$ für alle $k > 1$. Dann lässt sich die Wasserstein-Metrik zerlegen in:

$$\mathrm{WS}(v, w) = \sum_{k=1}^{K} d_k f_k = d_1 f_1 + \sum_{k=2}^{K} d_k f_k < d_1 f_1 + \sum_{k=2}^{K} d_1 f_k = d_1 \sum_{k=1}^{K} f_k = d_1.$$

□

Wasserstein-Metrik für Saliency Maps mit Rangdifferenzen als Distanz Es seien wieder $v_1, \ldots, v_n \in \mathbb{R}$ die Relevance Scores des Originalbildes und $w_1, \ldots, w_n \in \mathbb{R}$ die Relevance Scores eines zugehörigen Adversarial Examples. Wie zuvor werden die negativen Relevance Scores der LRP-Verfahren auf Null gesetzt und die Relevance Scores mit ihrer Summe skaliert.

Es werden nun getrennt für die skalierten Relevance Scores des Originalbildes $v'_1, \ldots, v'_n$ und die skalierten Relevance Scores des Adversarial Examples $w'_1, \ldots, w'_n$ die Ränge gebildet. Seien $r_1, \ldots, r_n$ die Ränge des Originalbildes und $s_1, \ldots, s_n$ die Ränge des Adversarial Examples. Die Ränge definieren somit jeweils eine univariate Trägermenge für die skalierten Relevance Scores. Als Distanz wird die absolute Differenz zwischen den Rängen gewählt $d_{ij} = |r_i - s_j|$. Die Wasserstein-Metrik kann nun nach Theorem 4.3 direkt berechnet werden.

Diese eindimensionale Rang-Wasserstein-Metrik gibt somit Auskunft darüber, wie sich die Relevance Scores unter relevanten Pixeln verschieben. Haben die gleichen Pixel sowohl im Originalbild als auch in der Saliency Map hohe Relevance Scores, so ist diese Wasserstein-Metrik gering. Unterscheiden sich die Ränge der Pixel im Originalbild und im Adversarial Example deutlich, so ist die eindimensionale Wasserstein-Metrik mit Rangdifferenzen groß. Analog zur zweidimensionalen Wasserstein-Metrik nimmt auch die Rang-Wasserstein-Metrik Werte zwischen 0 und der größtmöglichen Distanz an. Diese ist in diesem Fall die größte Rangdistanz.

Für den Vergleich der Wasserstein-Metriken und der Rang-Wasserstein-Metrik zwischen den Verfahren werden die Werte mit der oberen Schranke skaliert. Sie liegen somit zwischen 0 und 1. Die Einzelbetrachtung der Verfahren zeigt die unskalierten Wasserstein-Metriken.

4.4 Sliced Wasserstein-Metrik

Die zweidimensionale Wasserstein-Metrik beruht auf dem Transportproblem. Bei zwei Saliency Maps mit N Pixeln entstehen so N^2 Flüsse. Die Rechenkomplexität aktueller Lösungsalgorithmen für diese linearen Optimierungsprobleme beträgt $\mathcal{O}(N^3 \log N)$ [Kol+17]. Bei 32×32 Pixeln pro Saliency Map entstehen bereits $32^4 = 1.048.576$ Flüsse. Für die univariate Wasserstein-Metrik liegt die Komplexität der Berechnung nur bei $\mathcal{O}(N \log N)$ [Rab+11]. Um die schnelle Berechnung der univariaten Wasserstein-Metrik auszunutzen, wurde von Rabin u. a. [Rab+11] eine neue Metrik eingeführt, die sogenannte sliced Wasserstein-Metrik. In ihrer Veröffentlichung definieren sie diese für ungewichtete mehrdimensionale Punktwolken, verweisen allerdings auf die Erweiterung von gewichteten Punktwolken. Die Punktwolken entsprechen in dieser Arbeit den jeweiligen Trägermengen und die Gewichte sind die Wahrscheinlichkeiten.

Die Idee der sliced Wasserstein-Metrik ist es, eine höherdimensionale Trägermenge (Punktwolke) auf einen Einheitsvektor der Einheitssphäre zu projizieren, wie in Abbildung 5 dargestellt.

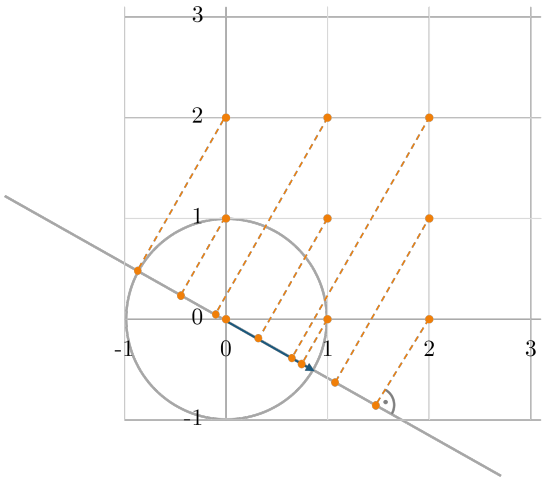


Abbildung 5.: Orthogonalprojektion von Pixelkoordinaten auf einen Einheitsvektor

Sei $S^{d-1} = \left\{\theta \in \mathbb{R}^d \;\middle|\; \|\theta\|_2 = 1\right\} \subset \mathbb{R}^d$ die Einheitssphäre und $\theta \in S^{d-1}$ ein Einheitsvektor. Dann ist $\theta^* : \mathbb{R}^d \to \mathbb{R}$ mit $x \mapsto \langle \theta, x \rangle$ eine Linearform, wobei $\langle \cdot, \cdot \rangle$ das Standardskalarprodukt ist [Nad+21]. Die Pixelkoordinaten werden somit auf ihr Skalarprodukt mit dem Einheitsvektor θ abgebildet und bilden eine neue diskrete und univariate Trägermenge. Nun müssen die Wahrscheinlichkeiten der Pixelkoordinaten in die neue Trägermenge überführt werden. Hierbei wird im Allgemeinen das Bildmaß verwendet. Es wird an dieser Stelle auf die mathematische Definition verzichtet. In der Praxis wird jedem neuen univariaten Träger die Wahrscheinlichkeit der ursprünglichen Pixelkoordinate mitgegeben [Fla+21].

Definition 4.4 (Sliced Wasserstein-Metrik)**.** Sei $S^{d-1} \subset \mathbb{R}^d$ die Einheitssphäre und $\theta \in S^{d-1}$ ein Einheitsvektor. Seien v und w zwei diskrete Verteilungen mit mehrdimensionaler Trägermenge $X \in \mathbb{R}^d$ und $Y \in \mathbb{R}^d$. Seien $X_\theta = \{\langle \theta, s \rangle \mid x \in X\}$ die Projektion der Trägermenge X und Seien $Y_\theta = \{\langle \theta, s \rangle \mid y \in Y\}$ die Projektion der Trägermenge Y auf den Einheitsvektor θ und μ_θ und ν_θ die auf die neuen Trägermengen abgebildeten Verteilungen. Dann berechnet sich die sliced Wasserstein-Metrik durch

$$\mathrm{SWS}(v, w) = \int_{S^{d-1}} \mathrm{WS}(\mu_\theta, \nu_\theta) \mathrm{d}\theta.$$

Die tatsächliche Berechnung der sliced Wasserstein-Metrik erfolgt über eine Approximation [Nad+21]. Sei σ die Uniformverteilung der Einheitsvektoren θ auf S^{d-1}. Seien $\{\theta_l\}, l = 1, \ldots, L$ unabhängig und identisch verteilte Einheitsvektoren aus dieser Verteilung. Dann wird die sliced Wasserstein-Metrik durch

$$\mathrm{SWS}(v, w) = \frac{1}{L} \sum_{l=1}^{L} \mathrm{WS}(\mu_{\theta_l}, \nu_{\theta_l})$$

approximiert.

4.5 Weitere Auswertungen

Statistische Kennzahlen Für jede Saliency Map werden das Maximum, der Mittelwert und die Varianz über die Relevance Scores berechnet. Für die Varianz wird dafür die Formel $s^2 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n}$ verwendet.

Erklärungsquote Für jede Saliency Map wird die Summe der Relevance Scores als Gesamterklärbarkeit definiert. Die Erklärungsquote zeigt dann, wie viele Pixel minimal nötig sind, um 90 % der Gesamterklärbarkeit zu erreichen. Für die minimale Pixelanzahl werden die Relevance Scores absteigend sortiert und kumuliert, bis mindestens 90 % der Gesamterklärbarkeit erreicht sind. Die Erklärungsquote misst, ob sich die Relevanz eher gleichmäßig auf viele Pixel verteilt, oder auf wenige Pixel konzentriert. Die Anzahl der relevanten Pixel wird als Quote in Bezug auf die Gesamtanzahl der Pixel angegeben.

Totale Variation Die totale Variation ist ein Maß für das Rauschen in einem Bild [ROF92]. Für die Saliency Maps wird sie mit folgender Formel berechnet. Seien $v_{i,j}$ die Relevance Scores des Pixels mit den Koordinaten (i, j). Dann ist die totale Variation gegeben durch

$$t = \sum_{i,j} \sqrt{|v_{i+1,j} - v_{i,j}|^2 + |v_{i,j+1} - v_{i,j}|^2}.$$

Große totale Variationen deuten auf ein großes Rauschen hin, also große Unterschiede in den Relvance Scors benachbarter Pixel. Sehr kleine Werte deuten darauf hin, dass benachbarte Pixel ähnliche Relevance Scores haben.

5 Implementierung

In diesem Kapitel werden die zentralen Implementierungen beschrieben. Die Umsetzung baut auf bestehenden Skripten und Funktionen von Dieter und Zisgen [DZ23b] auf. Da dieser Code bereits in Python verfasst wurde und unter anderem die Bibliothek TensorFlow [Aba+15] Anwendung findet, wurde auch der neue Code in Python geschrieben. Um eine Modularität und Erweiterbarkeit zu erreichen und Funktionen sinnvoll zu gruppieren, wurde ein objektorientierter Ansatz gewählt.

5.1 Convolutional Neural Network

In dieser Arbeit wird das CNN von Dieter und Zisgen [DZ23a; Die20] verwendet. Es ist darauf trainiert, die Bilder des CIFAR10-Datensatzes zu klassifizieren. Dieser beinhaltet 50.000 Bilder als Trainingsdaten und 10.000 Bilder als Testdaten. Alle Bilder haben eine Auflösung von 32×32 Pixeln und werden in die 10 Klassen Airplane, Automobile, Bird, Cat, Deer, Dog, Frog, Horse, Ship und Truck eingeteilt [Kri09]. Das CNN ist mit TensorFlow implementiert. Es ist bereits trainiert und hat eine Trainings-Accuracy von 90.87 % und eine Test-Accuracy von 89.02 %. Tabelle 2 zeigt einen Überblick über die Architektur des CNN [DZ23a].

5.2 Datengrundlage

Zusätzlich zum originalen CIFAR10-Datensatz [Kri09] wurden in dieser Arbeit die bereits erstellten Adversarial Examples aus dem Git-Repository [DZ23b] zum Paper [DZ23a] verwendet. Diese wurden im Ordner "Adversarial_Examples" bereitgestellt, wobei die 85403 Bilder unterteilt in 10 Target-Ordner abgespeichert waren. Anhand der Bild-ID und der Zielklasse aus dem Dateinamen wurden 8005 Originalbilder identifiziert. Die Bilder wurden mit den Originalbildern aus dem CIFAR10-Datensatz verglichen. Sie stimmten alle überein. Es gab außerdem 897 Adversarial Examples, deren Originalbild nicht in der Liste enthalten ist. Diese wurden aus dem Originaldatensatz nachgeladen und verarbeitet. Es waren somit insgesamt 8902 Originalbilder und 77398 Adversarial Examples. Damit umfasst der verwendete Datensatz vier Adversarial Examples weniger als in [DZ23a].

Nr.	Layertyp	Output Dim.	Trainierbare Parameter	
			Anz. Biases	Anz. Weights
1	Convolution2D	(32,32,32)	32	864
2	BatchNormalization	(32,32,32)	0	64
3	Activation (ReLU)	(32,32,32)	0	0
4	Convolution2D	(32,32,32)	32	9216
5	BatchNormalization	(32,32,32)	0	64
6	Activation (ReLU)	(32,32,32)	0	0
7	MaxPooling2D	(16,16,23)	0	0
8	Dropout	(16,16,23)	0	0
9	Convolution2D	(16,16,64)	64	18432
10	BatchNormalization	(16,16,64)	0	128
11	Activation (ReLU)	(16,16,64)	0	0
12	Convolution2D	(16,16,64)	64	36864
13	BatchNormalization	(16,16,64)	0	128
14	Activation (ReLU)	(16,16,64)	0	0
15	MaxPooling2D	(8,8,64)	0	0
16	Dropout	(8,8,64)	0	0
17	Convolution2D	(8,8,128)	128	73728
18	BatchNormalization	(8,8,128)	0	256
19	Activation (ReLU)	(8,8,128)	0	0
20	Convolution2D	(8,8,128)	128	147456
21	BatchNormalization	(8,8,128)	0	256
22	Activation (ReLU)	(8,8,128)	0	0
23	MaxPooling2D	(4,4,128)	0	0
24	Dropout	(4,4,128)	0	0
25	Flatten	(,2048)	0	0
26	Dense	(,10)	10	20480
27	Activation (Softmax)	(,10)	0	0

Tabelle 2.: Spezifikation des CNN nach [DZ23a]

Zur Verarbeitung der Bilddaten innerhalb der Python-Anwendung wurde eine Klasse namens *Image* definiert, welche die Repräsentation eines Bildes darstellt. Ebenso wurden zum Laden und Speichern der *Image*-Objekte im JSON-Format entsprechende Klassen und Methoden implementiert. Insbesondere aufgrund der rechenintensiven Funktionen für die Erstellung mehrerer Saliency Maps je Bild und der statistischen Auswertungen war es notwendig, eine zuverlässige Speicher- und Ladefunktion zu gewährleisten.

5.3 Saliency Maps

Für die Saliency Map-Methoden Vanilla Gradient und Grad-CAM wurden eigene Implementierungen geschrieben. Die Saliency Maps für die LRP-Verfahren wurden mit dem Code aus dem Repository [DZ23b] bestimmt. Jedes Verfahren wurde als separate Klasse implementiert.

Vanilla Gradient Für die Bestimmung des Gradienten der Score-Funktion wurde die Klasse *GradientTape* der Bibliothek Tensorflow [Aba+15] verwendet. Sie bietet eine Implementierung der automatischen Differenzierung, um effizient Gradienten einer Funktion zu berechnen.

Grad-CAM Als letzter Convolution Layer wird der Layer Nummer 20 aus Tabelle 2 gewählt. Der Gradient des Scores der gewünschten Klasse bis zum genannten Layer wird mittels der *GradientTape*-Klasse aus der Bibliothek Tensorflow [Aba+15] ermittelt. Anschließend wird die Saliency Map, wie in Kapitel 3.3.2 beschrieben, berechnet.

Für die Darstellung der Saliency Maps wurden die Relevance Scores innerhalb einer Saliency Map jeweils auf Werte zwischen $[-1, 1]$ skaliert.

Insgesamt wurden für jedes Bild fünf Saliency Maps erstellt: Vanilla Gradient, Grad-CAM, LRP-$\alpha\beta$ mit $\alpha = 1$ und $\alpha = 2$ und LRP-ε mit $\varepsilon = 1$. Da die Erstellung aller Saliency Maps rechenaufwändig ist, wurde hierfür ein Skript mit der Pythonstandardbibliothek Multiprocessing erstellt.

5.4 Statistische Methoden

Alle statistischen Metriken wurden jeweils für die positiven Saliency Maps berechnet. Daher wurde in jeder Metrik eine Funktion implementiert, die negative Relevance Scores auf Null setzt. Die Berechnungen der statistischen Methoden wurden in einer Klasse mit dem Namen *MetricService* konsolidiert, um Modularität zu gewährleisten.

Spearman-Rangkorrelationskoeffizient Die Formel für den Spearman-Rangkorrelationskoeffizienten wurde in einer Methode der Klasse *MetricService* gemäß Kapitel 4.1 implementiert.

Relevance Ranking Das Relevance Ranking von [DZ23b] wurde an die neue Datenstruktur angepasst und wie in Kapitel 4.2 in der Klasse *MetricService* implementiert. Die Methode liefert als Ergebnis die Rangdifferenz des relevantesten Pixels des Originalbildes sowie zwei Listen mit den IDs der Top-Pixel beider Saliency Maps.

Wasserstein-Metrik Die Wasserstein-Metrik entsteht durch die Lösung des Transportproblems zwischen zwei Saliency Maps. Dabei handelt es sich um ein lineares Optimierungsproblem. Da das Lösen von linearen Optimierungsproblemen einen gewissen Rechenaufwand benötigt, wie in Kapitel 4.4 beschrieben, wurden hier verschiedene Optimierungen vorgenommen.

Alle paarweisen euklidischen Distanzen zwischen den Pixelkoordinaten von zwei Saliency Maps wurden einmalig berechnet und abgespeichert. Die Nebenbedingungen des Transportproblems können mithilfe einer Matrix A festgelegt werden. Wie in Kapitel 4.3 beschrieben, sind diese nur abhängig von der Größe der Saliency Maps und wurden für 32×32 Pixel und 8×8 Pixel vordefiniert.

Eine weitere Reduktion der Komplexität des Optimierungsproblems kann für die LRP-Verfahren erzielt werden. Da in dieser Arbeit ausschließlich die positiven Relevance Scores der LRP-Verfahren betrachtet werden, entstehen hier viele Relevance Scores mit dem Wert Null. Für die Flüsse von einem Pixel k mit dem Relevance Score 0 im Originalbild zu den Pixeln im Adversarial Example gilt daher $\sum_{j \in J} f_{kj} = 0$ und durch die Nichtnegativitätsbedingung folgt $f_{kj} = 0$ für alle $j \in J$. Analog wird keine Masse zu Pixeln im Adversarial Example verschoben, die einen Relevance Score von 0 haben. Diese Pixel wurden daher aus der Menge der Quellen und Ziele für das Transportproblem herausgenommen. Damit verringern sich die Anzahl der gesuchten Werte und die Anzahl der Nebenbedingungen.

Für das Lösen von linearen Optimierungsproblemen gibt es ein großes Angebot von Solvern. Für diese Arbeit wurden die Solver Gurobi [GUR], Mosek [MOS] und Coin [COI] evaluiert. Bei manuellen Tests mit einem kleinen Datensatz erwies sich Gurobi mit der eigenen Python-Bibliothek gurobipy [GUR] als der effizienteste Solver für den in dieser Arbeit beschriebenen Anwendungsfall. Trotz der deutlich besseren Performance durch den Solver Gurobi musste zusätzlich auf leistungsstarke Hardware und den erneuten Einsatz von Multiprocessing zurückgegriffen werden. So konnte eine Laufzeitreduktion von geschätzten 2 Jahren auf mehrere Stunden erzielt werden.

Die eindimensionale Wasserstein-Metrik mit den Rangdistanzen wurde mit der Funktion *wasserstein_distance* der Python-Bibliothek SciPy [Vir+20] berechnet.

Sliced Wasserstein-Metrik Für die sliced Wasserstein-Metrik wurde keine eigene Implementierung vorgenommen, sondern die Funktion *sliced_wasserstein_distance* der Python-Bibliothek POT [Fla+21] verwendet.

6 Ergebnisse

In diesem Kapitel werden die Ergebnisse der Saliency Map-Methoden und der statistischen Kennzahlen dargestellt. Dafür werden erst alle Erklärmodelle einzeln betrachtet und anschließend miteinander verglichen.

6.1 Grad-CAM

Das Grad-CAM-Verfahren erzeugt aufgrund der Betrachtung des letzten Convolution Layers im hier genutzten CNN, Saliency Maps der Größe 8×8. Alle statistischen Auswertungen werden auf Basis dieser 64 Relevance Scores erstellt.

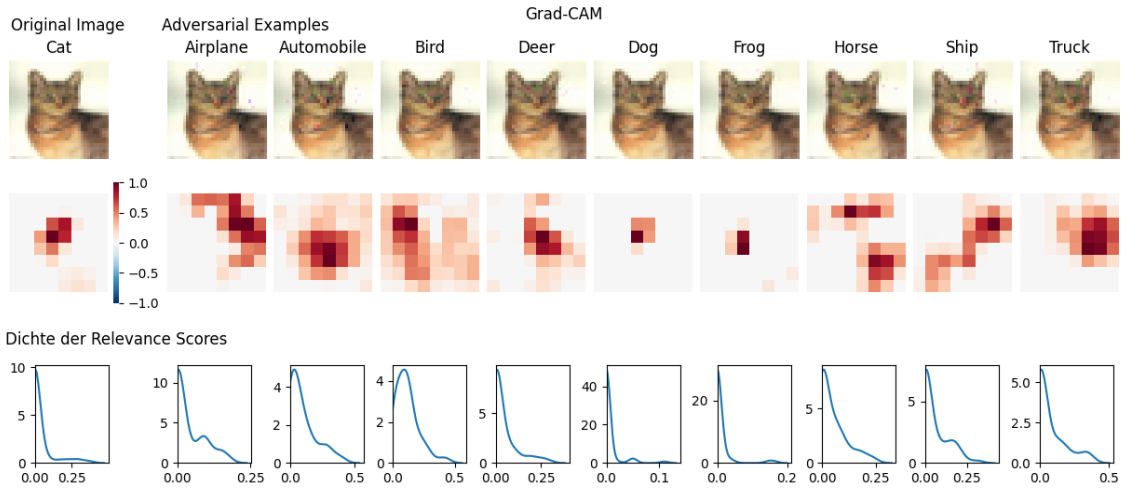


Abbildung 6.: Darstellung der Saliency Maps des Grad-CAM-Verfahrens für das Beispielbild Katze. Die erste Zeile zeigt das Originalbild sowie die Adversarial Examples. Die zweite Zeile zeigt die Relevance Scores als Heatmap. Die dritte Zeile zeigt die Dichte der Relevance Scores in der Saliency Map.

Es konnten für 548 (0,7 %) Adversarial Examples und 960 (10,8 %) Originalbilder keine Saliency Map ungleich 0 erstellt werden. Die vergleichenden Kennzahlen Spearman-Rangkorrelationskoeffizient, Wasserstein-Metrik mit euklidischen Distanzen, sliced Wasserstein-Metrik, Wasserstein-Metrik basierend auf den Rangdistanzen und das Relevance Ranking können nicht berechnet werden, wenn entweder das

Adversarial Example oder das Originalbild eine Null-Saliency Map hat. Insgesamt konnten für 68722 von 77398 Adversarial Examples die vergleichenden Metriken berechnet werden. Die Einzelmetriken wurden für 76850 Adversarial Examples und 7942 Originalbilder berechnet.

Ein Beispiel der Saliency Maps für ein Originalbild und dessen Adversarial Examples für das Grad-CAM-Verfahren zeigt Abbildung 6. Im Folgenden wird das Bild auch Katzenbild genannt. Die Visualisierung der Saliency Maps zeigt die unterschiedliche Auflösung von Bild und Saliency Map. Durch die Berechnung der Saliency Map mit den Feature Maps des letzten Convolution Layers, beträgt die Größe der Saliency Maps 8×8 Pixel. Dennoch können die Relevance Scores auf relevante Bereiche im Bild übertragen werden. Es fällt auf, dass die verschiedenen Saliency Maps durchaus zu verschiedenen relevanten Regionen führen. Für die Visualisierung der Saliency Maps werden die Relevance Scores skaliert, sodass diese für jedes Bild zwischen 0 und 1 liegen. Die Verteilungen der unskalierten Relevance Scores sind in der dritten Zeile von Abbildung 6 abgebildet.

Label	Relevance Scores				
	Mittelwert	Maximum	Varianz	Totale Variation	Erklärungs-quote
Airplane	0,035289	0,169089	0,002600	1,998191	0,296875
Automobile	0,094856	0,398856	0,010884	4,810187	0,500000
Bird	0,115131	0,437926	0,008700	5,122794	0,640625
Cat	**0,031378**	**0,364932**	**0,006039**	**3,085499**	**0,156250**
Deer	0,040219	0,290970	0,004675	3,110494	0,296875
Dog	0,004406	0,110168	0,000296	0,601797	0,062500
Frog	0,006877	0,162775	0,000762	0,914108	0,078125
Horse	0,046494	0,240957	0,003959	3,375329	0,375000
Ship	0,053739	0,313477	0,005849	3,651630	0,328125
Truck	0,069358	0,365315	0,011053	3,412356	0,296875

Tabelle 3.: Metriken des Katzenbildes für Grad-CAM. Das Originalbild ist in fetter Schrift hervorgehoben.

Verschiedene Kennzahlen dieser Verteilungen werden in Tabelle 3 zusammengefasst. Der Durchschnitt der Relevance Scores einer Saliency Map für die Bilder liegt in diesem Beispiel zwischen 0,0044 (Adversarial Example für die Klasse Dog) und 0,1151 (Adversarial Example für die Klasse Bird). Der höchste Relevance Score in jeder Saliency Map liegt zwischen 0,110 und 0,438. Die Varianz liegt zwischen 0,0003 und 0,011. Die totale Variation ist in der Saliency Map der Adversarials für die Klasse Bird mit 5,123 am größten und beim Adversarial Example der Klasse Dog mit 0,602 am kleinsten.

Der Durchschnitt, das Maximum, die Varianz und die totale Variation wurden von allen Saliency Maps berechnet. Somit können die Verteilungen der Werte betrachtet und die Unterschiede zwischen Originalbildern und Adversarial Examples analysiert werden. Abbildung 7 veranschaulicht die Boxplots dieser vier Metriken. In allen Boxplots dieser Arbeit zeigt die Box dabei die Quartile der Verteilungen. Die Whisker sind auf das 1,5-fache des Interquartilsabstands beschränkt.

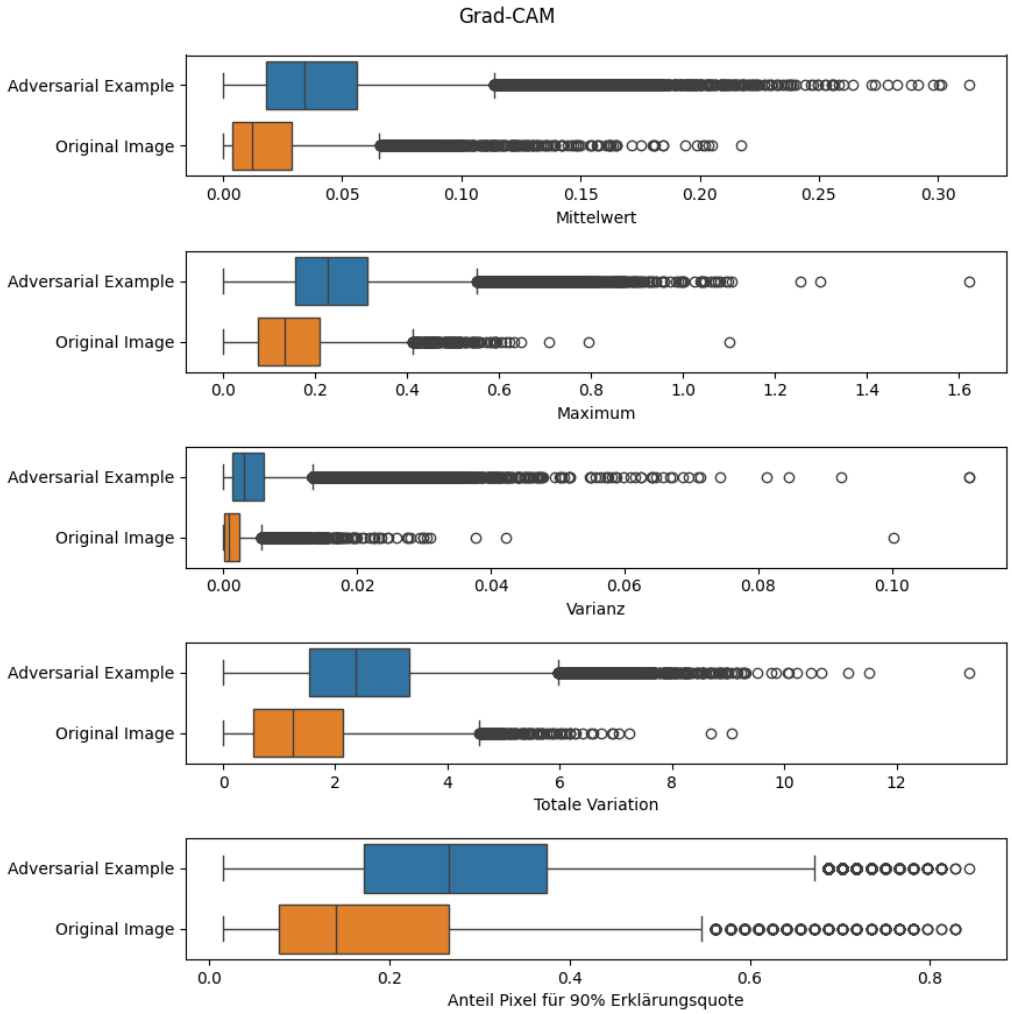


Abbildung 7.: Vergleich der Standardmetriken für Grad-CAM Saliency Maps zwischen Originalbildern und Adversarial Examples

Es fällt auf, dass für alle Metriken die Werte für die Originalbilder geringer sind als für die Adversarial Examples. Daraus lässt sich schlussfolgern, dass die Pixel der Originalbilder tendenziell geringere Relevance Scores aufweisen und diese weniger streuen. Darüber hinaus zeigt die Auswertung, dass Originalbilder weniger Pixel benötigen, um eine 90-prozentige Erklärbarkeit zu erreichen. Vergleiche der einzelnen Klassen befinden sich im Anhang.

Die verschiedenen Verteilungen der Lageparameter der Saliency Maps zeigen Unterschiede zwischen den beiden Gruppen Originalbilder und Adversarial Example.

Relevant ist, ob sich ein einzelnes Originalbild von seinen Adversarial Examples unterscheidet. Die Darstellung der Grad-CAM Saliency Maps für das Katzenbild in Abbildung 6 deutet bereits darauf hin. Um diese Unterschiede zu quantifizieren, zeigt Tabelle 4 verschiedene Metriken. Alle Metriken wurden für jedes Paar aus Adversarial Example und Originalbild berechnet. Anschließend werden die Quantile daraus ermittelt. Abbildung 8 zeigt die Verteilungen der Vergleichsmetriken und Tabelle 4 die Quantile.

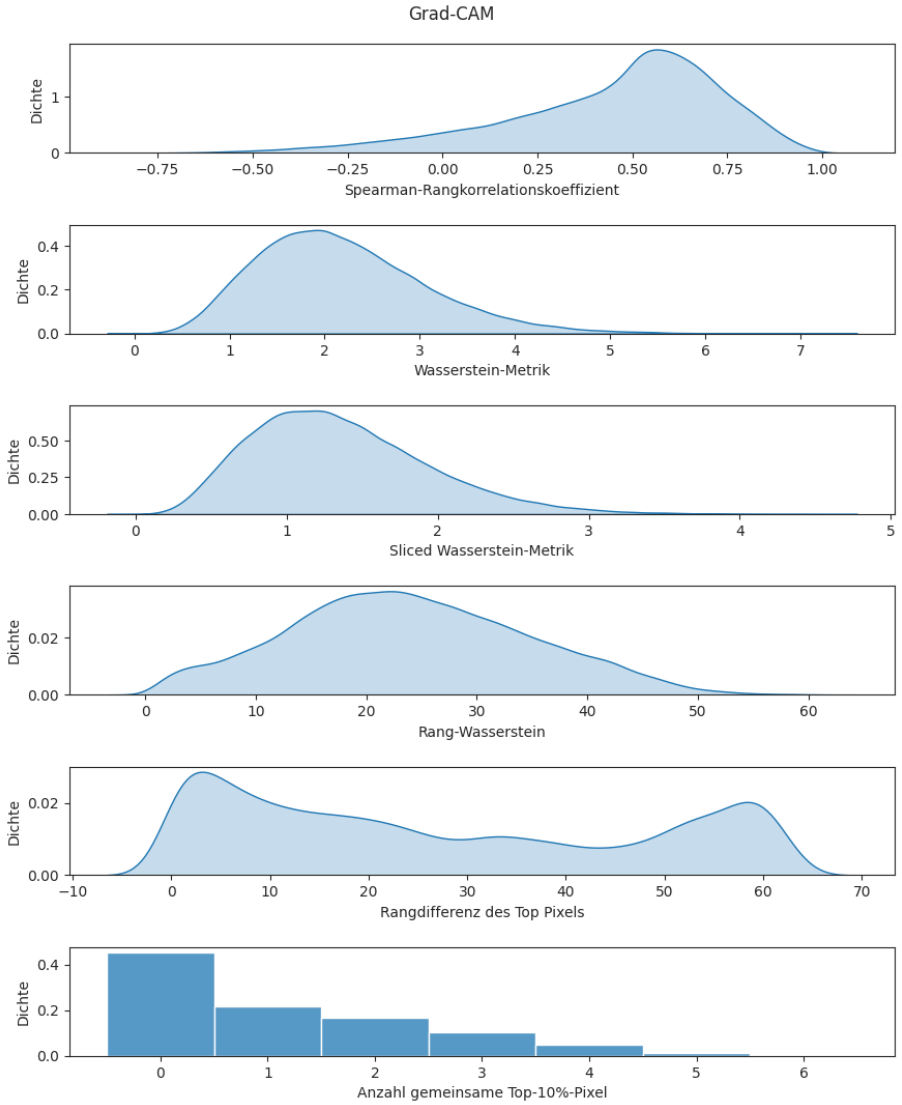


Abbildung 8.: Verteilungen der Vergleichsmetriken des Grad-CAM-Verfahrens

Der Spearman-Rangkorrelationskoeffizient ist eine Metrik, die aufzeigt, ob ein monotoner Zusammenhang zwischen den Pixelrängen besteht. Für das Grad-CAM-Verfahren zeigt sich deutlich, dass der Korrelationskoeffizient bereits im 0,1-Quantil positiv ist. Daraus lässt sich schlussfolgern, dass ein positiver Zusammenhang zwischen den Rängen besteht. Sind die Ränge der Pixel im Originalbild hoch, so sind die Ränge auch im Adversarial Example hoch. Der Median dieser Verteilung liegt bei

Metrik	Quantile					
	0,10	0,25	0,50	0,75	0,95	1,00
Spearman Kor.	0,0267	0,2885	0,5128	0,6509	0,8225	1,0
Wasserstein	1,1318	1,5358	2,0716	2,7121	3,821	7,3092
Sliced WS	0,6655	0,9262	1,2816	1,7093	2,4411	4,5867
Rang-Wasserstein	9,978	16,2404	23,3334	31,3538	42,5533	60,9391
Rangdifferenz	2	8	22	48	60	63

Tabelle 4.: Quantile der vergleichenden Metriken für Grad-CAM

0,5128, was bedeutet, dass über 50 % der Adversarial Examples mindestens einen mittleren positiven Zusammenhang der Ränge aufweisen. Da das 0,95-Quantil den Wert 0,8225 erreicht, haben nur etwa 5 % der Adversarial Examples eine starke positive Korrelation zwischen ihren Rängen und den Rängen des Originalbildes.

Für die Wasserstein-Metrik gibt es keine Einteilung in Effektstärken. Allerdings kann die Wasserstein-Metrik für das Grad-CAM-Verfahren maximal den Wert 9,90 annehmen. Der Median der Wasserstein-Metrik beträgt 2,0716 und das 0,95-Quantil 3,821 und nimmt daher einen eher niedrigen Wert an. Der Median der sliced Wasserstein-Metrik ist 1,2816. Die sliced Wasserstein-Metrik ergibt somit nicht die gleichen Werte wie die Wasserstein-Metrik.

Um den linearen Zusammenhang zwischen der Wasserstein-Metrik und der sliced Wasserstein-Metrik zu bewerten wurde der Person Korrelationskoeffizient für die Metriken über alle Saliency Map-Verfahren berechnet [Fah+23]. Dieser beträgt 0,9975 und ist mit einem p-Wert von 0,0 hoch signifikant. Die sliced Wasserstein-Metrik ist somit eine sehr gute Alternative zur regulären Wasserstein-Metrik. Zur Bewertung der Saliency Maps wird in dieser Arbeit dennoch die normale Wasserstein-Metrik verwendet, da bei dieser die obere Schranke bekannt ist.

Die Rang-Wasserstein-Metrik zeigt, wie sich die Masse zwischen den Rängen verschiebt. Die Metrik kann für das Grad-CAM-Verfahren zwischen 0 und 63 liegen. Mit einem Median von 22 liegt die Verteilung in der unteren Hälfte. Abbildung 9 zeigt beispielhaft, wie die Verteilung der Relevance Scores des Originalbildes auf die Verteilung der Relevance Scores eines Adversarial Examples verschoben wird. Dabei sind die Relevance Scores in beiden Grafiken nach dem Rang des Pixels im Originalbild sortiert, damit die Verschiebung deutlich wird.

Die letzte Metrik betrachtet die Rangdifferenz des Pixels, welcher im Originalbild den größten Relevance Score hat. Hier fällt auf, dass es sowohl viele Adversarial Examples gibt, die eine große Verschiebung erzeugen, als auch zahlreiche Adversarial Examples, die nur eine kleine bis mittlere Verschiebung verursachen. Nur bei 21,49 % der Adversarial Examples ist der Top-Pixel des Originalbildes auch unter den Top-10 %-Pixeln des Adversarial Examples.

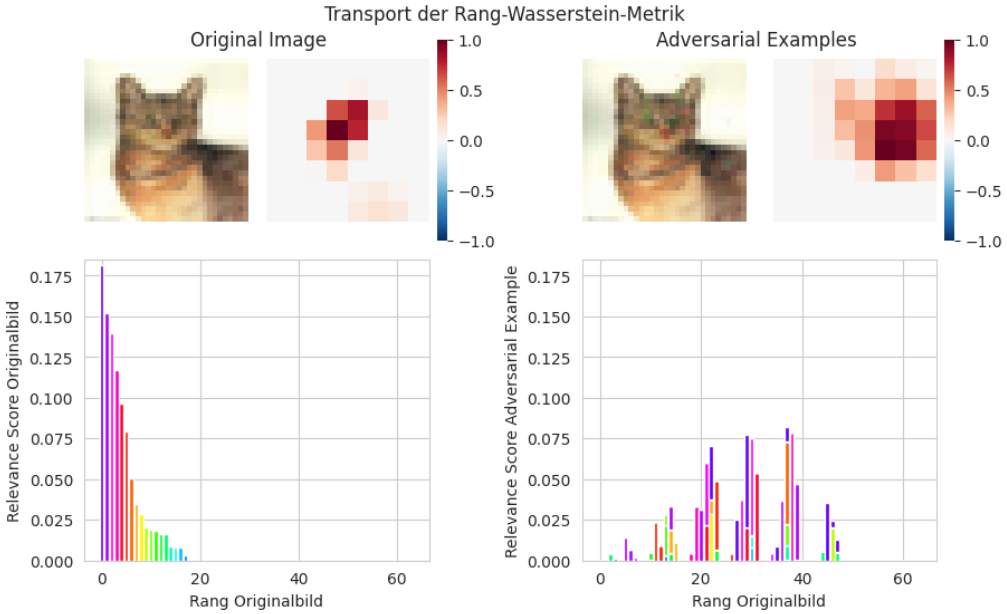


Abbildung 9.: Darstellung des Transports bei der Rang-Wasserstein-Metrik. Die x-Achse zeigt in beiden Fällen die Ränge des Originalbildes. Die y-Achse zeigt die Relevance Scores. Die Farben zeigen die Quellpixel im Originalbild.

6.2 Vanilla Gradient

Das Vanilla Gradient-Verfahren hat für alle Originalbilder und alle Adversarial Examples eine Saliency Map erstellt, ohne dass dabei Null-Saliency Maps entstanden sind. Daher werden die Metriken für alle 77398 Adversarial Examples und 8902 Originalbilder erstellt. Die Saliency Maps des Vanilla Gradient-Verfahrens haben die Größe 32×32 und weisen jedem Pixel einen Relevance Score zu.

Die Abbildung 10 zeigt das Katzenbild mit seinen Adversarial Examples und den entsprechenden Vanilla Gradient Saliency Maps. Die Relevance Scores sind in der Heatmap auf das Intervall von 0 bis 1 skaliert. Die dritte Zeile der Abbildung zeigt die Verteilung der unskalierten Relevance Scores in jedem Bild. Auffällig ist, dass die relevanten Pixel in jeder Saliency Map des Beispiels weit verstreut sind, sodass sich die Gesamtrelevanz auf viele Pixel verteilt. Daher erscheinen die Saliency Maps sehr verrauscht und es ist für Menschen kein direktes Muster erkennbar. Dennoch wird anhand des Beispielbildes deutlich, dass sich die Saliency Maps durchaus voneinander unterscheiden.

Tabelle 5 zeigt die Lageparameter, die zu den Vanilla Gradient Saliency Maps des Katzenbildes gehören. Der jeweilige Durchschnitt der Relevance Scores liegt zwischen 0,736 für das Adversarial Example der Klasse Dog und 1,262 für das Adversarial Example Ship. Das Originalbild weist mit einem Maximum von 5,814 den größten

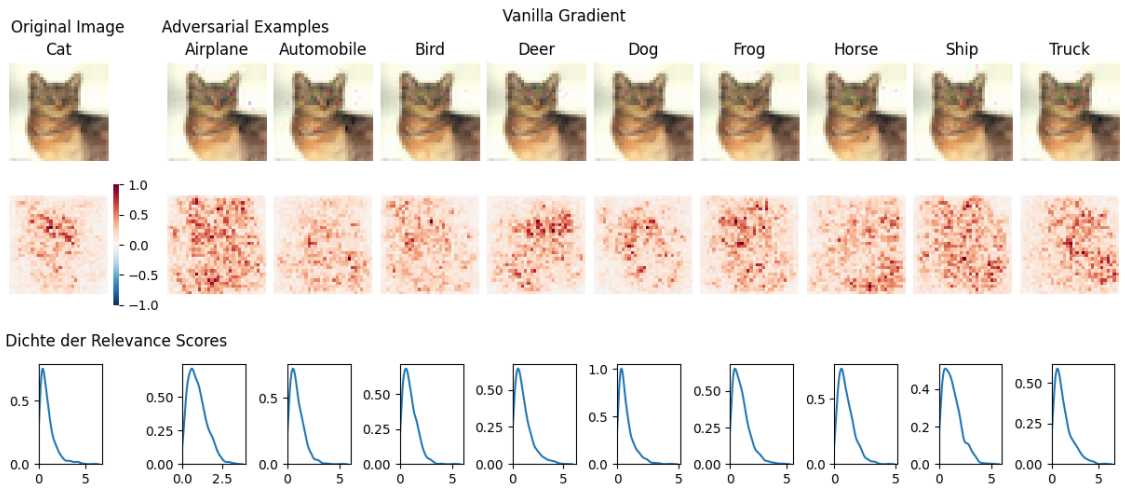


Abbildung 10.: Darstellung der Vanilla Gradient Saliency Maps für das Beispielbild Katze. Die erste Zeile zeigt das Originalbild sowie die Adversarial Examples. Die zweite Zeile zeigt die Relevance Scores als Heatmap. Die dritte Zeile zeigt die Dichte der Relevance Scores in der Saliency Map.

Relevance Score für einen Pixel auf, während das Adversarial Example für die Klasse Airplane den kleinsten Maximalwert von 3,359 erreicht.

Die Varianzen der Relevance Scores innerhalb der Saliency Maps liegen zwischen 0,345 und 0,848. Die totale Variation nimmt große Werte an, die zwischen 727,071 und 1031,931 liegen. Dies unterstützt den visuellen Eindruck des starken Rauschens in den Saliency Maps.

Der Anteil der benötigten Pixel, um 90 % der Gesamterklärung zu erzielen, liegt zwischen 0,668 und 0,7119. Daraus ergibt sich, dass viele Pixel benötigt werden, um eine hohe Erklärbarkeit zu erzielen und sich die Relevance Scores auf viele Pixel verteilen

Um allgemeine Aussagen über das Verfahren treffen zu können, werden die Mittelwerte, Maxima, Varianzen, totalen Variationen und Erklärungsquoten für alle Saliency Maps betrachtet und in Abbildung 11 als Boxplot dargestellt. Die Verteilungen der Mittelwerte und Maxima weisen darauf hin, dass für das Vanilla Gradient-Verfahren die Relevance Scores der Originalbilder größer sind als die Relevance Scores der Adversarial Examples. Darüber hinaus sind die Varianzen und die totalen Variationen der Relevance Scores der Originalbilder größer als die der Adversarial Examples. Dies deutet darauf hin, dass das Rauschen in den Saliency Maps der Originalbilder stärker ausgeprägt ist, als in den Adversarial Examples. Nur der Anteil der benötigten Pixel für 90 % Erklärbarkeit ist bei den Adversarial Examples höher als bei den Originalbildern. In beiden Fällen sind die meisten Werte sehr hoch, jedoch werden bei den Adversarial Examples tendenziell mehr Pixel benötigt.

Label	Relevance Scores				
	Mittelwert	Maximum	Varianz	Totale Variation	Erklärungs-quote
Airplane	0,945671	3,359080	0,345176	740,474456	0,711914
Automobile	0,952394	5,234673	0,431672	727,070510	0,701172
Bird	0,986936	5,183595	0,439455	777,021840	0,700195
Cat	**0,941978**	**5,814285**	**0,658516**	**769,121287**	**0,667969**
Deer	1,082025	5,262777	0,714691	850,230683	0,679688
Dog	0,735789	4,424508	0,378965	608,772467	0,673828
Frog	1,017813	4,713233	0,554354	816,493047	0,684570
Horse	0,975199	4,390486	0,413765	743,050300	0,711914
Ship	1,262215	5,027019	0,691501	1021,931903	0,698242
Truck	1,227322	5,647303	0,848082	1007,109080	0,687500

Tabelle 5.: Metriken des Katzenbildes für Vanilla Gradient. Das Originalbild ist in fetter Schrift hervorgehoben.

Die Verteilungen der Lageparameter der Saliency Maps deutet auf Unterschiede zwischen den Originalbildern und den Adversarial Examples hin. Die vergleichenden Metriken der Vanilla Gradient Saliency Maps sind in Abbildung 12 dargestellt. Die dazugehörigen Quantile sind in der Tabelle 6 enthalten.

Metrik	Quantile					
	0,10	0,25	0,50	0,75	0,95	1,00
Spearman Kor.	0,2329	0,3029	0,3813	0,4579	0,5643	0,8185
Wasserstein	1,3121	1,6327	2,1198	2,7910	4,0858	9,2827
Sliced WS	0,6977	0,9270	1,2734	1,7395	2,6194	5,9457
Rang-WS	106,9409	121,8761	138,5898	156,4257	185,8026	304,8011
Rangdifferenz	18	68	198	419	779	1023

Tabelle 6.: Quantile der vergleichenden Metriken für Vanilla Gradient

Der Spearman-Rangkorrelationskoeffizient zeigt hauptsächlich positive Werte, da das 0,1-Quantil den Wert 0,233 hat. Der 0,95-Quantil-Wert liegt bei 0,564, was darauf hinweist, dass die Korrelation zwischen den Rängen im Originalbild und im Adversarial Example in fast 95 % der Fälle nur schwach ist. Dies deutet auf Unterschiede zwischen den Saliency Maps hin.

Die Wasserstein-Metrik ist mit einem Median von 2,12 und einem 0,95-Quantil von 4,09 hingegen relativ gering, da die obere Schranke bei 43,84 liegt. Dies deutet auf sehr ähnliche Saliency Maps hin. Die niedrigen Wasserstein-Werte sind aufgrund der Tatsache erklärbar, dass die Saliency Maps ein hohes Maß an Rauschen aufweisen.

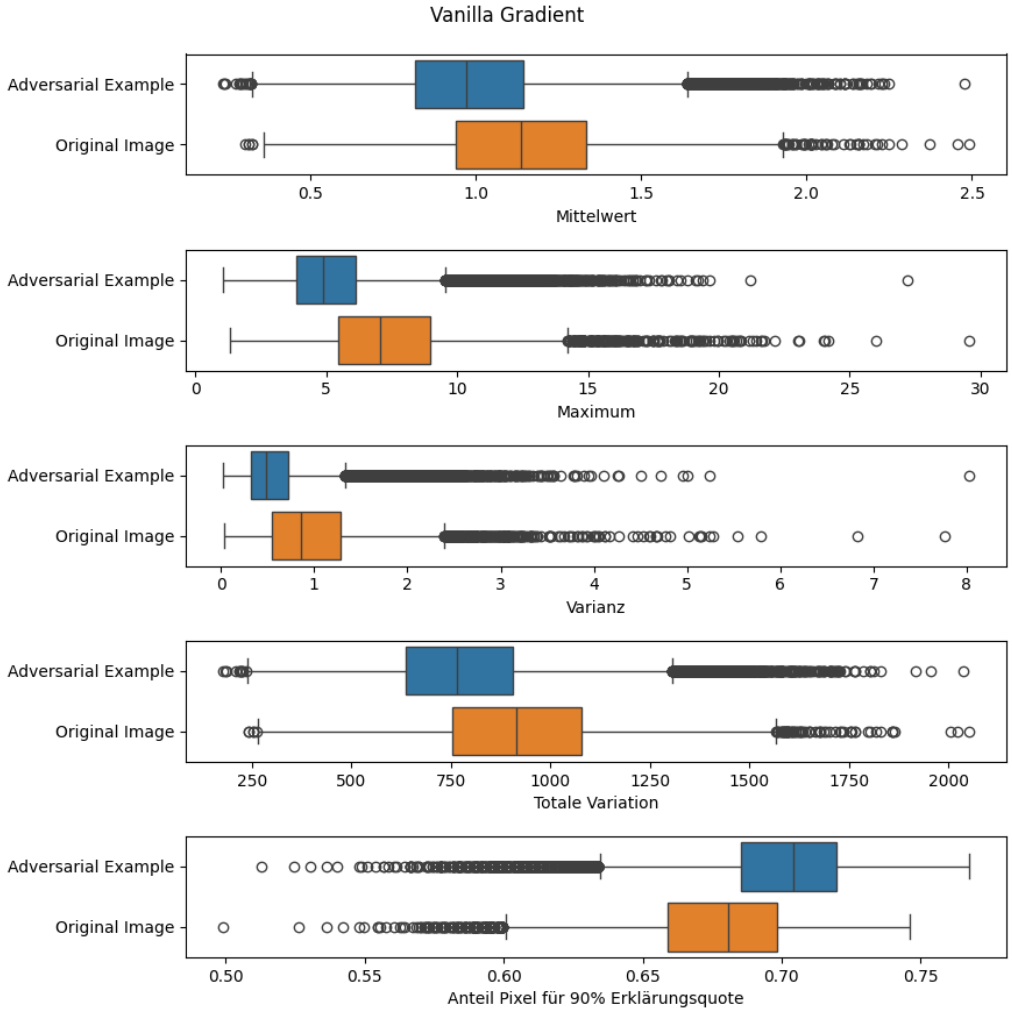


Abbildung 11.: Vergleich der Standardmetriken für Vanilla Gradient Saliency Maps zwischen Originalbildern und Adversarial Examples

Aufgrund der Verteilung der Relevance Scores auf viele Pixel, sowohl im Originalbild als auch im Adversarial Example, wird Masse häufig zu nahegelegenen Pixeln verschoben. Somit sind die Distanzen in der Lösung des Transportproblems klein und damit auch die Wasserstein-Metrik.

Die Wasserstein-Metrik mit Rangdistanzen kann für die Vanilla Gradient Saliency Maps maximal den Wert 1023 erreichen. Mit einem Median von 138,59 sind diese Wasserstein-Metriken eher gering. Dennoch zeigt das 0,1-Quantil mit 106,94, dass immer eine gewisse Veränderung der Masse unter den Rängen stattfindet. Verdeutlicht wird dies durch die Rangdifferenzen. In 25 % der Adversarial Examples sind die Rangdifferenzen kleiner als 68, durch das 0,75-Quantil von 419 haben aber auch 25% der Adversarial Examples große Differenzen. Bei 32,81 % der Adversarial Examples ist der Top-Pixel des Originalbildes auch unter den Top-10 %-Pixeln des Adversarial Example. Bei den meisten Adversarial Examples findet daher eine größere Verschiebung beim relevanten Pixel statt.

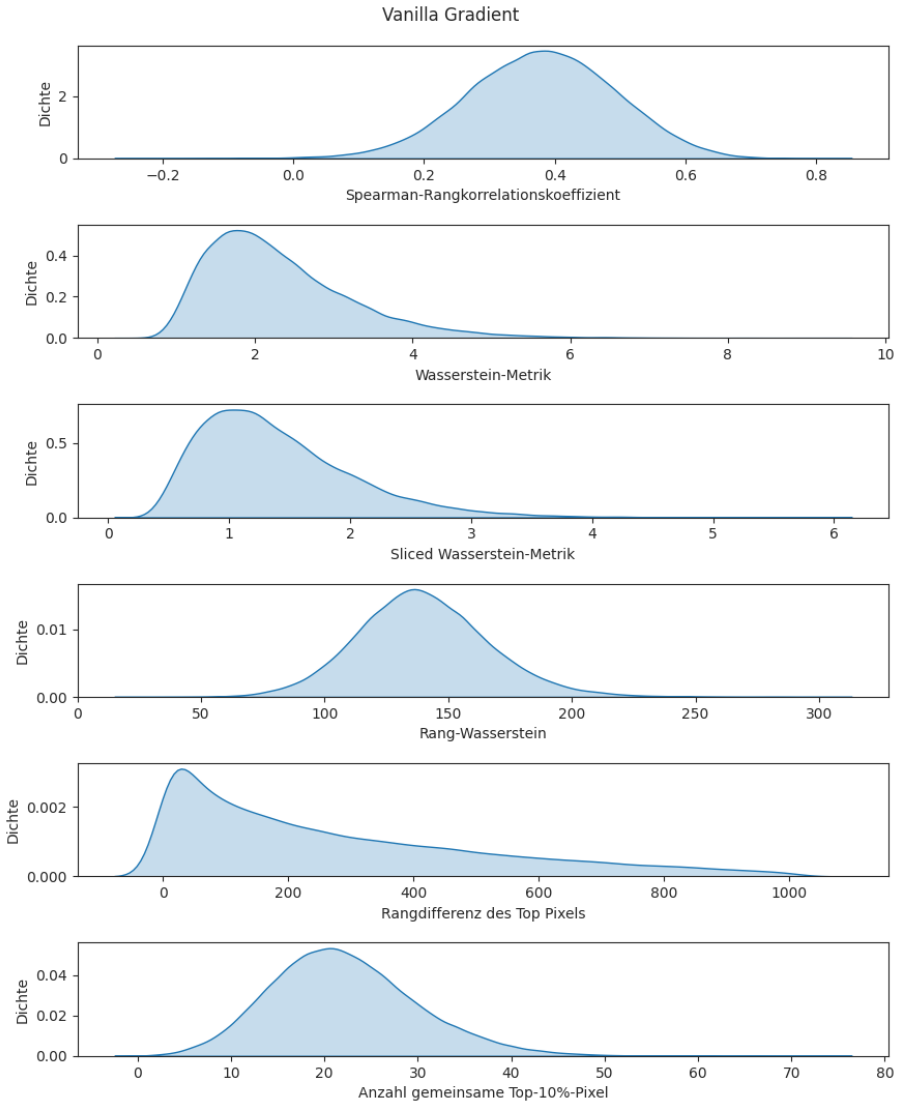


Abbildung 12.: Verteilungen der Vergleichsmetriken des Vanilla Gradient-Verfahrens

6.3 Layer-wise Relevance Propagation

Die LRP-Verfahren liefern auch negative Werte in den Saliency Maps. In den Abbildungen 13, 16 und 19 werden die Saliency Maps, inklusive der negativen Werte, mit der Farbe Blau dargestellt. Für die Berechnung der verschiedenen Kennzahlen wurden die negativen Werte auf Null gesetzt.

6.3.1 $\alpha\beta$-Regel mit $\alpha = 1$

Das LRP-Verfahren mit der $\alpha\beta$-Regel und $\alpha = 1$ hat für alle Originalbilder und alle Adversarial Examples die Saliency Map berechnet.

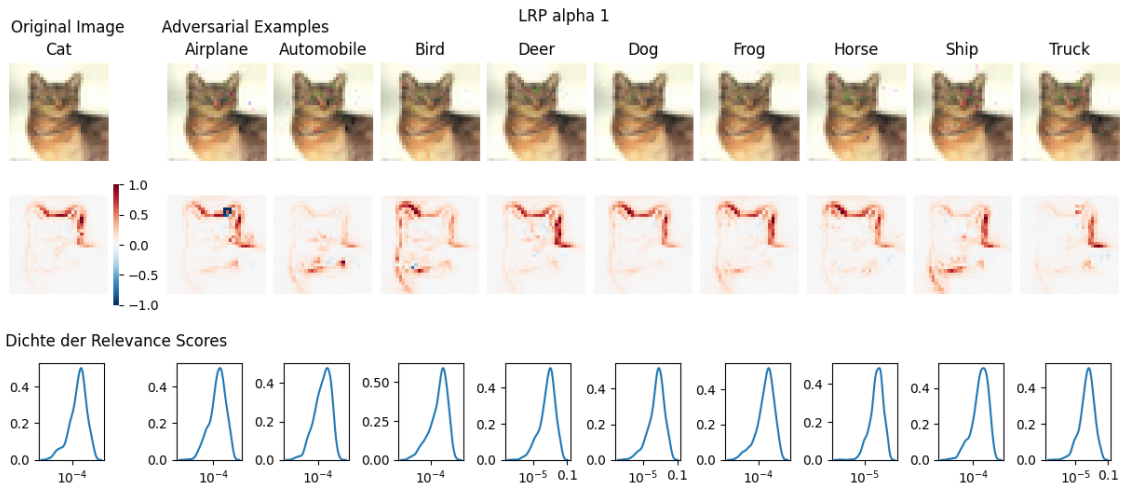


Abbildung 13.: Darstellung der Saliency Maps mit LRP und $\alpha = 1$ für das Beispielbild Katze. Die erste Zeile zeigt das Originalbild sowie die Adversarial Examples. Die zweite Zeile zeigt die Relevance Scores als Heatmap. Die dritte Zeile zeigt die Dichte der Relevance Scores in der Saliency Map. Die x-Achse ist logarithmiert.

Bei der Betrachtung der Saliency Maps fällt auf, dass in diesem Beispiel in fast allen Saliency Maps die relevanten Pixel eine Umrandung des Katzenkopfes darstellen. Für das menschliche Auge sind zwar geringfügige Unterschiede erkennbar, dennoch ähneln sich die Saliency Maps der verschiedenen Adversarial Examples sehr. Auffällig ist, dass das LRP-Verfahren relativ wenige Pixel als relevant markiert. Die dritte Zeile der Abbildung 13 zeigt die Verteilung der Relevance Scores in einer Saliency Map. Hierbei ist die x-Achse logarithmiert dargestellt, damit diese schiefe Verteilung überhaupt dargestellt werden kann.

Tabelle 7 zeigt die statistischen Kennzahlen der Saliency Maps. Das Originalbild weist den größten Mittelwert der Relevance Scores mit 0,0016 sowie den größten Maximalwert mit 0,0289 auf. Den kleinsten Mittelwert mit 0,00079 sowie den kleinsten Maximalwert mit 0,0117 hat das Adversarial Example der Klasse Airplane. Hinzu kommen noch sehr kleine Varianzen. Daran erkennt man auch die fokussierte Verteilung. Es gibt viele Pixel mit einem sehr kleinen Wert und somit wenige Pixel mit einem größeren Wert. Dies wird auch durch die Erklärungsquote unterstützt. 90 % der Erklärbarkeit verteilen sich in dem gezeigten Originalbild auf nur 28,5 % der Pixel. Das Adversarial Example der Klasse Bird hat mit 35,5 % die größte Erklärungsquote.

Abbildung 14 zeigt die Verteilungen der statistischen Kennzahlen der Saliency Maps. Analog zu den vorherigen Verfahren werden die x-Achsen der Verteilungen der Mittelwerte, der Maxima, der Varianzen und der totalen Variation mit logarithmierter x-Achse angezeigt. Die Boxplots für den Mittelwert zeigen, dass die Mehrzahl der Pixel in den Saliency Maps des LRP mit $\alpha = 1$ zumeist sehr kleine Relevance Scores erhalten und die Boxplots für die Erklärungsquote zeigen, dass nur wenige

Label	Relevance Scores				
	Mittelwert	Maximum	Varianz	Totale Variation	Erklärungs-quote
Airplane	0,000790	0,011722	0,000002	0,875160	0,303711
Automobile	0,000820	0,016793	0,000002	0,896972	0,328125
Bird	0,001130	0,013321	0,000003	1,195662	0,355469
Cat	**0,001606**	**0,028907**	**0,000013**	**1,792778**	**0,285156**
Deer	0,001473	0,020462	0,000009	1,546699	0,310547
Dog	0,001244	0,020267	0,000007	1,305044	0,306641
Frog	0,001465	0,020708	0,000008	1,495562	0,322266
Horse	0,001250	0,018541	0,000006	1,300630	0,316406
Ship	0,001284	0,017440	0,000005	1,298483	0,337891
Truck	0,000812	0,018228	0,000004	0,870795	0,294922

Tabelle 7.: Metriken des Katzenbildes für LRP mit $\alpha = 1$. Das Originalbild ist in fetter Schrift hervorgehoben.

Pixel als relevant markiert werden. Die Boxplots deuten darauf hin, dass es Unterschiede zwischen den Kennzahlenverteilungen von Originalbildern und Adversarial Examples gibt. So sind beispielsweise die Interquartilsabstände der Mittelwerte, der Maxima, der Varianzen und der totalen Variationen bei den Adversarial Examples geringer als bei den Originalbildern. Nur bei der Erklärungsquote sind keine relevanten Unterschiede ersichtlich.

Metrik	Quantile					
	0,10	0,25	0,50	0,75	0,95	1,00
Spearman Kor.	0,6707	0,7517	0,8155	0,8623	0,9114	0,9849
Wasserstein	1,5386	2,2670	3,5059	5,3332	9,2151	20,7231
Sliced WS	0,9030	1,3748	2,1799	3,3642	5,8572	13,6500
Rang-WS	19,4473	35,0065	63,1869	111,2284	268,8566	866,2490
Rangdifferenz	0	1	9	51	701	1023

Tabelle 8.: Quantile der vergleichenden Metriken für LRP mit $\alpha = 1$

Für die weitere Evaluierung zeigt die Abbildung 15 die Verteilungen der vergleichenden Metriken. In Tabelle 8 werden die zugehörigen Quantile dargestellt. Besonders auffällig ist der hohe Spearman-Rangkorrelationskoeffizient, der bereits im 0,1-Quantil einen Wert von 0,67 aufweist. Das bedeutet, dass annähernd alle Adversarial Examples mindestens einen mittleren positiven monotonen Zusammenhang mit dem Originalbild haben. Mit einem Median von 0,82 haben sogar über 50 % der Adversarial Examples einen starken Zusammenhang. Pixel mit hohen Rängen im Originalbild belegen somit auch hohe Ränge im Adversarial Example. Dies wird

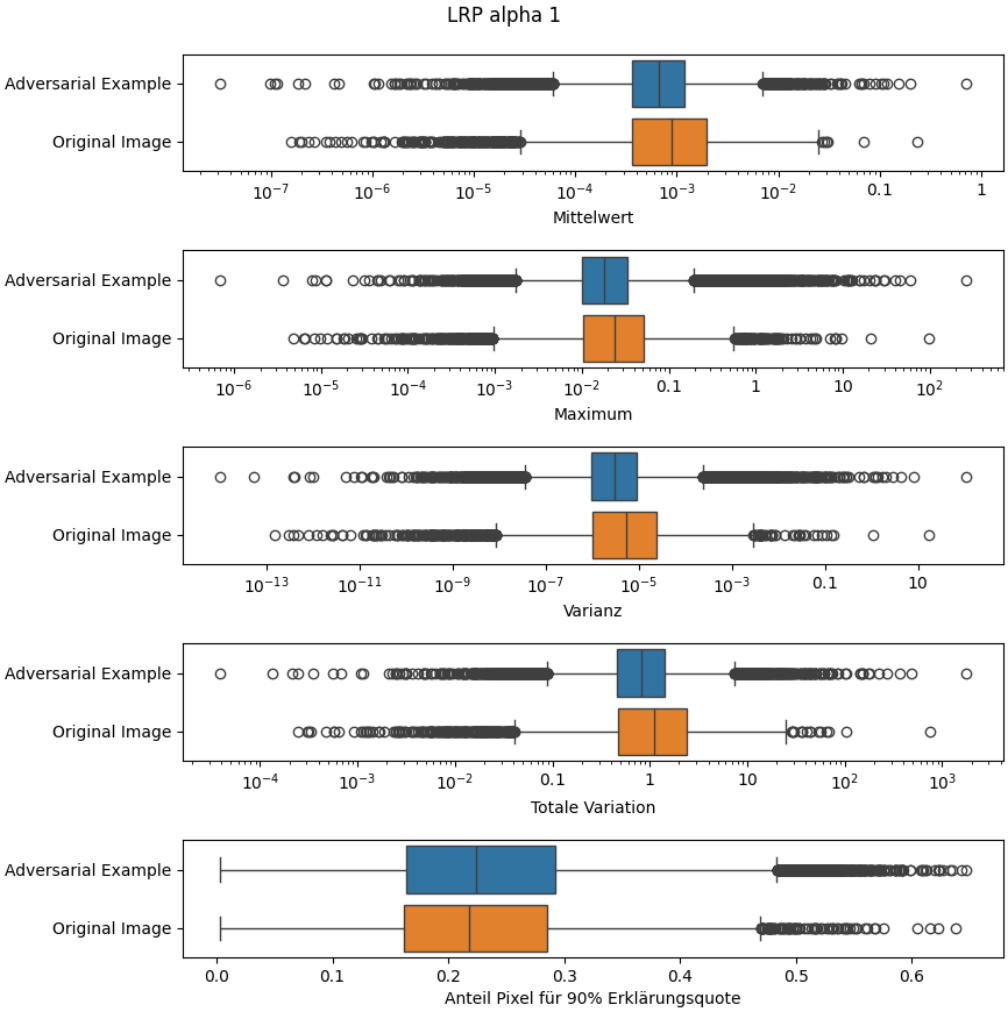


Abbildung 14.: Vergleich der Standardmetriken für Saliency Maps des LRP-Verfahrens mit $\alpha = 1$ zwischen Originalbildern und Adversarial Examples

zusätzlich unterstützt durch die Verteilung der Rangdifferenz des Top-Pixels. In der Hälfte der Adversarial Examples ist der Top-Pixel des Originalbildes nur 9 Ränge niedriger. Bei 82,35 % der Adversarial Examples ist der Top-Pixel des Originalbildes auch unter den Top-10 %-Pixeln des Adversarial Examples. Ein analoges Bild ergibt die Wasserstein-Metrik mit Rangdistanzen. Diese könnte bei den Saliency Maps höchstens 1023 annehmen. Mit einem Median von 63,19 und einem 0,95-Quantil von 268,85 sind sie niedrig. Auffällig ist, dass es Ausreißer nach oben gibt. Der Maximalwert der Rang-Wasserstein-Metrik über alle Adversarial Examples beträgt 866,25. Die Wasserstein-Metrik hat mit einem Median von 2,12 und einem Maximum von 9,28 sehr geringe Werte. Sie kann maximal den Wert 43,84 annehmen. Somit deuten alle Metriken darauf hin, dass es keinen großen Unterschied zwischen einer Saliency Map des Originalbildes und der Saliency Map eines Adversarial Examples gibt.

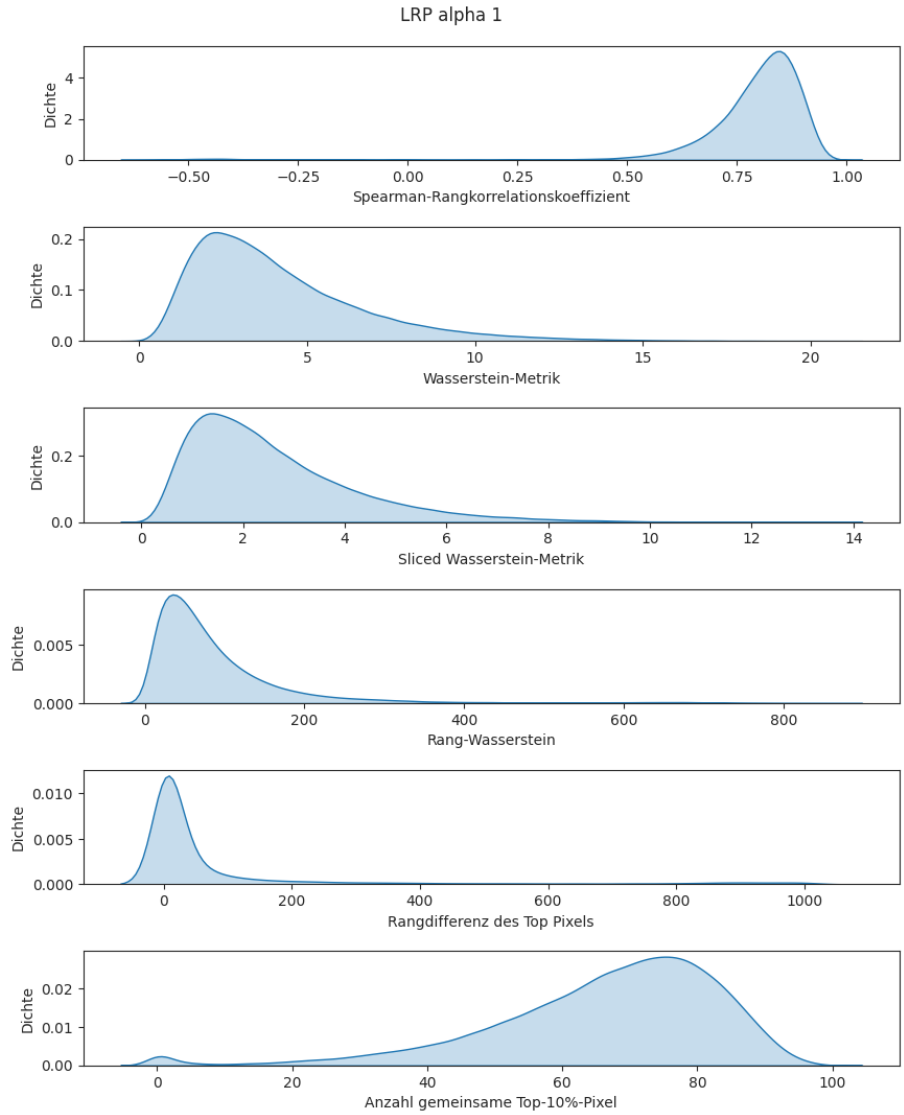


Abbildung 15.: Verteilungen der Vergleichsmetriken des LRP-Verfahrens mit $\alpha = 1$

6.3.2 $\alpha\beta$-Regel mit $\alpha = 2$

Die $\alpha\beta$-Regel mit dem Parameter $\alpha = 2$ hat für alle Originalbilder und alle Adversarial Examples eine Saliency Map erstellt. Abbildung 16 zeigt das Katzenbild mit allen Adversarial Examples und den zugehörigen Saliency Maps sowie die Verteilung der Relevance Scores in jeder Saliency Map. Auch hier ist die x-Achse logarithmiert.

Die visuelle Darstellung der Saliency Maps als Heatmap und die Verteilungen zeigen, dass das LRP-Verfahren für $\alpha = 2$ noch weniger Pixeln einen relativ hohen Relevance Score zuweist, als bei $\alpha = 1$. Es werden noch weniger Pixel als relevant gekennzeichnet. Im gezeigten Beispielbild erscheinen manche Saliency Maps bei der Betrachtung sehr ähnlich zum Originalbild, wie beispielsweise die Klassen Dog, Frog und Horse. Die anderen Saliency Maps weisen größere Unterschiede auf.

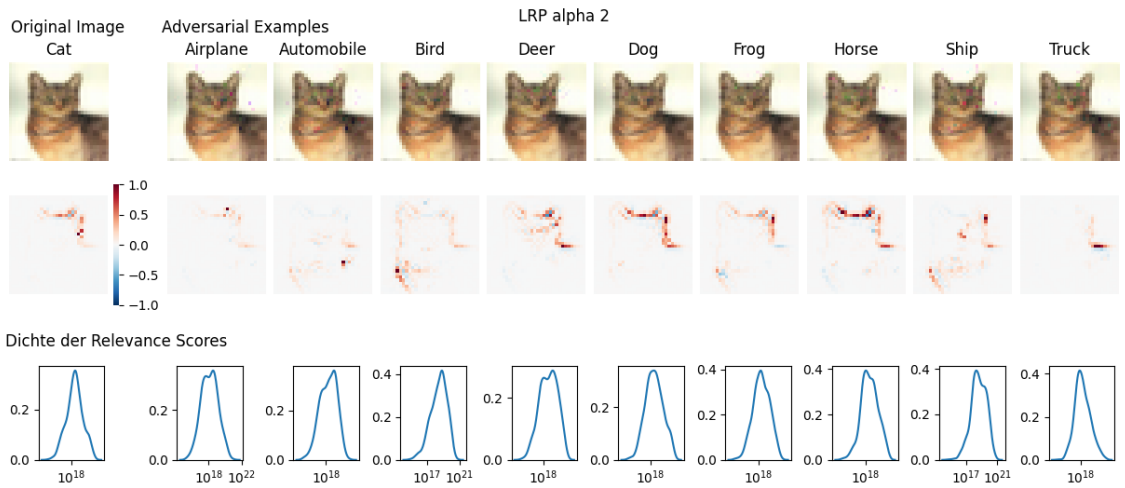


Abbildung 16.: Darstellung der Saliency Maps mit LRP und $\alpha = 2$ für das Beispielbild Katze. Die erste Zeile zeigt das Originalbild sowie die Adversarial Examples. Die zweite Zeile zeigt die Relevance Scores als Heatmap. Die dritte Zeile zeigt die Dichte der Relevance Scores in der Saliency Map. Die x-Achse ist logarithmiert.

Label	Relevance Scores				
	Mittelwert	Maximum	Varianz	Totale Variation	Erklärungsquote
Airplane	$7{,}021 \cdot 10^{18}$	$8{,}071 \cdot 10^{20}$	$1{,}230 \cdot 10^{39}$	$1{,}439 \cdot 10^{22}$	0,119
Automobile	$7{,}130 \cdot 10^{18}$	$5{,}942 \cdot 10^{20}$	$7{,}763 \cdot 10^{38}$	$1{,}485 \cdot 10^{22}$	0,158
Bird	$9{,}557 \cdot 10^{18}$	$3{,}915 \cdot 10^{20}$	$6{,}625 \cdot 10^{38}$	$1{,}801 \cdot 10^{22}$	0,207
Cat	$\mathbf{4{,}438} \cdot 10^{19}$	$\mathbf{3{,}145} \cdot 10^{21}$	$\mathbf{4{,}281} \cdot 10^{40}$	$\mathbf{9{,}277} \cdot 10^{22}$	**0,072**
Deer	$2{,}874 \cdot 10^{19}$	$1{,}177 \cdot 10^{21}$	$9{,}998 \cdot 10^{39}$	$5{,}895 \cdot 10^{22}$	0,120
Dog	$2{,}121 \cdot 10^{19}$	$7{,}621 \cdot 10^{20}$	$6{,}063 \cdot 10^{39}$	$3{,}940 \cdot 10^{22}$	0,105
Frog	$9{,}564 \cdot 10^{18}$	$4{,}562 \cdot 10^{20}$	$1{,}015 \cdot 10^{39}$	$1{,}792 \cdot 10^{22}$	0,158
Horse	$1{,}340 \cdot 10^{19}$	$4{,}246 \cdot 10^{20}$	$2{,}084 \cdot 10^{39}$	$2{,}624 \cdot 10^{22}$	0,135
Ship	$1{,}403 \cdot 10^{19}$	$5{,}985 \cdot 10^{20}$	$1{,}864 \cdot 10^{39}$	$2{,}554 \cdot 10^{22}$	0,166
Truck	$1{,}502 \cdot 10^{19}$	$1{,}637 \cdot 10^{21}$	$9{,}664 \cdot 10^{39}$	$2{,}720 \cdot 10^{22}$	0,065

Tabelle 9.: Metriken des Katzenbildes für LRP mit $\alpha = 2$. Das Originalbild ist in fetter Schrift hervorgehoben.

Tabelle 9 zeigt verschiedene Lageparameter der Saliency Maps. Dabei fällt auf, dass sehr hohe Relevance Scores vergeben werden. Der größte Relevance Score ist mit $3{,}15 \cdot 10^{21}$ im Originalbild zu finden. Diese hohen Relevance Scores führen auch zu sehr großen Mittelwerten, welche zwischen $7{,}02 \cdot 10^{18}$ für das Adversarial Example der Klasse Airplane und $4{,}44 \cdot 10^{19}$ für das Originalbild liegen. Allerdings zeigt die Erklärungsquote, welche im Beispiel zwischen 6,54 % und 20,70 % liegt, dass sich

diese sehr hohen Relevance Scores auf sehr wenige Pixel verteilen. Die Mehrzahl der Pixel ist damit relativ gesehen nicht relevant für die Erklärung der Klasse.

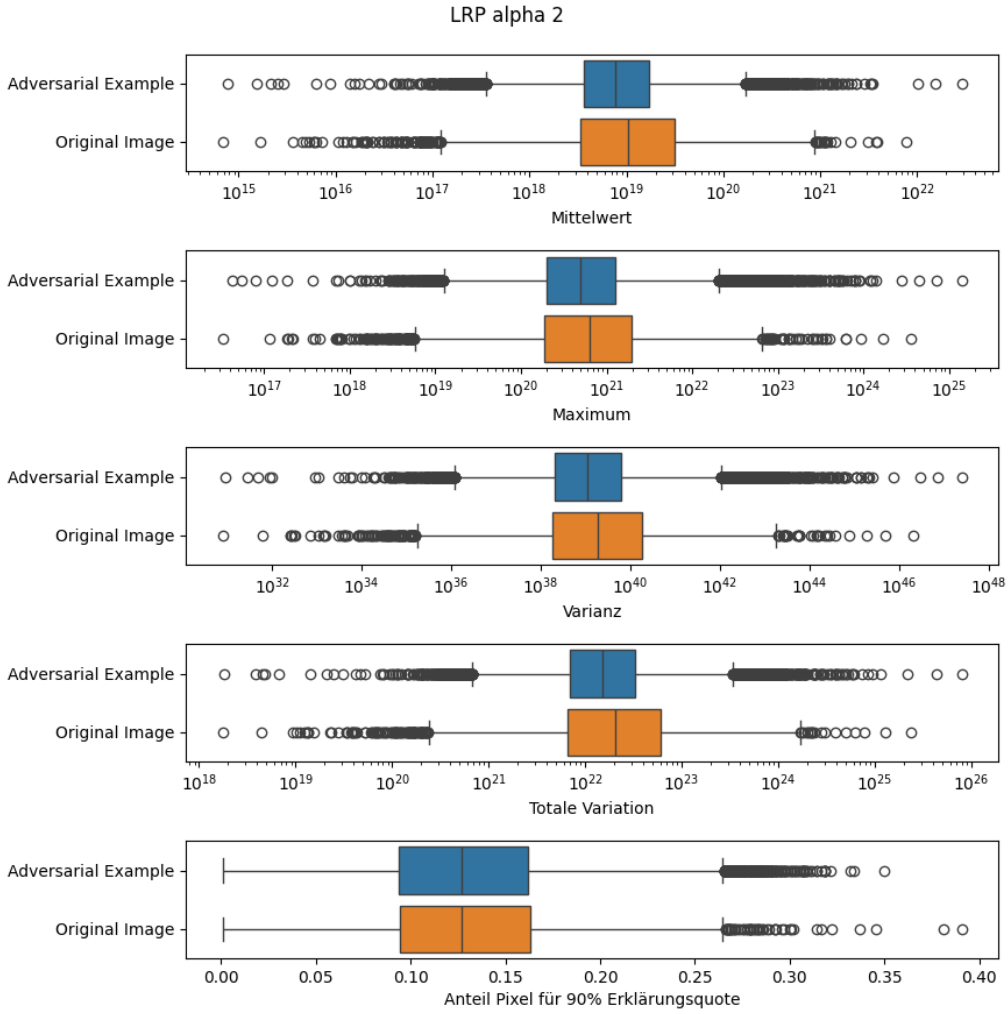


Abbildung 17.: Vergleich der Standardmetriken für Saliency Maps des LRP-Verfahrens mit $\alpha = 2$ zwischen Originalbildern und Adversarial Examples

Abbildung 17 zeigt die Verteilungen der Lageparameter für die Originalbilder und die Adversarial Examples als Boxplots. Für die bessere Darstellung ist die x-Achse für die Mittelwerte, die Maxima, die Varianzen und die totalen Variationen logarithmiert. Die Boxplots zeigen gewisse Unterschiede in den Verteilungen. Insbesondere der Interquartilsabstand ist bei den Adversarial Examples geringer. Nur die Erklärungsquoten unterscheiden sich zwischen den beiden Gruppen nicht.

Die vergleichenden Metriken werden in Abbildung 18 als Verteilung und in Tabelle 10 mit den Quantilen dargestellt. Hierbei fällt der eher niedrige Spearman-Rangkorrelationskoeffizient auf. Mit einem 0,75-Quantil von 0,46 haben über 75 % der Adversarial Examples nur eine geringe positive Korrelation bei den Rängen zum Originalbild. Somit haben Pixel mit einem hohen Rang im Originalbild nicht

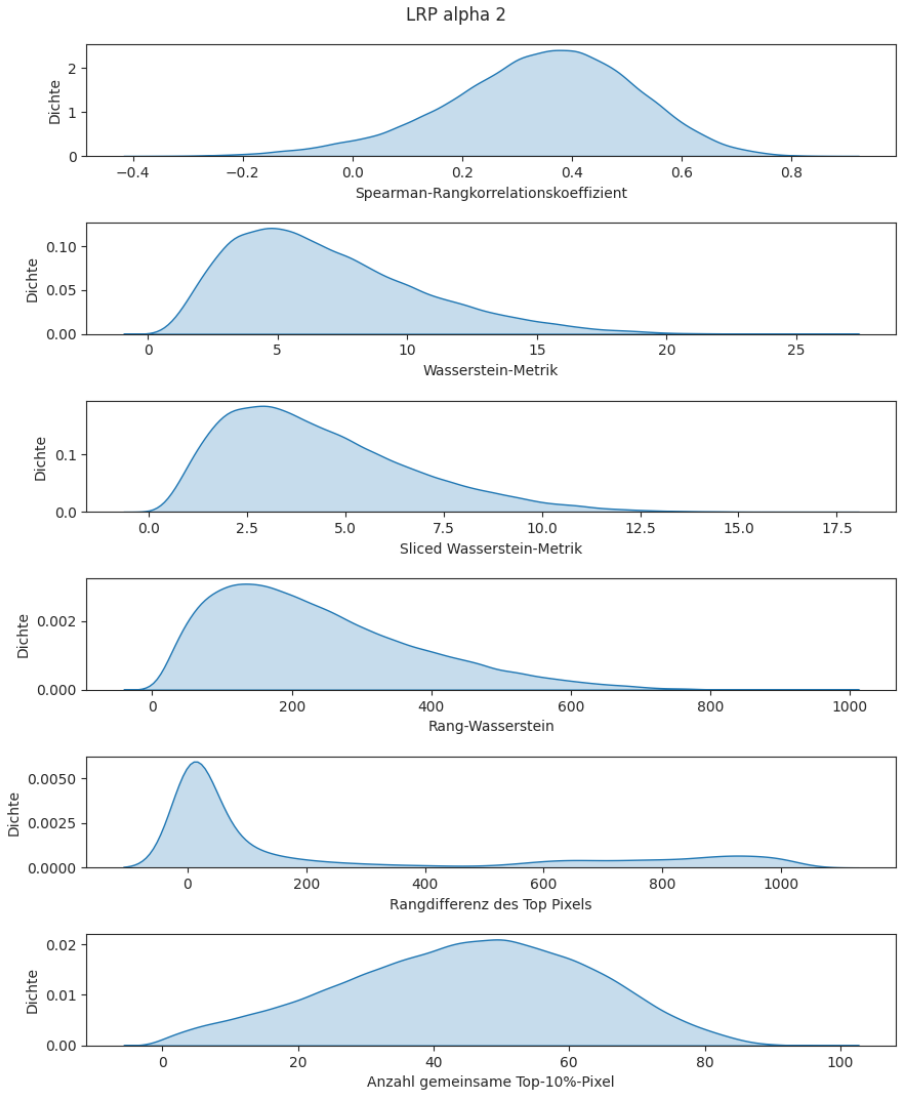


Abbildung 18.: Verteilungen der Vergleichsmetriken des LRP-Verfahrens mit $\alpha = 2$

Metrik	Quantile					
	0,10	0,25	0,50	0,75	0,95	1,00
Spearman Kor.	0,1156	0,2356	0,3544	0,4612	0,6002	0,8692
Wasserstein	2,7411	4,1066	6,2201	9,0559	14,0301	26,2578
Sliced WS	1,6105	2,4777	3,8633	5,7280	8,9524	17,3087
Rang-WS	71,9623	123,1733	207,1587	322,6167	520,4510	966,9688
Rangdifferenz	0	5	36	432	946	1023

Tabelle 10.: Quantile der vergleichenden Metriken für LRP mit $\alpha = 2$

zwingend auch einen hohen Rang im Adversarial Example. Die Rangdifferenz zeigt jedoch, dass die Verschiebung des Top-Pixels im Originalbild in 50 % der Fälle maximal 36 Ränge umfasst. Bei 63,24 % der Adversarial Examples ist der Top-Pixel

des Originalbildes auch unter den Top-10 %-Pixeln des Adversarial Examples. Mit einem Median von 207,16 ist die Rang-Wasserstein-Metrik relativ niedrig, da ihre obere Schranke 1023 beträgt. Auch die normale Wasserstein-Metrik liegt mit einem Median von 3,86 eher im niedrigen Bereich. Die obere Schranke für die Wasserstein-Metrik ist in diesem Fall 43,84.

6.3.3 ε-Regel mit $\varepsilon = 1$

Zuletzt wurde das LRP-Verfahren mit der ε-Regel ausgewertet, wobei der Parameter $\varepsilon = 1$ gesetzt wurde. Hierbei ist aufgefallen, dass es 291 Adversarial Examples und 31 Originalbilder gibt, für die keine vollständige Saliency Map berechnet werden konnte. In jeder dieser Saliency Maps hat mindestens ein Relevance Score den Wert *NaN*. Diese wurden aus der Analyse herausgenommen.

Abbildung 19 zeigt die Visualisierung der Saliency Maps und die Verteilungen der Relevance Scores. Die Visualisierungen als Heatmap deuten an, dass relativ wenige

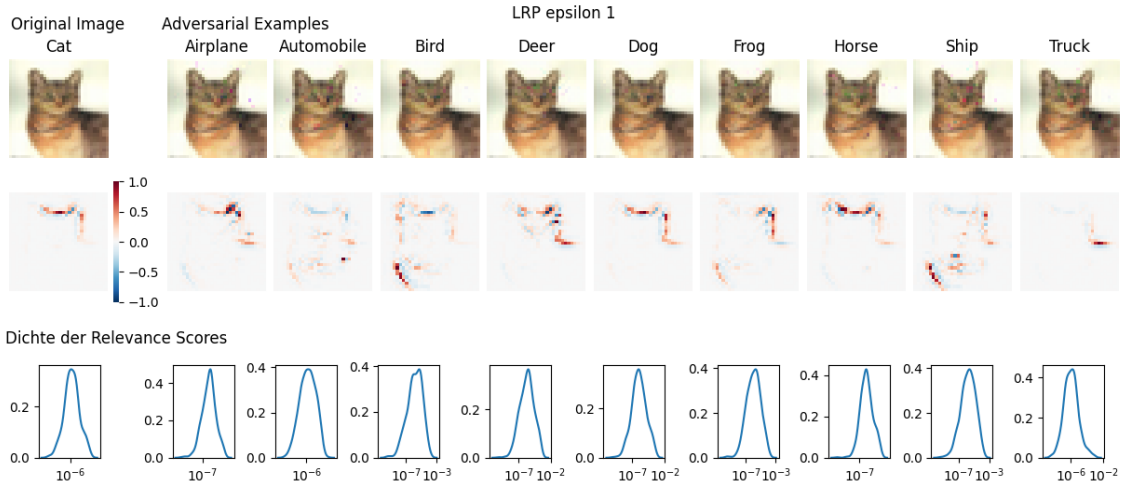


Abbildung 19.: Darstellung der Saliency Maps mit LRP und $\varepsilon = 1$ für das Beispielbild Katze. Die erste Zeile zeigt das Originalbild sowie die Adversarial Examples. Die zweite Zeile zeigt die Relevance Scores als Heatmap. Die dritte Zeile zeigt die Dichte der Relevance Scores in der Saliency Map. Die x-Achse ist logarithmiert.

Pixel einen Einfluss auf die Klassifikation des CNN haben. Die Saliency Maps der Klassen Dog, Deer oder Horse weisen visuell einen höheren Ähnlichkeitsgrad zur Saliency Map des Originalbildes auf. Andere Adversarial Examples erzeugen Saliency Maps mit unterschiedlichen relevanten Pixeln, wie zum Beispiel bei den Klassen Truck oder Bird.

Tabelle 11 präsentiert die verschiedenen Lageparameter der Relevance Scores für die unterschiedlichen Klassen des Katzenbildes. Das LRP-Verfahren berechnet niedrige Relevance Scores, da für das Katzenbild ein maximaler Relevance Score von 0,001

Label	Relevance Scores				
	Mittelwert	Maximum	Varianz	Totale Variation	Erklärungs-quote
Airplane	0,000003	0,000121	$9{,}078698 \cdot 10^{-11}$	0,005074	0,139648
Automobile	0,000004	0,000223	$1{,}382280 \cdot 10^{-10}$	0,008185	0,164062
Bird	0,000004	0,000138	$1{,}381193 \cdot 10^{-10}$	0,007505	0,167969
Cat	**0,000015**	**0,001086**	$\mathbf{5{,}679232} \cdot 10^{-9}$	**0,029731**	**0,064453**
Deer	0,000010	0,000424	$1{,}346909 \cdot 10^{-9}$	0,021280	0,107422
Dog	0,000008	0,000414	$1{,}145876 \cdot 10^{-9}$	0,014806	0,081055
Frog	0,000003	0,000169	$1{,}434040 \cdot 10^{-10}$	0,006605	0,147461
Horse	0,000006	0,000252	$4{,}882364 \cdot 10^{-10}$	0,011131	0,102539
Ship	0,000004	0,000158	$2{,}088727 \cdot 10^{-10}$	0,009349	0,148438
Truck	0,000007	0,000862	$2{,}255472 \cdot 10^{-9}$	0,013330	0,070312

Tabelle 11.: Metriken des Katzenbildes für LRP mit $\varepsilon = 1$. Das Originalbild ist in fetter Schrift hervorgehoben.

im Originalbild zu finden ist. Die Mittelwerte variieren lediglich zwischen 0,000003 bei den Klassen Airplane und Frog und 0,000015 bei der Klasse Cat. Die Erklärungsquoten, welche zwischen 6,45 % (für die Klasse Cat) und 16,80 % (für die Klasse Bird) liegen, zeigen, dass die meisten Relevance Scores sich auf wenige Pixel konzentrieren.

Abbildung 20 illustriert die Verteilungen der Lageparameter sowohl für die Originalbilder als auch für die Adversarial Examples durch Boxplots. Um eine klarere Darstellung zu ermöglichen, sind die x-Achsen für die Mittelwerte, die Maxima, die Varianzen und die totalen Variationen logarithmisch skaliert. Die Boxplots zeigen gewisse Unterschiede in den Verteilungen. Besonders auffällig ist, dass der Interquartilsabstand (IQR) bei den Adversarial Examples erneut geringer ausfällt, was auf eine kompaktere Verteilung hindeutet. Lediglich die Erklärungsquoten zeigen keine großen Unterschiede zwischen den beiden Gruppen.

Metrik	Quantile					
	0,10	0,25	0,50	0,75	0,95	1,00
Spearman Kor.	0,0758	0,1878	0,3013	0,4090	0,5700	0,8748
Wasserstein	2,3694	3,5638	5,4121	7,9858	12,6286	25,0435
Sliced WS	1,3820	2,1451	3,3484	5,0353	8,0586	16,1478
Rang-WS	74,2723	131,2690	220,7223	335,7375	521,7157	869,3360
Rangdifferenz	0	3	19	551	946	1023

Tabelle 12.: Quantile der vergleichenden Metriken für LRP epsilon 1

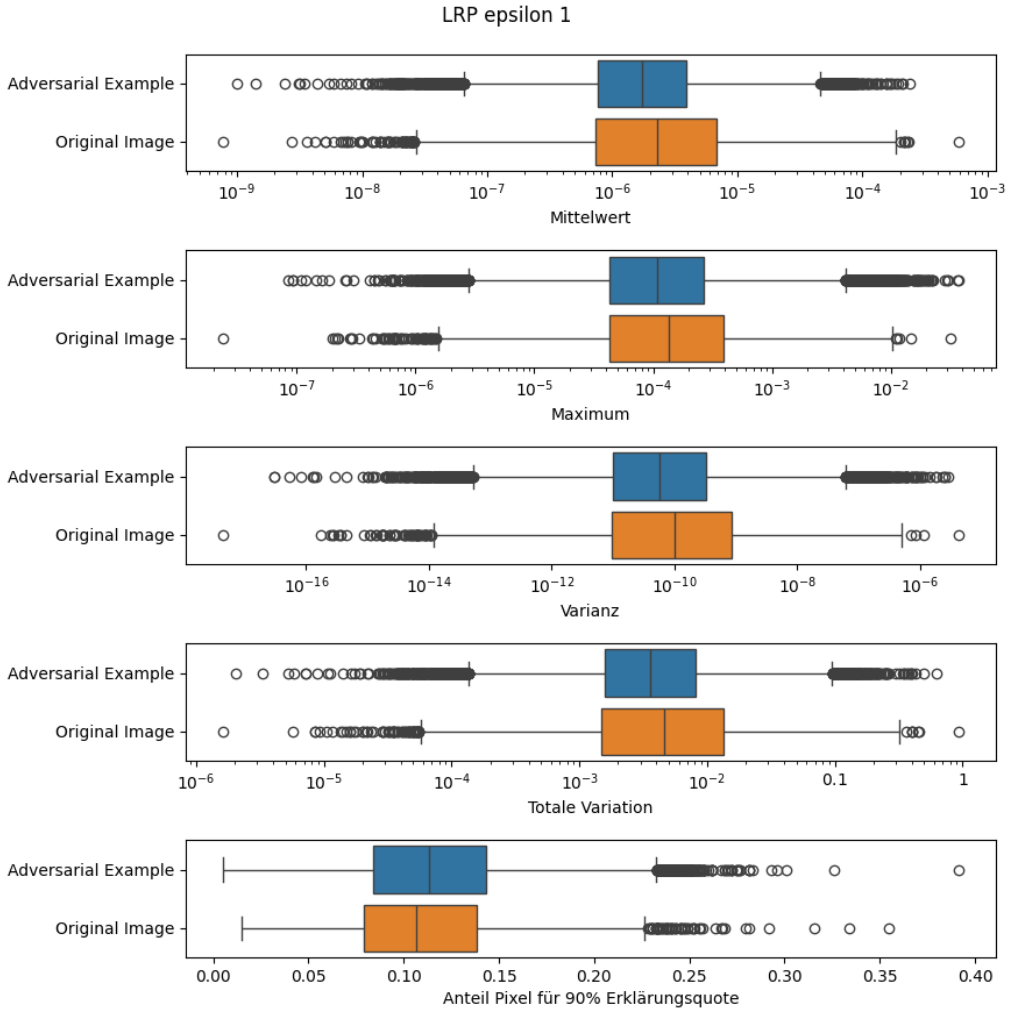


Abbildung 20.: Vergleich der Standardmetriken für Saliency Maps des LRP-Verfahrens mit $\varepsilon = 1$ zwischen Originalbildern und Adversarial Examples

Die quantitativen Vergleichsmetriken für die LRP-ε sind in Abbildung 21 und Tabelle 12 dargestellt. Es wird direkt ersichtlich, dass der Spearman-Rangkorrelationskoeffizient insgesamt niedrig ist. Bei einem 0,95-Quantil von 0,57 zeigt sich, dass fast 95 % der Adversarial Examples eine schwache positive Rangkorrelation zum Originalbild aufweisen. Dies impliziert, dass Pixel, die im Originalbild einen hohen Rang haben, nicht unbedingt einen hohen Rang im Adversarial Example besitzen. Die Rangdifferenz verdeutlicht, dass in 50 % der Fälle der Top-Pixel im Originalbild um maximal 19 Ränge verschoben wird. Bei 68,84 % der Adversarial Examples gehört der Top-Pixel des Originalbildes weiterhin zu den Top 10 % der Pixel im Adversarial Example. Allerdings zeigt sowohl die Verteilung in Abbildung 21 als auch das 0,95-Quantil, dass es einige Ausreißer bei der Rangverschiebung nach oben gibt. Die Rang-Wasserstein-Metrik zeigt mit einem Median von 220,72 ebenfalls eher geringe Werte, da die maximale Schranke

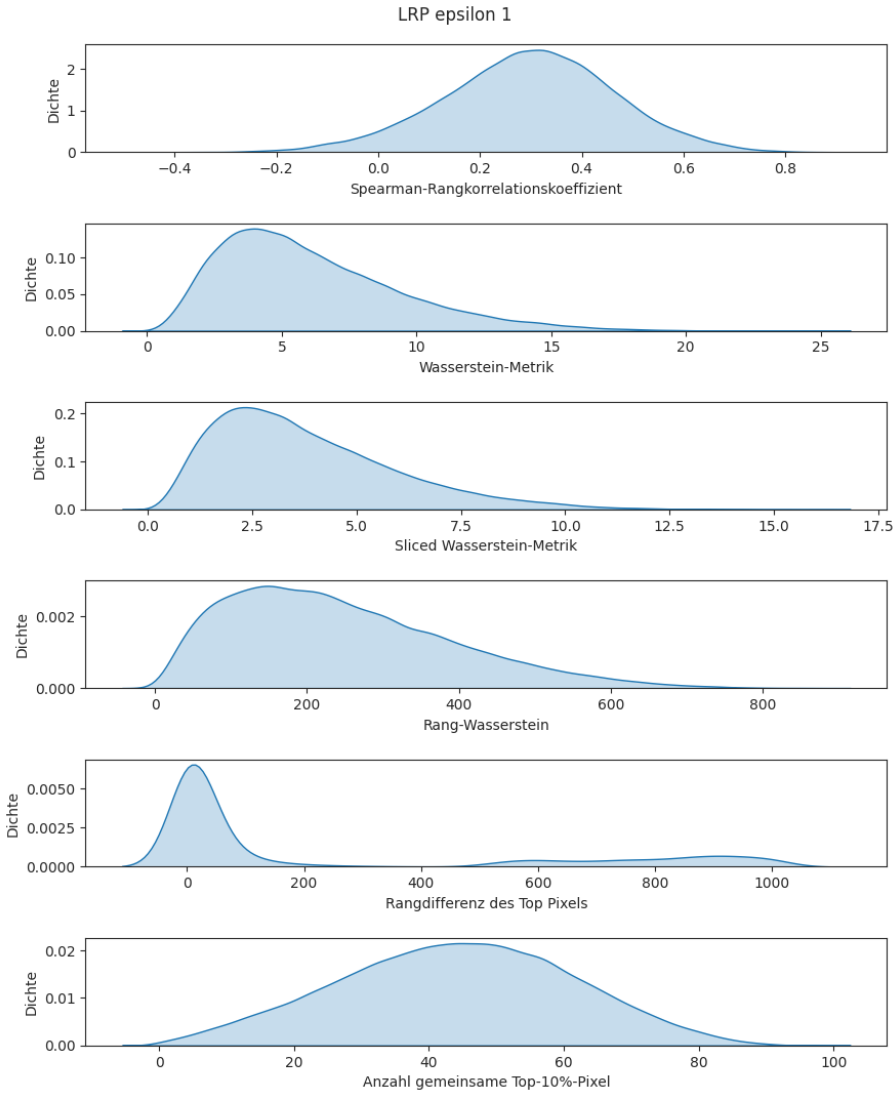


Abbildung 21.: Verteilungen der Vergleichsmetriken des LRP-Verfahrens mit $\varepsilon = 1$

bei 1023 liegt. Auch die Wasserstein-Metrik liegt mit einem Median von 5,4121 im unteren Bereich, wobei die maximale Schranke in diesem Fall 43,84 beträgt.

6.4 Vergleich der Saliency Map-Methoden

Dieses Kapitel stellt die verschiedenen Saliency Map-Methoden im Vergleich dar. In den vorangegangenen Abschnitten wurde deutlich, dass die Werte der berechneten Relevance Scores stark voneinander abweichen. Für eine Vergleichbarkeit der Methoden wurden daher alle vergleichenden Metriken geeignet skaliert.

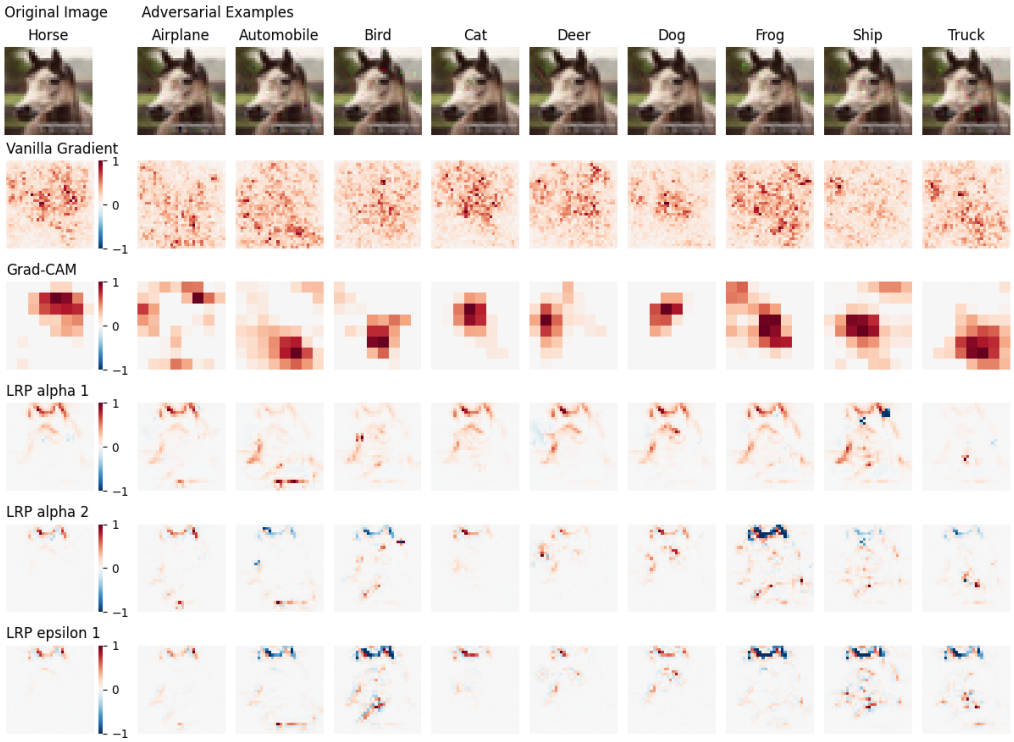


Abbildung 22.: Visuelle Darstellung der Saliency Maps für alle Verfahren anhand des Beispielbildes Pferd

Die Abbildung 22 zeigt die visuelle Darstellung der Saliency Maps aller Verfahren für ein weiteres Beispielbild. Die LRP-Verfahren weisen in dieser Darstellung einen hohe visuelle Ähnlichkeit auf. Das Vanilla Gradient-Verfahren zeigt unterschiedliche, aber auch sehr verrauschte Saliency Maps. Das Grad-CAM-Verfahren hebt sich durch seine geringe Auflösung ab. Allerdings zeigt das Grad-CAM-Verfahren bei subjektiver Betrachtung die größten visuellen Unterschiede zwischen den Saliency Maps der Originalbilder und den Adversarial Examples.

Ein wesentlicher Unterschied zwischen den Verfahren liegt in den Verteilungen von Relevance Scores auf die Pixel. Abbildung 23 zeigt beispielhaft für das Katzenbild, wie sich der Anteil der Erklärbarkeit erhöht, je größer der Anteil betrachteter Pixel ist. Ausschließlich bei dem Vanilla Gradient-Verfahren sind viele Pixel erforderlich, um einen hohen Anteil an der Erklärbarkeit zu erhalten. Beim Grad-CAM-Verfahren unterscheidet sich die Anzahl relevanter Pixel in den Saliency Maps stark.

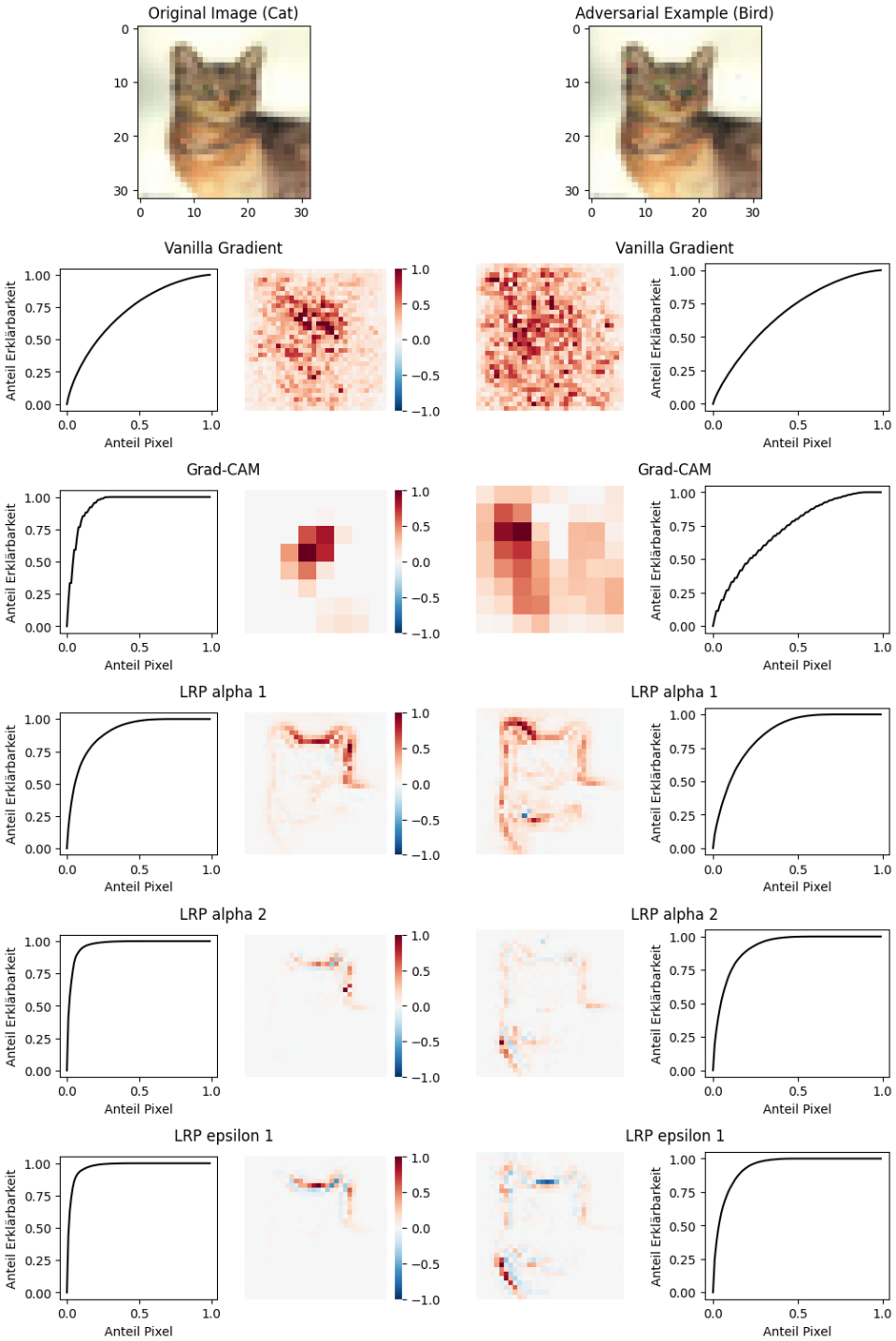


Abbildung 23.: Visuelle Darstellung der Saliency Maps und Erklärungsquoten für alle Verfahren anhand des Beispielbildes Katze

Die Verteilung der Erklärungsquoten, also dem benötigten Anteil an Pixeln für 90 % Erklärbarkeit, der Originalbilder und Adversarial Examples je Erklärmodell, ist in Abbildung 24 dargestellt. Die Quantile werden in Tabelle 13 zusammengefasst. Da die Erklärungsquoten bereits in Prozent angegeben werden, ist diese Metrik nicht skaliert.

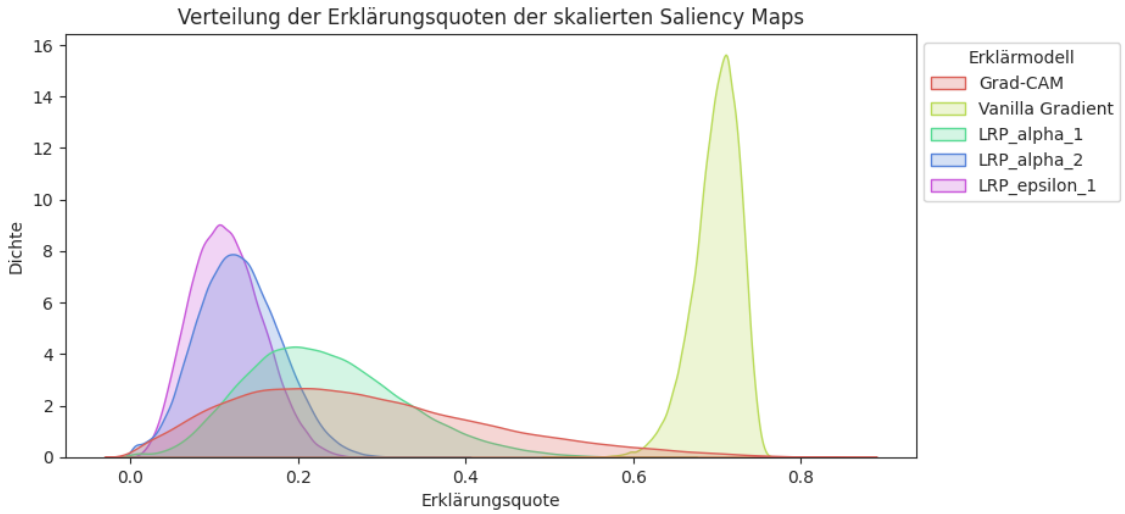


Abbildung 24.: Verteilung der Erklärungsquote

Erklärmodell	Quantile						
	0,00	0,10	0,25	0,50	0,75	0,95	1,00
Grad-CAM	0,0156	0,0781	0,1562	0,25	0,3594	0,5625	0,8438
Vanilla Gradient	0,499	0,6611	0,6826	0,7021	0,7188	0,7373	0,7676
LRP $\alpha = 1$	0,0029	0,1191	0,1641	0,2227	0,291	0,3994	0,6475
LRP $\alpha = 2$	0,001	0,0664	0,0938	0,127	0,1621	0,2129	0,3906
LRP $\varepsilon = 1$	0,001	0,0605	0,083	0,1123	0,1426	0,1875	0,3916

Tabelle 13.: Quantile der Verteilung der Erklärungsquote

Der Vergleich der Verteilungen zeigt, dass die Erklärungsquote bei den Vanilla Gradient Saliency Maps am höchsten ist und eine kleine Streuung aufweist. Dieses Verfahren verteilt die Relevance Scores in allen Saliency Maps daher auf die meisten Pixel. Im Gegensatz dazu weist das Grad-CAM-Verfahren die flachste Verteilung auf. Das bedeutet, dass es sowohl Saliency Maps mit wenig relevanten Pixeln, als auch Saliency Maps mit vielen relevanten Pixeln gibt. Die LRP-Verfahren mit $\alpha = 2$ und $\varepsilon = 1$ verteilen die Relevanz auf die wenigsten Pixel. Wenn die Fokussiertheit von Relevanz auf wenige Pixel als Qualitätskriterium definiert würde, schneidet das Vanilla Gradient-Verfahren am schlechtesten ab.

Für die Vergleichbarkeit wurde die totale Variation auch für die skalierten Saliency Maps aller Methoden berechnet. Das hat zur Folge, dass die Werte für das Vanilla

Gradient-Verfahren sehr gering ausfallen. Dies widerspricht der visuellen Darstellung, wobei das Vanilla Gradient-Verfahren ein starkes Rauschen aufweist. Die Werte für die Heatmaps sind nicht auf eine Dichte, sondern mittels des Maximalwertes auf das Intervall $[0, 1]$ skaliert. Da die totale Variation stark von den Größen der Relevance Scores abhängt, zeigt sie für die fokussierten Verfahren die größten Werte und für das stark verteilte Vanilla Gradient-Verfahren die geringsten Werte. Das Grad-CAM-Verfahren weist mit 0,88 ebenfalls einen relativ kleinen Wert für die totale Variation auf. Allerdings deuten die großen Abstände zwischen den Quantilen darauf hin, dass die Saliency Maps je nach Bild stark variieren.

Kombiniert man die Ergebnisse zur Erklärungsquote mit der totalen Variation der skalierten Saliency Maps, so lässt sich vermuten, dass eine geringe Erklärungsquote zu einer hohen totalen Variation führt.

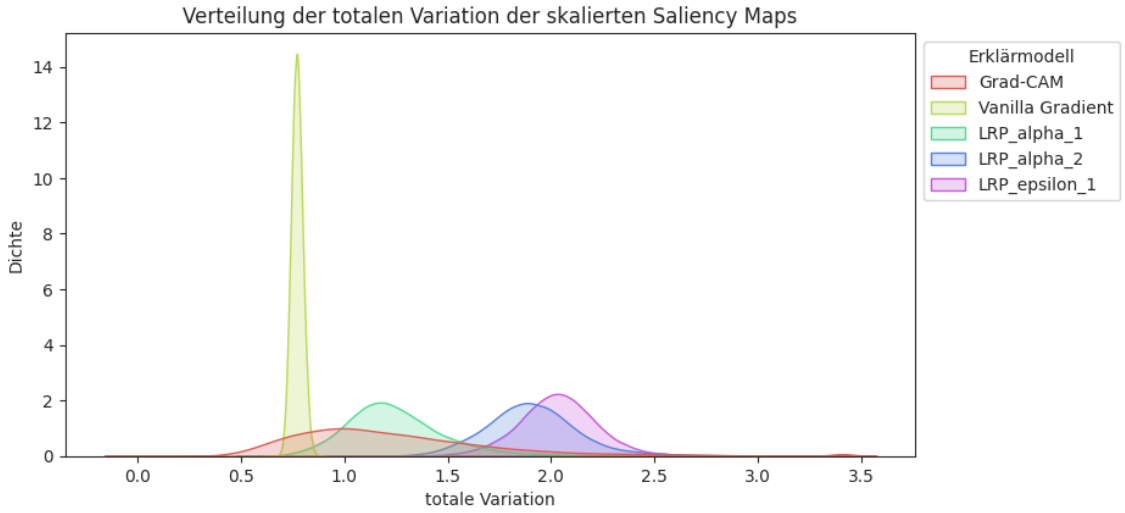


Abbildung 25.: Verteilung der totalen Variation der skalierten Saliency Maps

Erklärmodell	Quantile						
	0,00	0,10	0,25	0,50	0,75	0,95	1,00
Grad-CAM	0,0	0,689	0,878	1,1371	1,4855	2,2847	3,4142
Vanilla Gradient	0,6182	0,7351	0,7513	0,7697	0,7885	0,8169	0,8938
LRP $\alpha = 1$	0,5484	0,9719	1,086	1,218	1,3784	1,7073	3,1712
LRP $\alpha = 2$	0,7696	1,6295	1,762	1,9006	2,0452	2,3333	3,4039
LRP $\varepsilon = 1$	0,9106	1,7903	1,9099	2,0312	2,1524	2,3528	3,2254

Tabelle 14.: Quantile der Verteilung der totalen Variation der skalierten Saliency Maps

Tabelle 15 zeigt die Quantile der Verteilungen des jeweils größten Relevance Scores einer Saliency Map je Erklärmodell. Analog zur Erklärungsquote verteilt das Vanilla Gradient-Verfahren die kleinsten Relevance Scores. Der größte Relevance Score in

allen skalierten Vanilla Gradient Saliency Maps liegt bei gerade einmal 0,02. Dieser Wert ist damit kleiner als die Mediane aller anderen Methoden.

Erklärmodell	Quantile						
	0,00	0,10	0,25	0,50	0,75	0,95	1,00
Grad-CAM	0,0	0,0577	0,0784	0,1116	0,1681	0,3713	1,0
Vanilla Gradient	0,0026	0,0039	0,0043	0,0049	0,0058	0,0076	0,0228
LRP $\alpha = 1$	0,0045	0,0142	0,0189	0,0269	0,041	0,0982	0,6369
LRP $\alpha = 2$	0,0129	0,0328	0,0426	0,0595	0,0892	0,1995	0,9856
LRP $\varepsilon = 1$	0,0113	0,0341	0,0436	0,0587	0,082	0,1459	0,717

Tabelle 15.: Quantile der Verteilung der maximalen Relevance Scores der skalierten Saliency Maps

Die drei Einzelmetriken Erklärungsquote, totale Variation und das Maximum der skalierten Saliency Maps zeigen, wie unterschiedlich die verschiedenen Methoden ihre Relevance Scores verteilen. Das Vanilla Gradient-Verfahren verteilt kleine Relevance Scores auf viele Pixel, wohingegen die LRP-Verfahren hohe Relevance Scores auf wenige Pixel verteilen. Bei den Grad-CAM Saliency Maps gibt es die größten Unterschiede. Es gibt sowohl Saliency Maps mit großen und konzentrierten Relevance Scores als auch Saliency Maps mit verteilten kleinen Relevance Scores.

Aufgrund der unterschiedlichen Dimensionen der Saliency Maps können die Wasserstein-Metriken nicht direkt miteinander verglichen werden. Um die Metriken dennoch für alle Methoden vergleichbar zu machen, wurden alle Ergebnisse durch die jeweilige obere Schranke geteilt. Die so skalierten Wasserstein-Metriken können nun nur noch Werte zwischen 0 und 1 annehmen.

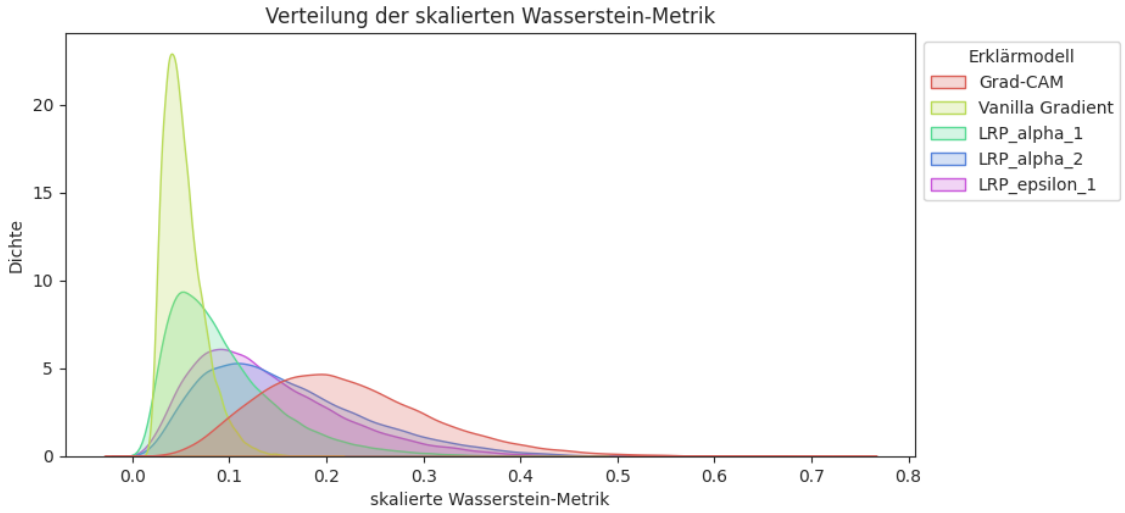


Abbildung 26.: Verteilung der skalierten Wasserstein-Metrik

Erklärmodell	Quantile						
	0,00	0,10	0,25	0,50	0,75	0,95	1,00
Grad-CAM	0,0	0,1143	0,1551	0,2093	0,274	0,386	0,7383
Vanilla Gradient	0,0116	0,0299	0,0372	0,0484	0,0637	0,0932	0,2117
LRP $\alpha = 1$	0,0055	0,0351	0,0517	0,08	0,1217	0,2102	0,4727
LRP $\alpha = 2$	0,005	0,0625	0,0937	0,1419	0,2066	0,32	0,5989
LRP $\varepsilon = 1$	0,0031	0,054	0,0813	0,1234	0,1822	0,2881	0,5712

Tabelle 16.: Quantile der Verteilung der skalierten Wasserstein-Metrik

Die Verteilung der Wasserstein-Metrik für das Grad-CAM-Verfahren hat mit 0,21 den größten Median. Das deutet darauf hin, dass bei diesem Verfahren die größten Unterschiede zwischen Saliency Map des Originalbildes und den Adversarial Examples bestehen.

Überraschenderweise hat das Vanilla Gradient-Verfahren die geringsten Wasserstein-Metriken, da die Visualisierung der Heatmap unterschiedliche Saliency Maps zeigt. Hier spielt wieder die Skalierung der Relevance Scores eine entscheidende Rolle. Da beim Vanilla Gradient-Verfahren wenig Masse auf viele Pixel verteilt wird, muss beim Transport einer Verteilung in die andere nur wenig Masse über kurze Distanzen bewegt werden. Somit kommen kleine Wasserstein-Metriken zu Stande.

Den umgekehrten Fall beobachtet man bei den LRP-Verfahren. Obwohl die Visualisierung als Heatmap auf sehr ähnliche Saliency Maps hindeutet, haben die Wasserstein-Metriken der $\alpha\beta$-Regeln den zweitgrößten Median. Diese Verfahren konzentrieren hohe Relevance Scores auf wenige Pixel. Sobald in der Saliency Map eines Adversarial Examples einige Pixel mit größerer euklidischer Distanz zu den relevanten Pixeln des Originalbildes hohe Relevance Scores erhalten, führt dies zu einer größeren Wasserstein-Metrik.

Durch die Skalierung der Wasserstein-Metrik wird allerdings deutlich, dass alle Verfahren relativ kleine Werte aufweisen, was darauf hindeutet, dass sie eher ähnliche Saliency Maps erzeugen.

Erklärmodell	Quantile						
	0,00	0,10	0,25	0,50	0,75	0,95	1,00
Grad-CAM	0,0	0,1584	0,2578	0,3704	0,4977	0,6754	0,9673
Vanilla Gradient	0,0229	0,1045	0,1191	0,1355	0,1529	0,1816	0,2979
LRP $\alpha = 1$	0,0016	0,019	0,0342	0,0618	0,1087	0,2628	0,8468
LRP $\alpha = 2$	0,0045	0,0703	0,1204	0,2025	0,3154	0,5087	0,9452
LRP $\varepsilon = 1$	0,0026	0,0726	0,1283	0,2158	0,3282	0,51	0,8498

Tabelle 17.: Quantile der Verteilung der skalierten Rang-Wasserstein-Metrik

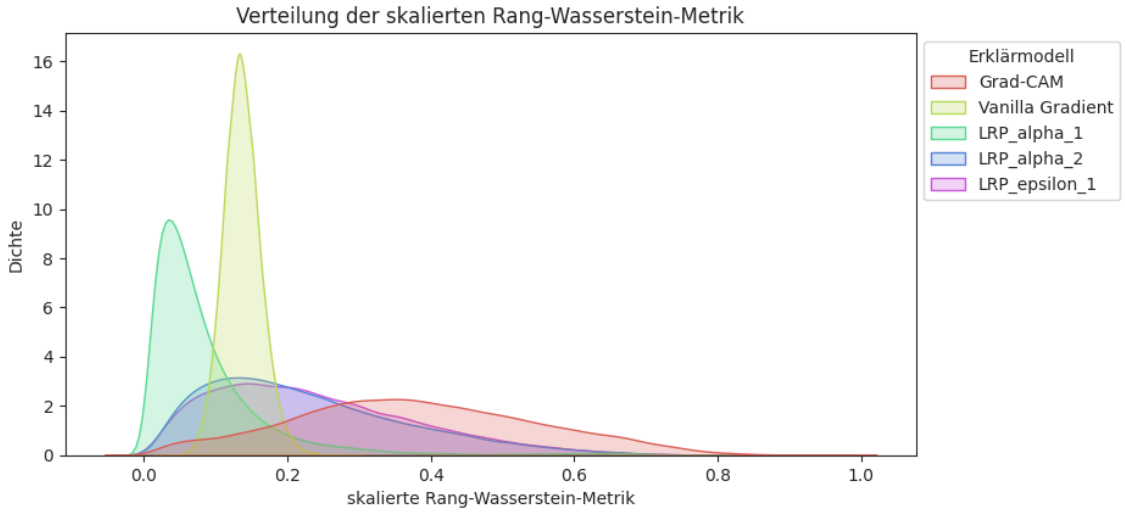


Abbildung 27.: Verteilung der skalierten Rang-Wasserstein-Metrik

Die Rang-Wasserstein-Metrik zeigt, wie sich Masse zwischen den Rängen verschiebt. Sie wurde auch für den Vergleich mit ihrer oberen Schranke skaliert, sodass Werte zwischen 0 und 1 entstanden sind. In diesem Fall, hat das LRP-Verfahren mit $\alpha = 1$ den kleinsten Median, was auf die geringste Verschiebung von Masse unter den Rängen hinweist. Das Vanilla Gradient-Verfahren erzielt wieder sehr kleine Werte. Mit einem Median von 0,14 und einem Interquartilsabstand von 0,03 belegt dieses Verfahren bei der Rang-Wasserstein-Metrik Platz zwei. Auch die LRP-Verfahren mit $\alpha = 2$ und $\varepsilon = 1$ haben relativ kleine Mediane mit 0,20 und 0,22. Allerdings streuen die Metriken in beiden Fällen stark. Es gibt somit sowohl sehr ähnliche als auch eher unterschiedliche Saliency Map-Paare.

Das Grad-CAM-Verfahren hat im Vergleich den größten Median mit 0,37, aber auch die größte Streuung mit einem Interquartilsabstand von 0,24.

Erklärmodell	Quantile						
	0,00	0,10	0,25	0,50	0,75	0,95	1,00
Grad-CAM	-0,7915	0,0285	0,2908	0,5146	0,652	0,8331	1,0
Vanilla Gradient	-0,2376	0,2329	0,3029	0,3813	0,4579	0,5643	0,8185
LRP $\alpha = 1$	-0,6046	0,6707	0,7517	0,8155	0,8623	0,9114	0,9849
LRP $\alpha = 2$	-0,3634	0,1156	0,2356	0,3544	0,4612	0,6002	0,8692
LRP $\varepsilon = 1$	-0,4502	0,0758	0,1878	0,3013	0,409	0,57	0,8748

Tabelle 18.: Quantile der Verteilung des Spearman-Rangkorrelationskoeffizienten

Der Spearman-Rangkorrelationskoeffizient ist eine häufig genutzte Metrik zum Vergleich von Saliency Maps, da er monotone Zusammenhänge zwischen Rängen quantifiziert. Er nimmt standardmäßig nur Werte aus dem Intervall $[-1, 1]$ an und wurde daher für den Vergleich der Erklärmodelle nicht skaliert. Beim Spearman-

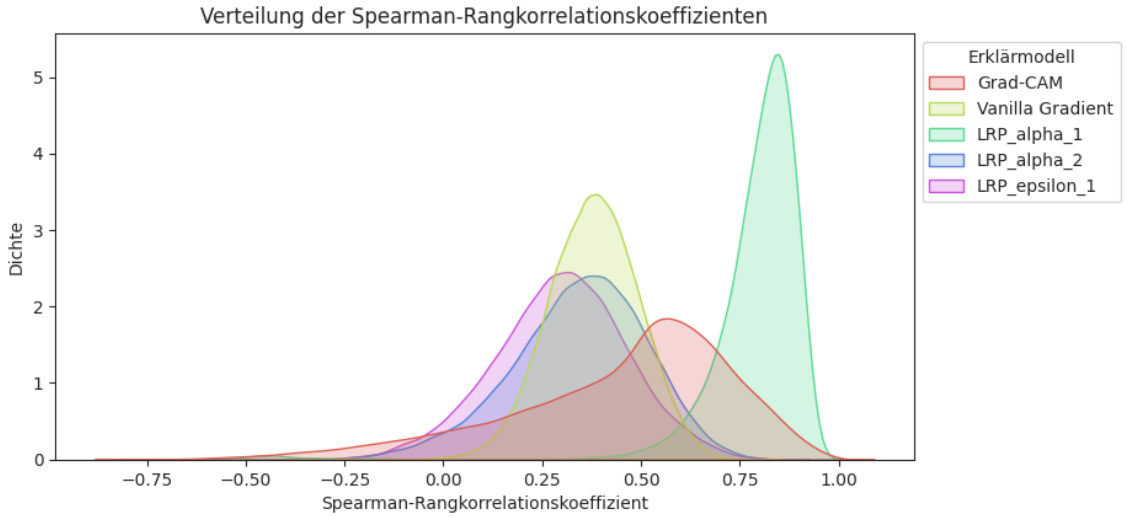


Abbildung 28.: Verteilung des Spearman-Rangkorrelationskoeffizienten

Rangkorrelationskoeffizienten deuten große positive Werte auf eine hohe Korrelation hin und somit auf ähnliche Saliency Maps.

Mit einem Median von 0,86 nehmen die Saliency Maps des LRP-Verfahrens mit $\alpha = 1$ die größten Werte an. Sie haben auch die geringste Streuung mit einem Interquartilsabstand von 0,1. Das Grad-CAM-Verfahren weist beim Spearman-Rangkorrelationskoeffizienten den zweitgrößten Median mit einem Wert von 0,51 auf. Allerdings hat das Erklärmodell auch die größte Streuung mit einem Interquartilsabstand von 0,36. Die anderen Verfahren liegen mit Medianen zwischen 0,30 und 0,38 sehr nahe beieinander. Das Vanilla Gradient-Verfahren hat dabei die kleinste Streuung.

Insgesamt haben alle Verfahren hauptsächlich positive Korrelationen bei den Rängen, da alle 0,1-Quantile größer als 0 sind.

Erklärmodell	Quantile						
	0,00	0,10	0,25	0,50	0,75	0,95	1,00
Grad-CAM	0,0	0,0317	0,127	0,3492	0,7619	0,9524	1,0
Vanilla Gradient	0,0	0,0176	0,0665	0,1935	0,4096	0,7615	1,0
LRP $\alpha = 1$	0,0	0,0	0,001	0,0088	0,0499	0,6852	1,0
LRP $\alpha = 2$	0,0	0,0	0,0049	0,0352	0,4223	0,9247	1,0
LRP $\varepsilon = 1$	0,0	0,0	0,0029	0,0186	0,5386	0,9247	1,0

Tabelle 19.: Quantile der Verteilung der skalierten Rangdifferenz des Top-Pixels

Zuletzt wird das Relevance Ranking betrachtet. Dafür gibt es zwei Metriken. Die erste Metrik berechnet die absolute Rangdistanz des Top-Pixels im Originalbild. Diese Metrik wurde mit der maximal möglichen Distanzverschiebung auf das Intervall $[0, 1]$ für alle Methoden skaliert. Abbildung 29 zeigt die Verteilung der Metrik

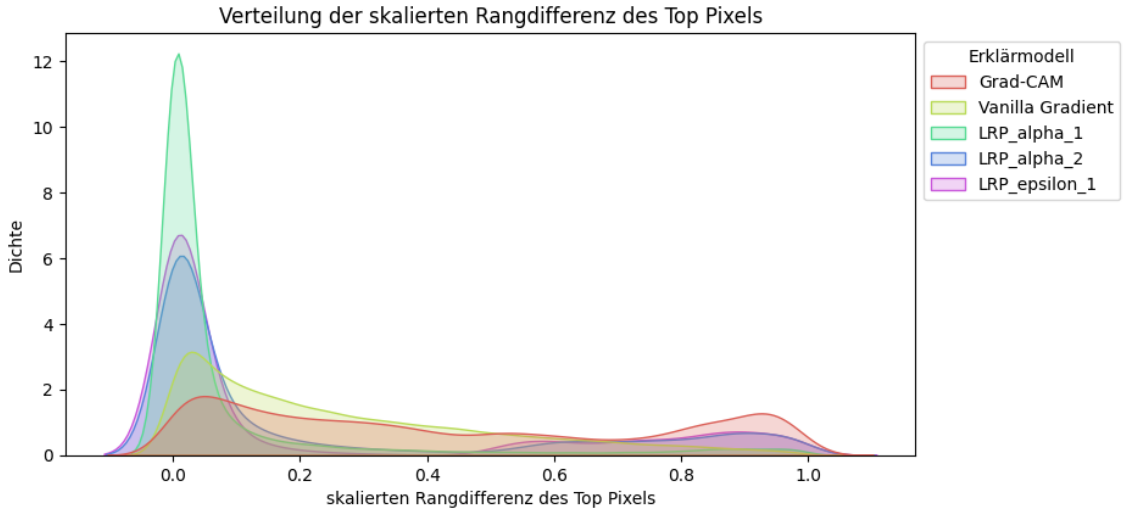


Abbildung 29.: Verteilung der skalierten Rangdifferenz des Top-Pixels

und Tabelle 19 die entsprechenden Quantile. Je größer die Verschiebung, desto unterschiedlicher sind die Saliency Maps.

Am auffälligsten ist die geringe Verschiebung des Top-Pixels bei dem LRP-Verfahren mit $\alpha = 1$. Mit einem Median von 0,009 und einem Interquartilsabstand von 0,049 ist die Verschiebung nicht nur klein, sondern sie streut auch am wenigsten. Die anderen beiden LRP-Verfahren haben ebenfalls kleine Mediane bei der Rangverschiebung, streuen aber mehr. Die meisten großen Verschiebungen liefert das Grad-CAM-Verfahren.

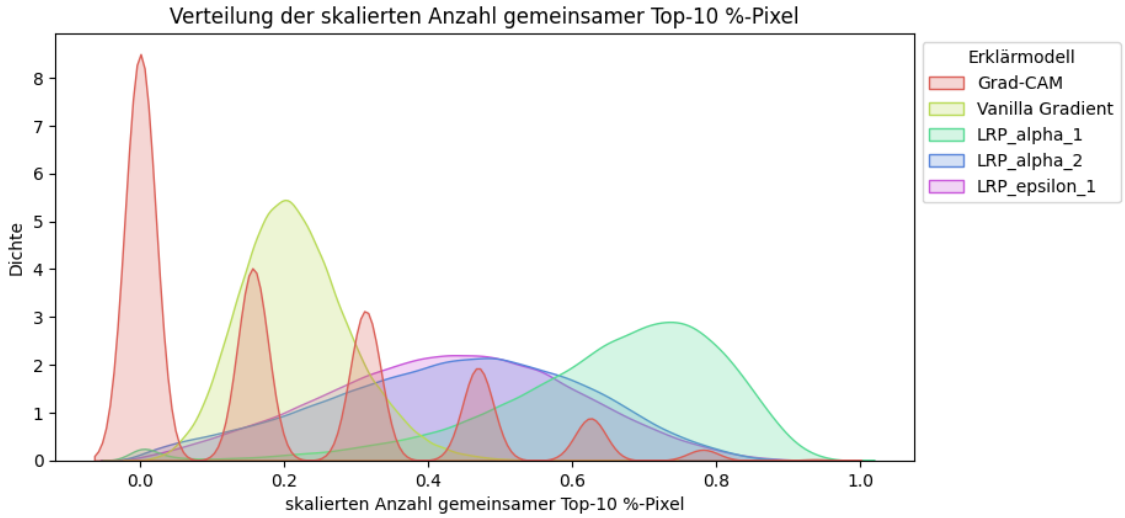


Abbildung 30.: Verteilung der skalierten Anzahl gemeinsamer Top-10 %-Pixel

Die zweite Metrik des Relevance Rankings ist die Größe der Schnittmenge von den Top-10 %-Pixeln im Originalbild und im Adversarial Example. Auch diese Metrik wurde durch die maximal mögliche Schnittmenge geteilt und so auf das Intervall

Erklärmodell	Quantile						
	0,00	0,10	0,25	0,50	0,75	0,95	1,00
Grad-CAM	0,0	0,0	0,0	0,1562	0,3125	0,625	0,9375
Vanilla Gradient	0,0	0,127	0,166	0,2051	0,2637	0,3516	0,7227
LRP $\alpha = 1$	0,0	0,4297	0,5664	0,6738	0,7617	0,8594	0,9668
LRP $\alpha = 2$	0,0	0,1855	0,3125	0,4492	0,5664	0,7129	0,9473
LRP $\varepsilon = 1$	0,0	0,2051	0,3125	0,4297	0,5469	0,7031	0,9473

Tabelle 20.: Quantile der Verteilung der skalierten Anzahl gemeinsamer Top-10 %-Pixel

zwischen 0 und 1 skaliert. Die Saliency Maps gelten als unterschiedlich, je kleiner die Schnittmenge ist.

Beim Grad-CAM-Verfahren ist die maximale Anzahl an gemeinsamen Pixeln $0,1 \cdot 64 \approx 6$ Pixel. Daher entsteht in der Grafik eine wellenförmige Verteilung. Der Median zeigt, dass das Grad-CAM Verfahren die geringsten Schnittmengen aufweist. Auffällig ist das Vanilla Gradient-Verfahren. Mit einem Median von 0,2 und dem kleinsten Interquartilsabstand von 0,089 weisen die Vanilla Gradient Saliency Maps nur wenige gemeinsame Pixel unter den Top-10 %-Pixeln auf. Das LRP-Verfahren mit $\alpha = 1$ weist die größten Schnittmengen unter den relevantesten Pixeln auf. Die beiden LRP-Verfahren mit $\alpha = 2$ und $\beta = 1$ liegen mit den Medianen 0,45 und 0,43 im Mittelfeld und weisen große Streuungen auf.

7 Diskussion

Die Auswertung verschiedener Metriken zu den Saliency Maps hat zu folgenden Ergebnissen geführt:

Die verschiedenen Saliency Map-Methoden unterscheiden sich deutlich voneinander. Den ersten Unterschied gibt es bei der Größe der Relevance Scores. Durch die unterschiedlichen Größen ist kein direkter Vergleich der Saliency Maps zwischen verschiedenen Methoden möglich. Daher müssen die verschiedenen Metriken geeignet skaliert werden.

Einen weiteren Unterschied zeigt die Verteilung der Erklärbarkeit auf die Pixel. Hierbei sticht das Vanilla Gradient-Verfahren hervor, da es jeweils viele Pixel pro Bild als relevant für die Klassifikation einstuft. Im Gegensatz dazu konzentrieren die LRP-Verfahren die Relevance Scores auf wenige Pixel. Das Vanilla Gradient-Verfahren benötigt nicht nur viele Pixel für die Erklärbarkeit, sondern diese verteilen sich auch über das Bild. Die Visualisierungen deuten ein starkes Rauschen an. Mithilfe der totalen Variation wurde versucht, diesen Umstand zu quantifizieren. Allerdings waren die Ergebnisse kontraintuitiv, da das Vanilla Gradient-Verfahren die geringste totale Variation bei den skalierten Saliency Maps aufwies. Um das Rauschen zu quantifizieren und vergleichbare Werte für verschiedene Methoden zu erhalten, muss daher eine bessere Skalierung gefunden, oder eine andere Kennzahl verwendet werden.

Die Wasserstein-Metrik zeigt für beide Distanzmaße, die euklidischen Distanzen und die Rangdistanzen, beim Grad-CAM-Verfahren die größten Ergebnisse. Auch bei den beiden Metriken des Relevance Rankings schneidet das Grad-CAM-Verfahren am besten ab. Das deutet darauf hin, dass die Saliency Maps dieses Verfahrens die größten Unterschiede zwischen den Originalbildern und den Adversarial Examples aufweisen. Damit reagiert das Verfahren am meisten auf die geänderten Bilder. Dem widerspricht allerdings der Spearman-Rangkorrelationskoeffizient. Hier hat das Grad-CAM-Verfahren relativ hohe Werte, aber auch eine große Streuung, was insgesamt auf eine eher mittlere Korrelation hinweist.

Im Vergleich schneidet das LRP-Verfahren mit $\alpha = 1$ am schlechtesten ab, da sowohl die Wasserstein-Metrik als auch die Rang-Wasserstein-Metrik sehr kleine Werte aufweisen. Außerdem ergeben das Relevance Ranking große Schnittmengen bei den Top-10 %-Pixeln und die kleinsten Verschiebungen des Top-Pixels.

Das Vanilla Gradient-Verfahren kann bei den Wasserstein-Metriken ebenfalls nicht überzeugen. Es hat die schlechtesten Werte bei den euklidischen Distanzen und liegt

auch bei der Rangdistanz im unteren Bereich. Nur bei der Schnittmenge der Top-10 %-Pixel erzielt das Vanilla Gradient-Verfahren gute Werte und landet auf Platz 2.

Das LRP-Verfahren mit $\alpha = 2$ und $\varepsilon = 1$ liegt im Vergleich im Mittelfeld, erzielt im Median aber keine hohen Wasserstein-Metriken, wobei die Streuung durchaus größer ist. Beim Spearman-Rangkorrelationskoeffizienten erzielen beide Verfahren aber die besten Ergebnisse.

Die Qualität von Saliency Map-Methoden zu messen ist nicht trivial. Die Nutzung von Adversarial Examples scheint ein hilfreiches Werkzeug zu sein. Es bedarf aber guter Kennzahlen, um die Unterschiede der Saliency Maps zwischen einem Originalbild und seinen Adversarial Examples zu messen. In dieser Arbeit wurden dafür die Wasserstein-Metrik als neue Kennzahl eingeführt und das bekannte Relevance Ranking und der Spearman-Rangkorrelationskoeffizient verwendet. Der Vergleich der Saliency Map-Methoden anhand verschiedener Metriken führt zu unterschiedlichen Ergebnissen. Die Metriken identifizieren unterschiedliche Methoden als die beste, das heißt als diejenige Methode, welche die unterschiedlichsten Saliency Maps erzeugt.

Wie in der Literaturrecherche vorgestellt, gibt es bereits erste Ansätze, die Qualität von Metriken zu messen. In dieser Arbeit wurde keine Bewertung der Metriken vorgenommen. Es kann daher keine Aussage darüber getroffen werden, welche Metrik die bessere Aussage über die Ähnlichkeit von zwei Saliency Maps und somit die Qualität der Erklärmethode trifft.

Die Arbeit hat gezeigt, dass die Wasserstein-Metrik mit euklidischen Distanzen zwischen den Pixelkoordinaten sehr aufwändig zu berechnen ist. Selbst bei den Saliency Maps mit 32×32 Pixeln war ein Rechenaufwand von mehreren Stunden notwendig, um die Ergebnisse für alle Originalbilder und Adversarial Examples zu erhalten. Als Approximation wurde hier bereits die sliced Wasserstein-Metrik vorgestellt, allerdings wurde der Zusammenhang zwischen beiden Metriken nicht weiter untersucht. Es wurde keine Skalierung der sliced Wasserstein-Metrik für einen Vergleich der Methoden vorgenommen.

Der Vergleich der einzelnen Metriken der Saliency Maps, wie dem Mittelwert, der Varianz oder auch der totalen Variation, zwischen den Adversarial Examples und den Originalbildern basiert in dieser Arbeit auf der Interpretation der Boxplots. Durch statistische Tests dieser Kennzahlen ließen sich konkrete Aussagen über das Ausmaß und die Signifikanz der Unterschiede zwischen den Gruppen machen.

Die neuen Wasserstein-Metriken haben sich durch die Skalierung gut für den Vergleich von Methoden geeignet. Eine Aussage über die Qualität eines einzelnen Verfahrens war allerdings schwierig. Hier könnte sich eine neue Forschung anschließen, um die Wasserstein-Metriken in Effektstärken zu unterteilen. Ab welchem Wert gelten Adversarial Examples und Originalbilder als ähnlich und ab welchem Wert als unähnlich?

Eine weitere Forschungsfrage ergibt sich aus der Betrachtung der Kennzahlen einzelner Saliency Maps, wie die Erklärungsquote, der maximale Relevance Score oder

die totale Variation. Hier wäre es interessant zu untersuchen, welchen Einfluss diese Kennzahlen auf die vergleichenden Metriken, beispielsweise die Wasserstein-Metrik haben. Gibt es einen Einfluss? Wie groß ist dieser? So kann ein besseres Verständnis für die genutzte Vergleichsmetrik erhalten werden. Außerdem könnte untersucht werden, inwieweit eine geeignetere Skalierung gewählt werden kann, um die Vergleichbarkeit von verschiedenen Saliency Map-Methoden zu verbessern.

Die Literaturrecherche hat bereits ergeben, dass es viele Ansätze für die Bewertung der Qualität von Saliency Map-Methoden gibt. Diese Arbeit hat dafür die Nutzung von Adversarial Examples um neue Metriken ergänzt. Ein logischer nächster Schritt wäre die Erstellung eines umfassenden Kriterienkatalogs, der mehrere Qualitätsaspekte berücksichtigt. Dabei könnten neben der Verlässlichkeit der Methoden auch deren Nutzen und Anwendbarkeit für den Menschen eine zentrale Rolle spielen. Vorab wäre zu evaluieren, welche Kriterien es für die Nutzbarkeit für Menschen geben kann und wie diese gemessen werden. Die Metriken sollten dabei mit Grenzwerten versehen werden, um eine eindeutige Interpretation zu ermöglichen wie die Effektstärken beim Spearman-Rangkorrelationskoeffizienten.

Zuletzt sollten die Nutzung der Adversarial Examples und die Anwendung der vorgestellten Metriken noch weiter erprobt werden. Dafür können zum einen weitere Saliency Map-Methoden untersucht werden. Zum anderen wäre es interessant, die bisher untersuchten Methoden anhand anderer CNNs und anderer Datensätze zu untersuchen.

8 Fazit

In dieser Arbeit wurden die Saliency Map-Methoden Grad-CAM, Vanilla Gradient und Layer-wise Relevance Propagation untersucht und verglichen. Für das LRP-Verfahren wurden die beiden Zerlegungsregeln $\alpha\beta$-Regel und ε-Regel mit den Parametern $\alpha = 1$, $\alpha = 2$ und $\varepsilon = 1$ betrachtet. Dafür wurde der CIFAR-10 Datensatz und das Convolutional Neural Network von Dieter verwendet. Ziel war es, die Qualität der Saliency Map-Methoden mithilfe von Adversarial Examples zu vergleichen.

Die zugrunde liegende Annahme war, dass Saliency Map-Methoden, die bei Originalbildern und Adversarial Examples ähnliche Saliency Maps erzeugen, nicht auf die Manipulation reagieren und als weniger zuverlässig gelten. Um die Ähnlichkeit der Saliency Maps der Originalbilder und der Adversarial Examples zu messen, wurde neben dem Relevance Ranking und dem Spearman-Rangkorrelationskoeffizienten vor allem die Wasserstein-Metrik genutzt. Sie misst den Unterschied anhand des minimalen Aufwandes, die skalierte Saliency Map des Originalbildes in die Saliency Maps seiner Adversarial Examples zu verschieben. Dafür wurden zwei Distanzmaße gewählt. Zum einen die euklidischen Distanzen zwischen den Pixeln und zum anderen die absolute Differenz zwischen den Rängen der Pixel. Für den Vergleich der Methoden wurden die Wasserstein-Metriken jeweils mit ihrer oberen Schranke skaliert.

Im Ergebnis war auffällig, dass die verschiedenen Methoden sehr unterschiedliche Saliency Maps erzeugen. Sie unterscheiden sich in der Größe der verteilten Relevance Scores und der Verteilung der Relevance Scores auf die Pixel. So verteilt das Vanilla Gradient-Verfahren die Relevanz auf viele Pixel und die LRP-Verfahren konzentrieren sich auf wenige Pixel. Das Grad-CAM-Verfahren erzeugt Saliency Maps der Größe 8×8 und die Verteilung der Relevance Scores in einer Saliency Map variiert am meisten.

Die vergleichenden Metriken haben ebenfalls Unterschiede zwischen den Methoden gezeigt. Bei den Wasserstein-Metriken hat sich das Grad-CAM-Verfahren als beste Methode herausgestellt. Der Spearman-Rangkorrelationskoeffizient hingegen bewertet das LRP mit $\varepsilon = 1$ am besten.

Eine grundlegende Erkenntnis der Arbeit ist, dass die verschiedenen Metriken zu unterschiedlichen Ergebnissen führen. Hier kann in zukünftiger Forschung angeknüpft werden, um einen umfassenden Kriteriankatalog mit Bewertungsempfehlungen zu erstellen. Abschließend lässt sich sagen, dass die Qualität der verschiedenen Saliency Map-Methoden auch von der Wahl der Metrik abhängt. Es besteht weiterhin

Forschungsbedarf, um die Aussagekraft der verschiedenen Metriken zu bewerten und zu standardisieren. Die Arbeit leistet einen wertvollen Beitrag zur Methodendiskussion und legt die Basis für weiterführende Untersuchungen in diesem Bereich.

A Software- und Hardwarespezifikationen

Laptop	
CPU	Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz 1.80 GHz
Arbeitsspeicher	16 GB
Betriebssysten	Windows 10 Pro, 64-Bit-Betriebssystem

OpenStack Instanz	
CPU	16 VCPU auf h_da OpenStack Cluster
Arbeitsspeicher	32 GB
Betriebssysten	Ubuntu 22.04 LTS (Jammy Jellyfish) Cloud Image

Tabelle 21.: Hardwarespezifikationen

Bibliothek	Verion	Lizenz	Website
ipykernel	6.29.1	BSD	https://docs.jupyter.org/ https://ipython.org/
gurobipy	11.0.2	Free Academic	https://www.gurobi.com/
matplotlib	3.8.2	PSF	https://matplotlib.org/
Mosek	10.1.31	Personal Academic License	https://www.mosek.com/
numpy	1.26.4	BSD	https://numpy.org/
pandas	2.2.1	BSD	https://pandas.pydata.org/
POT	0.9.3	MIT	https://pythonot.github.io/
PuLP	2.8.0	MIT	https://coin-or.github.io/pulp/
seaborn	0.13.2	BSD	https://seaborn.pydata.org/
tensorflow	2.15.0	Apache-Lizenz, Version 2.0	https://www.tensorflow.org/

Tabelle 22.: Verwendete Python Bibliotheken

B Tabellen und Grafiken

Label	Typ	Anzahl	Mean	Std.	Min.	1. Q	Median	3. Q	Max.
0	Adv.	7187	0.032	0.028	0.000	0.010	0.024	0.045	0.204
	Orig.	617	0.017	0.027	0.000	0.001	0.005	0.021	0.201
1	Adv.	7675	0.059	0.033	0.000	0.035	0.055	0.0783	0.301
	Orig.	906	0.024	0.026	0.000	0.006	0.016	0.032	0.217
2	Adv.	7848	0.039	0.030	0.000	0.016	0.033	0.056	0.224
	Orig.	786	0.027	0.033	0.000	0.004	0.012	0.038	0.185
3	Adv.	8113	0.031	0.020	0.000	0.017	0.027	0.042	0.152
	Orig.	727	0.026	0.022	0.000	0.011	0.019	0.034	0.165
4	Adv.	7799	0.037	0.027	0.000	0.016	0.031	0.051	0.192
	Orig.	838	0.017	0.023	0.000	0.003	0.009	0.023	0.172
5	Adv.	7822	0.039	0.026	0.000	0.019	0.033	0.053	0.283
	Orig.	783.0	0.021	0.018	0.000	0.008	0.016	0.028	0.122
6	Adv.	7880	0.036	0.025	0.000	0.016	0.032	0.051	0.164
	Orig.	937	0.018	0.022	0.000	0.004	0.010	0.022	0.137
7	Adv.	7615	0.032	0.023	0.000	0.016	0.027	0.043	0.196
	Orig.	874	0.020	0.020	0.000	0.005	0.014	0.026	0.180
8	Adv.	7318	0.066	0.047	0.000	0.029	0.058	0.093	0.313
	Orig.	823	0.028	0.037	0.000	0.003	0.013	0.039	0.205
9	Adv.	7593	0.040	0.028	0.000	0.019	0.036	0.055	0.203
	Orig.	651	0.013	0.020	0.000	0.001	0.005	0.017	0.130

Tabelle 23.: Auswertung des Durchschnitts der Relevance Scores je Saliency Maps für Grad-CAM

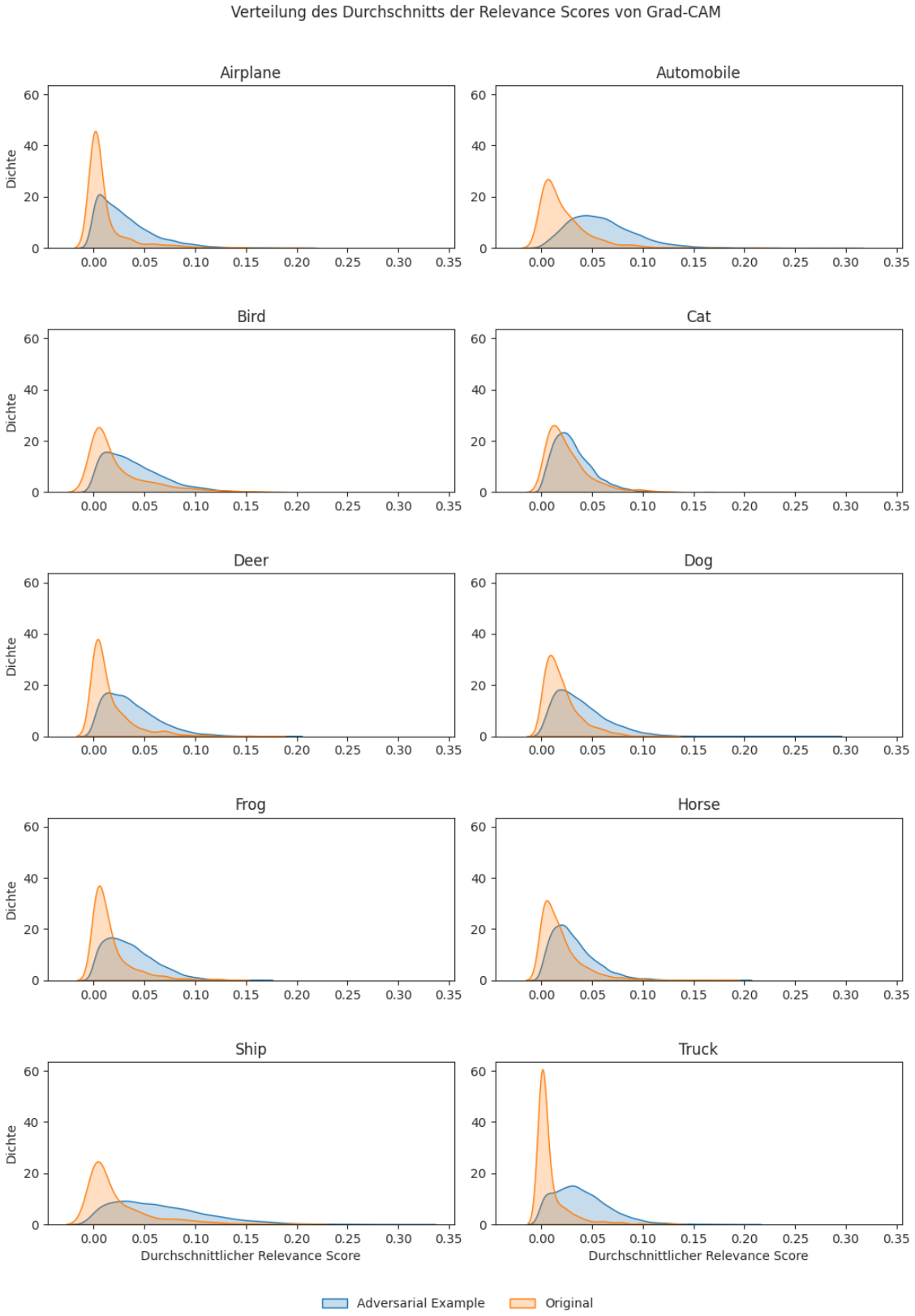


Abbildung 31.: Verteilung des Durchschnitts der Relevance Scores für Grad-CAM

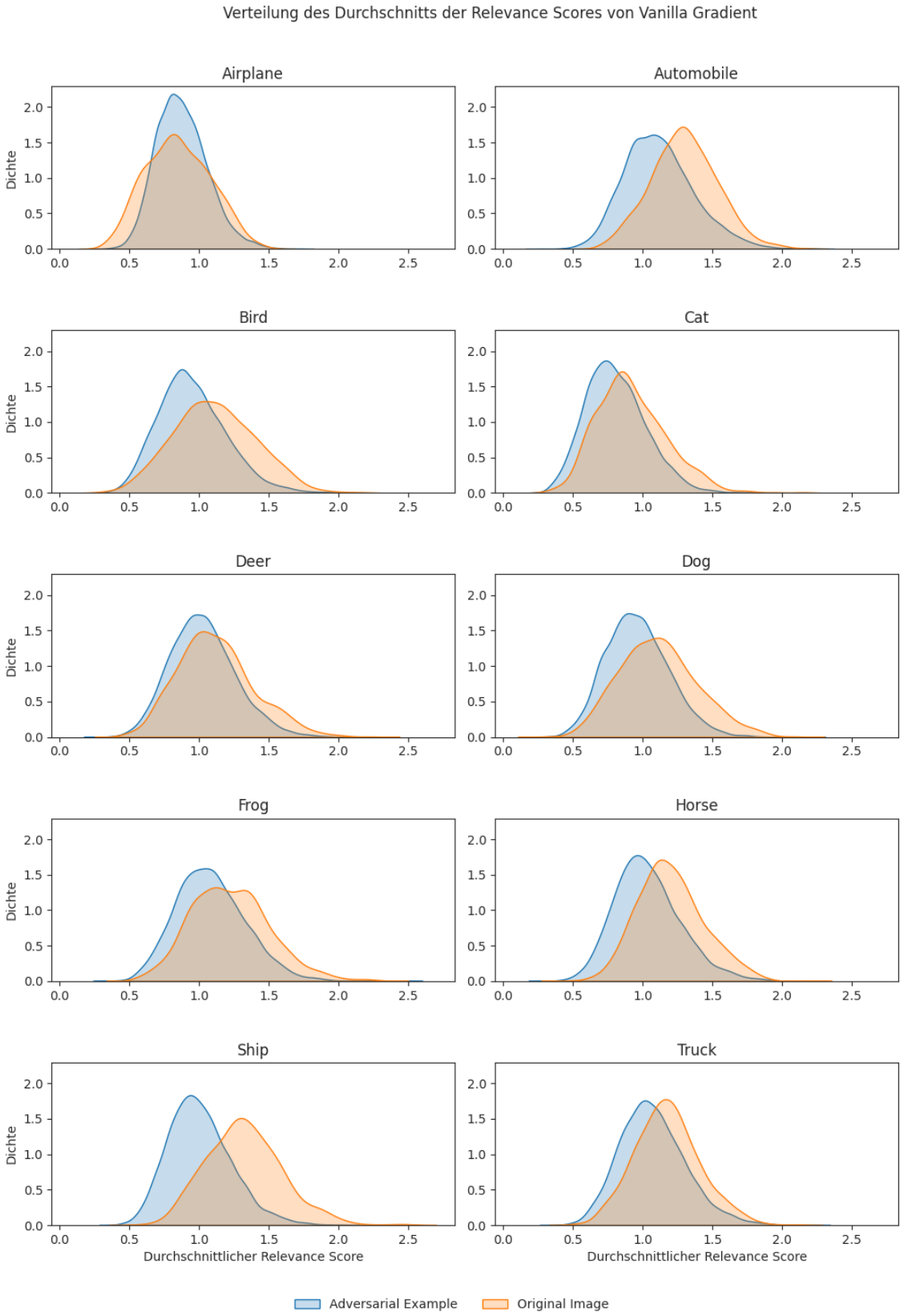


Abbildung 32.: Verteilung des Durchschnitts der Relevance Scores für Vanilla Gradient

Label	Typ	Anzahl	Mean	Std.	Min.	1. Q	Median	3. Q	Max.
0	Adv.	7391.0	0.874	0.183	0.344	0.742	0.859	0.989	1.729
	Orig.	897.0	0.848	0.230	0.310	0.675	0.834	1.014	1.510
1	Adv.	7678.0	1.116	0.251	0.285	0.938	1.095	1.268	2.249
	Orig.	962.0	1.297	0.237	0.681	1.138	1.292	1.449	2.158
2	Adv.	7896.0	0.949	0.245	0.318	0.777	0.923	1.099	2.160
	Orig.	858.0	1.110	0.293	0.301	0.907	1.102	1.313	2.182
3	Adv.	8118.0	0.805	0.218	0.235	0.649	0.783	0.941	2.114
	Orig.	727.0	0.920	0.251	0.312	0.749	0.886	1.067	2.151
4	Adv.	7825.0	1.025	0.237	0.291	0.860	1.010	1.172	2.166
	Orig.	905.0	1.118	0.273	0.461	0.928	1.092	1.273	2.226
5	Adv.	7827.0	0.961	0.227	0.294	0.802	0.947	1.105	2.195
	Orig.	794.0	1.112	0.276	0.324	0.921	1.098	1.283	2.084
6	Adv.	7927.0	1.081	0.251	0.364	0.902	1.062	1.239	2.476
	Orig.	970.0	1.224	0.280	0.545	1.019	1.203	1.401	2.288
7	Adv.	7623.0	1.034	0.239	0.300	0.867	1.009	1.174	2.164
	Orig.	904.0	1.200	0.239	0.454	1.038	1.177	1.346	2.172
8	Adv.	7386.0	1.003	0.227	0.397	0.843	0.981	1.140	2.034
	Orig.	943.0	1.322	0.273	0.562	1.129	1.311	1.488	2.492
9	Adv.	7727.0	1.071	0.236	0.379	0.904	1.052	1.215	2.226
	Orig.	942.0	1.175	0.233	0.507	1.017	1.165	1.315	2.112

Tabelle 24.: Auswertung des Durchschnitts der Relevance Scores je Saliency Maps für Vanilla Gradient

Label	Typ	Anzahl	Mean	Std.	Min.
0	Adv.	7391.0	0.000844	0.001451	9.839105e-08
Orig.	897.0	0.001123	0.001006	2.003512e-07	0.000431
1	Adv.	7678.0	0.001285	0.002350	1.100229e-07
Orig.	962.0	0.004290	0.003788	2.591200e-06	0.001649
2	Adv.	7896.0	0.000987	0.000844	3.151826e-08
Orig.	858.0	0.001260	0.008136	1.575859e-07	0.000273
3	Adv.	8118.0	0.000656	0.000818	1.871729e-07
Orig.	727.0	0.000433	0.000470	4.853087e-07	0.000115
4	Adv.	7825.0	0.000983	0.002659	1.638243e-06
Orig.	905.0	0.000664	0.000836	2.664622e-07	0.000156
5	Adv.	7827.0	0.001058	0.002375	2.506585e-06
Orig.	794.0	0.000933	0.000994	3.455096e-07	0.000288
6	Adv.	7927.0	0.000749	0.001031	4.564996e-06
Orig.	970.0	0.000943	0.001096	1.298008e-06	0.000319
7	Adv.	7623.0	0.000878	0.001175	4.248185e-07
Orig.	904.0	0.002056	0.003243	8.053618e-07	0.000649
8	Adv.	7386.0	0.001177	0.008406	2.305376e-06
Orig.	943.0	0.001640	0.001674	2.353015e-07	0.000576
9	Adv.	7727.0	0.000989	0.001795	1.162878e-07
Orig.	942.0	0.002448	0.002084	1.907041e-07	0.000963

Tabelle 25.: Auswertung des Durchschnitts der Relevance Scores je Saliency Maps für LRP $\alpha = 1$

Label	Typ	1. Q	Median	3. Q	Max.
0	Adv.	0.000318	0.000611	0.001084	0.091680
Orig.	897.0	0.000872	0.001561	0.009692	0.010
1	Adv.	0.000447	0.000836	0.001545	0.119691
Orig.	962.0	0.003254	0.005818	0.029762	0.030
2	Adv.	0.000434	0.000782	0.001320	0.017958
Orig.	858.0	0.000631	0.001297	0.237015	0.237
3	Adv.	0.000264	0.000489	0.000845	0.040944
Orig.	727.0	0.000274	0.000557	0.003653	0.004
4	Adv.	0.000381	0.000707	0.001215	0.197906
Orig.	905.0	0.000388	0.000841	0.006754	0.007
5	Adv.	0.000399	0.000745	0.001320	0.155827
Orig.	794.0	0.000619	0.001241	0.012710	0.013
6	Adv.	0.000316	0.000574	0.000953	0.065329
Orig.	970.0	0.000631	0.001238	0.017540	0.018
7	Adv.	0.000364	0.000654	0.001130	0.063197
Orig.	904.0	0.001386	0.002612	0.069745	0.070
8	Adv.	0.000442	0.000805	0.001390	0.717437
Orig.	943.0	0.001164	0.002218	0.023254	0.023
9	Adv.	0.000326	0.000618	0.001149	0.108486
Orig.	942.0	0.001886	0.003362	0.015124	0.015

Tabelle 26.: Auswertung des Durchschnitts der Relevance Scores je Saliency Maps für LRP $\alpha = 1$ Quantile

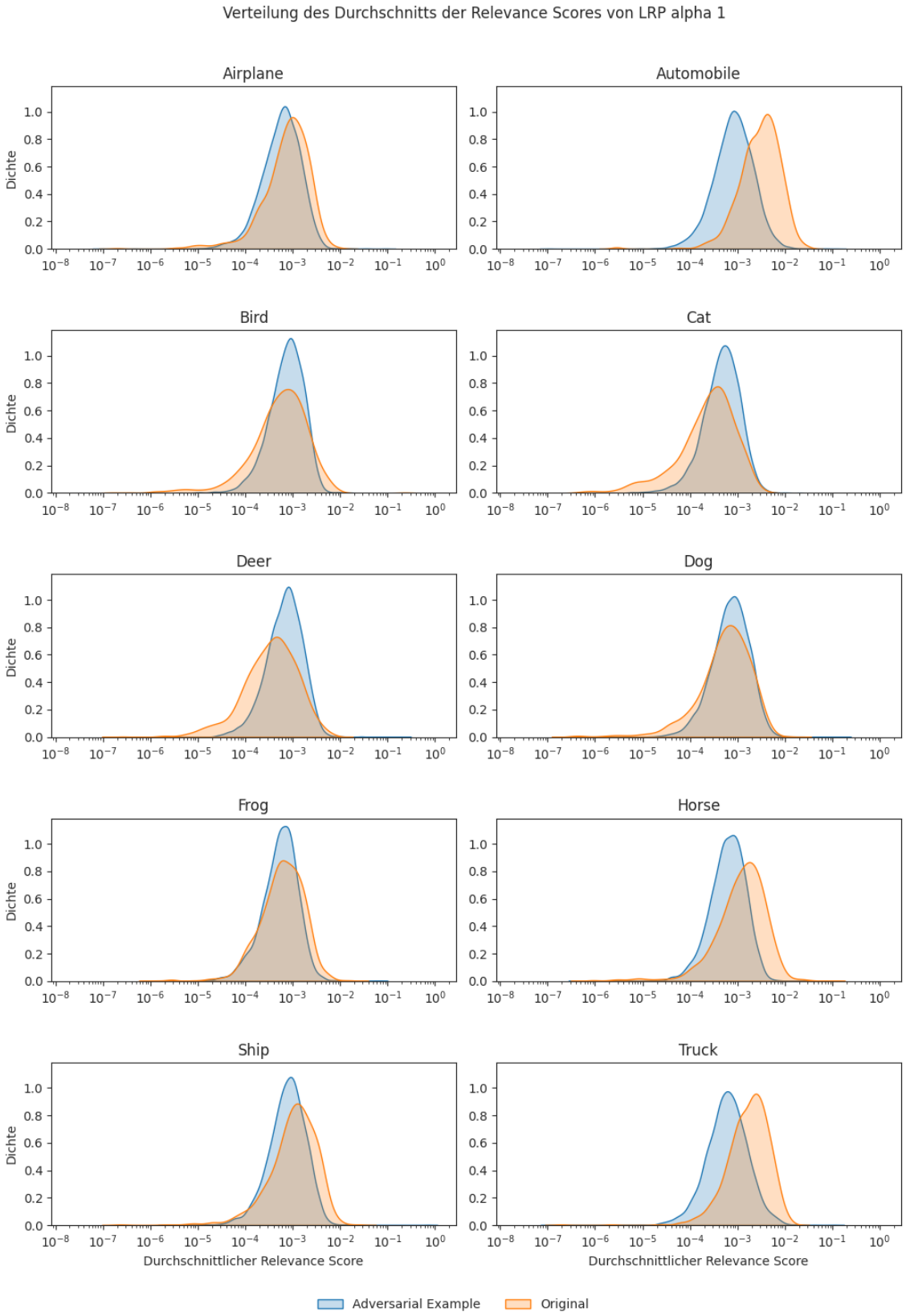


Abbildung 33.: Verteilung des Durchschnitts der Relevance Scores für LRP $\alpha = 1$

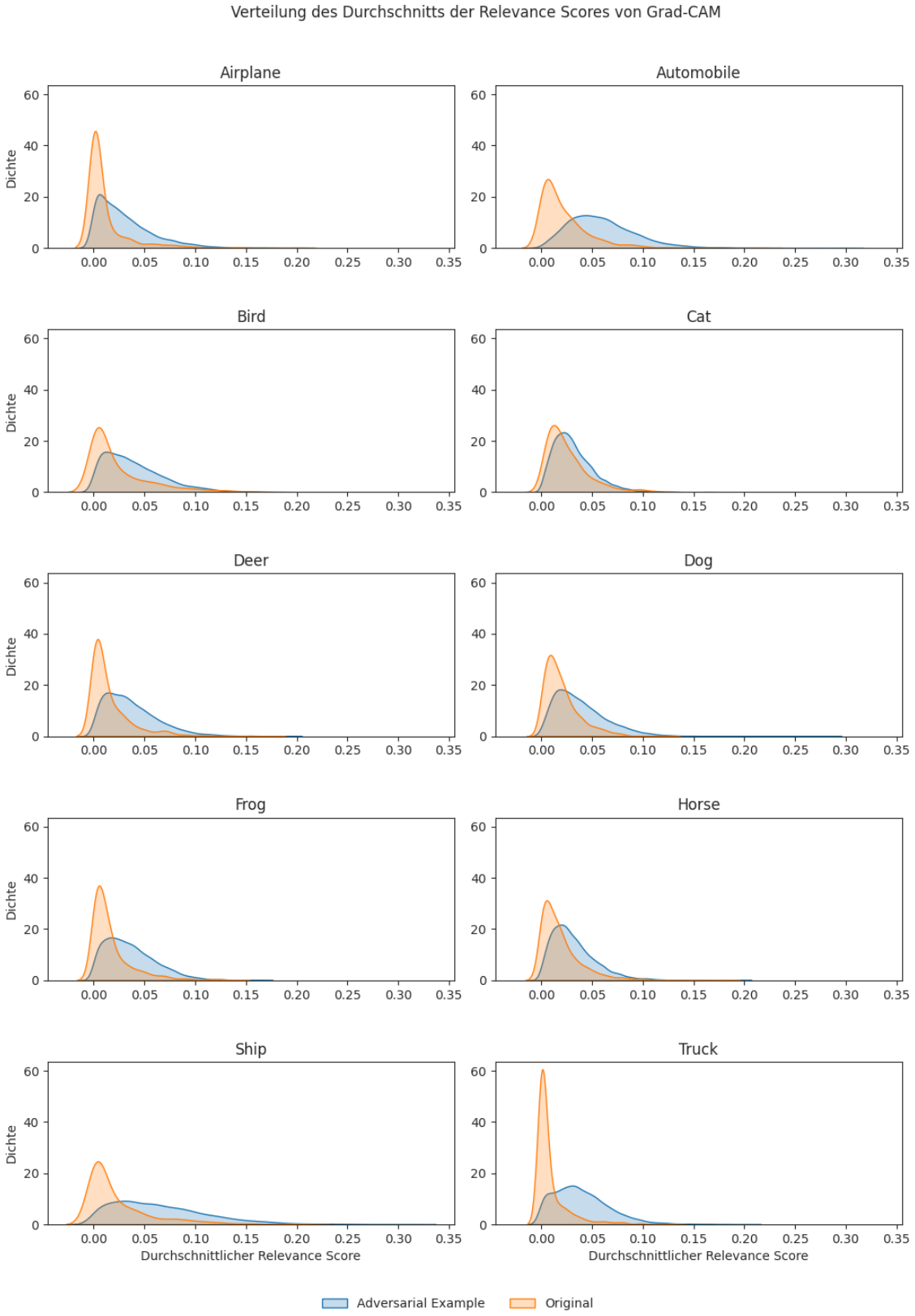


Abbildung 34.: Verteilung des Durchschnitts der Relevance Scores für Grad-CAM

Label	Typ	Anzahl	Mean	Std.	Min.
0	Adv.	7391.0	1.279585e+19	3.708818e+19	1.550630e+15
	Orig.	897.0	1.556201e+19	2.559961e+19	4.864036e+15
1	Adv.	7678.0	2.951471e+19	7.852207e+19	2.156891e+15
	Orig.	962.0	1.459892e+20	2.491328e+20	1.143562e+17
2	Adv.	7896.0	1.832954e+19	4.474450e+19	7.714999e+14
	Orig.	858.0	2.528230e+19	1.379201e+20	6.840595e+14
3	Adv.	8118.0	1.300066e+19	1.739227e+20	2.543328e+15
	Orig.	727.0	5.619441e+18	9.201398e+18	3.608047e+15
4	Adv.	7825.0	1.720555e+19	6.302332e+19	1.575338e+16
	Orig.	905.0	1.321578e+19	3.900987e+19	6.328972e+15
5	Adv.	7827.0	2.466512e+19	5.486443e+19	9.905977e+16
	Orig.	794.0	1.549653e+19	2.840622e+19	1.057730e+16
6	Adv.	7927.0	8.077442e+18	1.554575e+19	7.889473e+16
	Orig.	970.0	1.172508e+19	1.895652e+19	3.534865e+16
7	Adv.	7623.0	1.324467e+19	1.189792e+20	1.013527e+17
	Orig.	904.0	5.350248e+19	2.699583e+20	1.183722e+16
8	Adv.	7386.0	1.992752e+19	3.373365e+20	8.702413e+15
	Orig.	943.0	2.756887e+19	4.230759e+19	4.517149e+15
9	Adv.	7727.0	2.102546e+19	6.479950e+19	2.832994e+16
	Orig.	942.0	5.425397e+19	7.990574e+19	1.652300e+15

Tabelle 27.: Auswertung des Durchschnitts der Relevance Scores je Saliency Maps für LRP $\alpha = 2$

Label	Typ	1. Q	Median	3. Q	Max.
0	Adv.	2.787492e+18	5.970587e+18	1.315869e+19	2.360136e+21
	Orig.	3.295779e+18	7.471890e+18	1.698964e+19	2.432160e+20
1	Adv.	6.070497e+18	1.252381e+19	2.805281e+19	3.403182e+21
	Orig.	2.740224e+19	6.640290e+19	1.733095e+20	3.785169e+21
2	Adv.	4.418782e+18	9.268106e+18	2.018097e+19	2.856573e+21
	Orig.	3.056534e+18	8.501593e+18	2.184869e+19	3.879172e+21
3	Adv.	2.579018e+18	5.573532e+18	1.142034e+19	1.548646e+22
	Orig.	8.373758e+17	2.313384e+18	6.157527e+18	9.014572e+19
4	Adv.	3.795259e+18	8.619356e+18	1.782549e+19	3.425387e+21
	Orig.	1.525855e+18	4.340190e+18	1.166796e+19	6.919806e+20
5	Adv.	5.071781e+18	1.122955e+19	2.614779e+19	1.948269e+21
	Orig.	2.458406e+18	6.640845e+18	1.663128e+19	3.444656e+20
6	Adv.	2.269632e+18	4.537510e+18	8.713718e+18	5.442255e+20
	Orig.	2.646669e+18	6.074563e+18	1.283631e+19	2.381783e+20
7	Adv.	3.590409e+18	6.911655e+18	1.348513e+19	1.025055e+22
	Orig.	6.916759e+18	1.872500e+19	5.067163e+19	7.783443e+21
8	Adv.	4.269841e+18	8.626126e+18	1.733516e+19	2.878625e+22
	Orig.	5.189910e+18	1.236823e+19	3.303379e+19	4.536893e+20
9	Adv.	4.029664e+18	8.872660e+18	1.979636e+19	3.325449e+21
	Orig.	1.242335e+19	2.871577e+19	6.160447e+19	8.977310e+20

Tabelle 28.: Auswertung des Durchschnitts der Relevance Scores je Saliency Maps für LRP $\alpha = 2$ Quantile

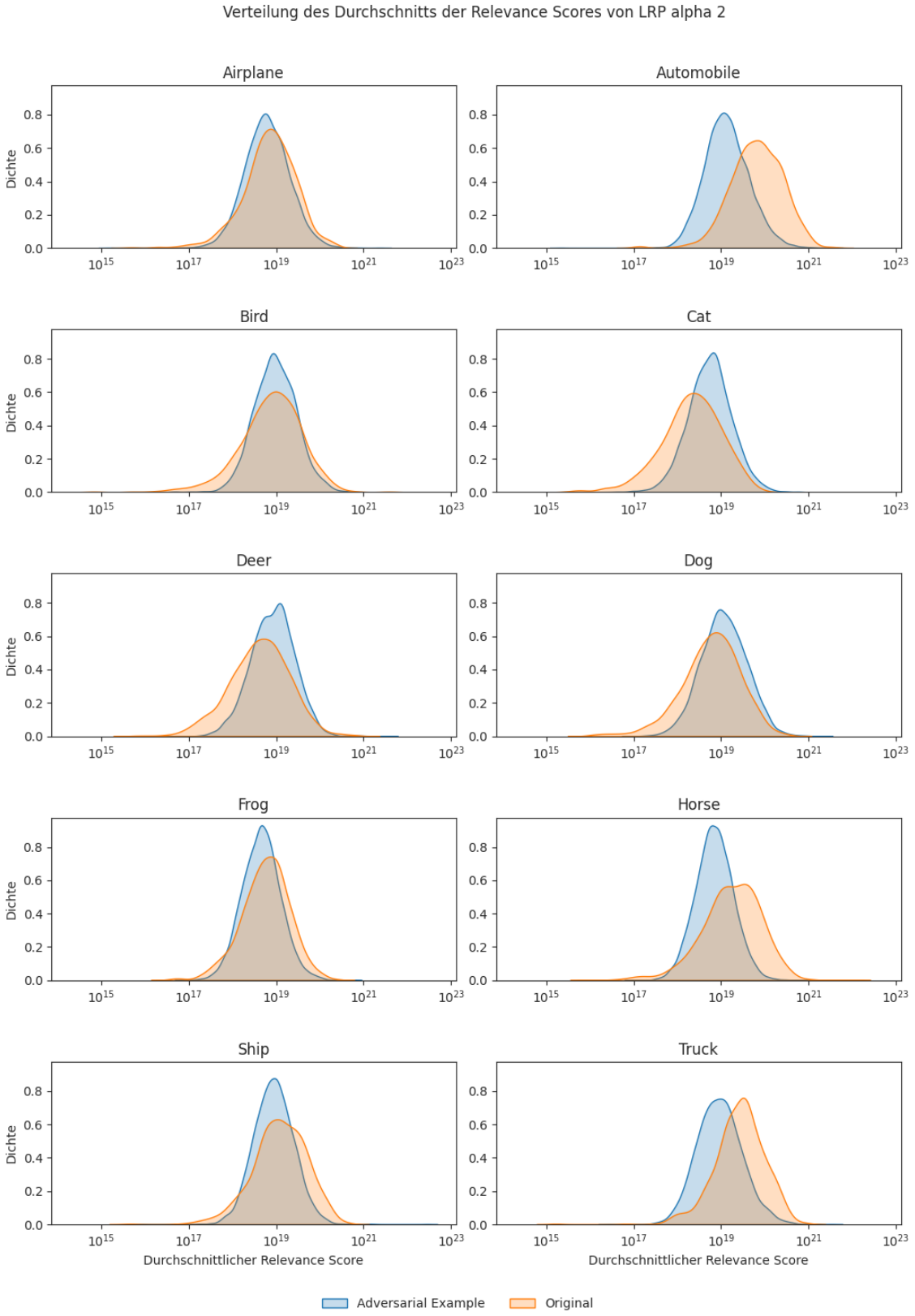


Abbildung 35.: Verteilung des Durchschnitts der Relevance Scores für LRP $\alpha = 2$

Label	Typ	Anzahl	Mean	Std.	Min.
0	Adv.	7368.0	0.000002	0.000004	3.190983e-09
	Orig.	895.0	0.000003	0.000005	2.740148e-09
1	Adv.	7654.0	0.000006	0.000009	6.715149e-09
	Orig.	959.0	0.000028	0.000039	1.178348e-07
2	Adv.	7858.0	0.000004	0.000006	1.406749e-09
	Orig.	854.0	0.000004	0.000007	7.645064e-10
3	Adv.	8080.0	0.000002	0.000003	1.005217e-09
	Orig.	724.0	0.000001	0.000002	3.580097e-09
4	Adv.	7798.0	0.000004	0.000006	7.556174e-09
	Orig.	903.0	0.000003	0.000006	4.193424e-09
5	Adv.	7800.0	0.000006	0.000010	1.646623e-08
	Orig.	794.0	0.000004	0.000007	7.302940e-09
6	Adv.	7891.0	0.000002	0.000003	8.119950e-09
	Orig.	965.0	0.000002	0.000003	2.111116e-08
7	Adv.	7600.0	0.000003	0.000004	1.578403e-08
	Orig.	900.0	0.000010	0.000014	7.062900e-09
8	Adv.	7353.0	0.000003	0.000005	1.440326e-08
	Orig.	940.0	0.000006	0.000009	5.115930e-09
9	Adv.	7705.0	0.000004	0.000007	2.203908e-08
	Orig.	937.0	0.000010	0.000013	6.775821e-09

Tabelle 29.: Auswertung des Durchschnitts der Relevance Scores je Saliency Maps für LRP $\varepsilon = 1$

Label	Typ	1. Q	Median	3. Q	Max.
0	Adv.	5.568581e-07	1.253071e-06	0.000003	0.000069
	Orig.	6.795511e-07	1.782676e-06	0.000004	0.000066
1	Adv.	1.285238e-06	2.786640e-06	0.000006	0.000194
	Orig.	5.579321e-06	1.417770e-05	0.000036	0.000592
2	Adv.	9.203295e-07	2.074672e-06	0.000005	0.000157
	Orig.	5.829836e-07	1.593842e-06	0.000004	0.000065
3	Adv.	4.957634e-07	1.101323e-06	0.000002	0.000053
	Orig.	1.893840e-07	5.563390e-07	0.000001	0.000017
4	Adv.	8.849598e-07	2.080950e-06	0.000005	0.000208
	Orig.	3.082002e-07	8.631956e-07	0.000002	0.000083
5	Adv.	1.224052e-06	3.030383e-06	0.000007	0.000238
	Orig.	6.662547e-07	1.760217e-06	0.000005	0.000069
6	Adv.	5.279452e-07	1.143703e-06	0.000002	0.000044
	Orig.	4.895472e-07	1.160646e-06	0.000003	0.000045
7	Adv.	7.572792e-07	1.594349e-06	0.000003	0.000063
	Orig.	1.809487e-06	4.702061e-06	0.000012	0.000134
8	Adv.	8.734614e-07	1.935915e-06	0.000004	0.000079
	Orig.	1.049752e-06	3.024779e-06	0.000008	0.000077
9	Adv.	7.670324e-07	1.671198e-06	0.000004	0.000212
	Orig.	2.448532e-06	5.518640e-06	0.000012	0.000132

Tabelle 30.: Auswertung des Durchschnitts der Relevance Scores je Saliency Maps für LRP $\varepsilon = 1$ Quantile

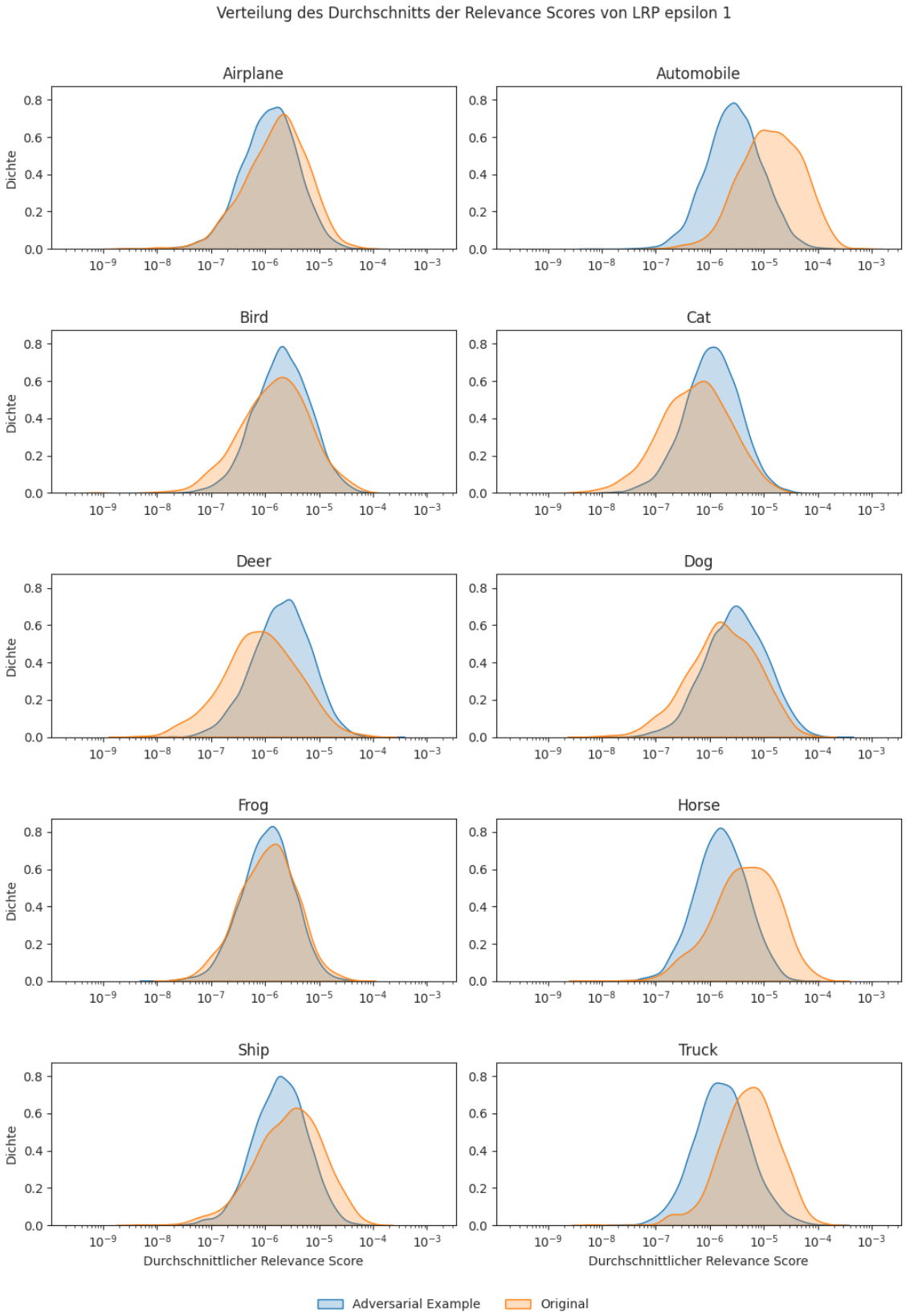


Abbildung 36.: Verteilung des Durchschnitts der Relevance Scores für LRP $\varepsilon = 1$

Literaturverzeichnis

[Aba+15] Martín Abadi u. a. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems*. URL: www.tensorflow.org.. Letzter Zugriff am: 11.07.2024.

[Ade+18] Julius Adebayo u. a. "Sanity Checks for Saliency Maps". In: *Advances in Neural Information Processing Systems*. Curran Associates, Inc., Okt. 2018. DOI: https://doi.org/10.48550/arXiv.1810.03292. URL: http://arxiv.org/abs/1810.03292.

[Agg18] Charu C. Aggarwal. *Neural Networks and Deep Learning*. Cham: Springer International Publishing, 2018. ISBN: 978-3-319-94462-3. DOI: 10.1007/978-3-319-94463-0. URL: http://link.springer.com/10.1007/978-3-319-94463-0.

[Ahm+20] Kazi Ahmed Asif Fuad u. a. "Features Understanding in 3D CNNs for Actions Recognition in Video". In: *2020 Tenth International Conference on Image Processing Theory, Tools and Applications (IPTA)*. 2020, S. 1–6. DOI: 10.1109/IPTA50016.2020.9286629.

[Alb+19] Maximilian Alber u. a. "iNNvestigate Neural Networks!" In: *Journal of Machine Learning Research* 20 (2019), S. 1–8. URL: http://jmlr.org/papers/v20/18-540.html.

[Anc+17] Marco Ancona u. a. "Towards better understanding of gradient-based attribution methods for Deep Neural Networks". In: *International Conference on Learning Representations*. Nov. 2017. DOI: 10.48550. URL: http://arxiv.org/abs/1711.06104.

[Bac+15] Sebastian Bach u. a. "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation". In: *PLoS ONE* 10.7 (Juli 2015). ISSN: 19326203. DOI: 10.1371/journal.pone.0130140.

[Bae+10] David Baehrens u. a. *How to Explain Individual Classification Decisions*. Techn. Ber. 2010, S. 1803–1831.

[Bis09] Christopher M. Bishop. *Pattern recognition and machine learning*. 8th printing. New York NY: Springer, 2009. ISBN: 9780387310732.

[Bou+22] Luca Bourroux u. a. "Multi Layered Feature Explanation Method for Convolutional Neural Networks". In: *Pattern Recognition and Artificial Intelligence*. Hrsg. von Mounîm El Yacoubi u. a. Cham: Springer International Publishing, 2022, S. 603–614. ISBN: 978-3-031-09037-0.

[BT94] Herbert Büning und Götz Trenkler. *Nichtparametrische statistische Methoden*. Berlin: de Gruyter, 1994.

[Cha+17] Supriyo Chakraborty u. a. "Interpretability of deep learning models: A survey of results". In: *2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*. IEEE, Aug. 2017, S. 1–6. ISBN: 978-1-5386-0435-9. DOI: 10.1109/UIC-ATC.2017.8397411. URL: https://ieeexplore.ieee.org/document/8397411/.

[COI] Inc. COIN-OR Foundation. *COIN|OR Computational Infrastructure for Operations Research*. URL: https://www.coin-or.org/. Letzter Zugriff am: 03.07.2024.

[Dem87] Eugene Demidenko. *Mixed Models*. Wiley, Jan. 1987. ISBN: 9781118091579. DOI: 10.1002/9781118651537.

[Die20] Tamara Regina Dieter. "Analysis of Adversarial Examples with Layer-wise Relevance Propagation". Masterthesis. Darmstadt: Hochschule Darmstadt, 2020.

[DZ23a] Tamara R. Dieter und Horst Zisgen. "Evaluation of the Explanatory Power Of Layer-wise Relevance Propagation using Adversarial Examples". In: *Neural Processing Letters* 55.7 (Dez. 2023), S. 8531–8550. ISSN: 1573773X. DOI: 10.1007/s11063-023-11166-8.

[DZ23b] Tamara R. Dieter und Horst Zisgen. *Tools for Evaluating the Explainatory Power of LRP*. URL: https://github.com/tamaradi/Evaluation-of-the-Explainatory-Power-of-LRP%20https://doi.org/10.5281/zenodo.7498422. Letzter Zugriff am: 01.07.2024. DOI: 10.5281/zenodo.7498422.

[Erh+09] Dumitru Erhan u. a. *Visualizing Higher-Layer Features of a Deep Network*. Techn. Ber. Université de Montréal, 2009. URL: https://www.researchgate.net/publication/265022827.

[Eur24] Europäische Kommission. *Verordnung (EU) 2024/1689 des Europäischen Parlaments und des Rates zur Festlegung harmonisierter Vorschriften für künstliche Intelligenz und zur Änderung der Verordnungen (EG) Nr. 300/2008, (EU) Nr. 167/2013, (EU) Nr. 168/2013, (EU) 2018/858, (EU) 2018/1139 und (EU) 2019/2144 sowie der Richtlinien 2014/90/EU, (EU) 2016/797 und (EU) 2020/1828 (Verordnung über künstliche Intelligenz)*. URL: https://eur-lex.europa.eu/eli/reg/2024/1689/oj?locale=de. Letzter Zugriff am:13.07.2024.

[Fah+23] Ludwig Fahrmeir u. a. *Statistik*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2023. ISBN: 978-3-662-67525-0. DOI: 10.1007/978-3-662-67526-7. URL: https://link.springer.com/10.1007/978-3-662-67526-7.

[Fla+21] Rémi Flamary u. a. “POT: Python Optimal Transport”. In: *Journal of Machine Learning Research* 22 (2021), S. 1–8. URL: http://jmlr.org/papers/v22/20-451.html..

[GAZ19] Amirata Ghorbani, Abubakar Abid und James Zou. “Interpretation of Neural Networks Is Fragile”. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 33.01 (Juli 2019), S. 3681–3688. ISSN: 2374-3468. DOI: 10.1609/aaai.v33i01.33013681. URL: https://ojs.aaai.org/index.php/AAAI/article/view/4252.

[GBC16] Ian Goodfellow, Yoshua Bengio und Aaron Courville. *Deep learning*. Adaptive computation and machine learning. Cambridge, Massachusetts, 2016. ISBN: 9780262035613. URL: http://scans.hebis.de/HEBCGI/show.pl?38727953_toc.pdf.

[GSS14] Ian J. Goodfellow, Jonathon Shlens und Christian Szegedy. “Explaining and Harnessing Adversarial Examples”. In: *CoRR* (Dez. 2014). URL: http://arxiv.org/abs/1412.6572.

[Gun+19] David Gunning u. a. “XAI-Explainable artificial intelligence”. In: *Science Robotics* 4.37 (Dez. 2019). ISSN: 24709476. DOI: 10.1126/scirobotics.aay7120.

[GUR] GUROBI OPTIMIZATION LLC. *Gurobi Optimizer*. URL: https://www.gurobi.com/. Letzter Zugriff am: 03.07.2024.

[Hit41] Frank L. Hitchcock. “The Distribution of a Product from Several Sources to Numerous Localities”. In: *Journal of Mathematics and Physics* 20.1-4 (Apr. 1941), S. 224–230. ISSN: 0097-1421. DOI: 10.1002/sapm1941201224. URL: https://onlinelibrary.wiley.com/doi/10.1002/sapm1941201224.

[HJM19] Juyeon Heo, Sunghwan Joo und Taesup Moon. “Fooling Neural Network Interpretations via Adversarial Model Manipulation”. In: *NeurIPS* (Feb. 2019). URL: http://arxiv.org/abs/1902.02041.

[Kol+17] Soheil Kolouri u. a. “Optimal Mass Transport: Signal processing and machine-learning applications”. In: *IEEE Signal Processing Magazine* 34.4 (Juli 2017), S. 43–59. ISSN: 10535888. DOI: 10.1109/MSP.2017.2695801.

[Kri09] Alex Krizhevsky. *The CIFAR-10 dataset*. URL: https://www.cs.toronto.edu/~kriz/cifar.html. Letzter Zugriff am: 12.07.2024.

[LBH98] Y. Lavin, R. Batra und L. Hesselink. “Feature comparisons of vector fields using Earth mover’s distance”. In: *Proceedings Visualization ’98 (Cat. No.98CB36276)*. IEEE, 1998, S. 103–109. ISBN: 0-8186-9176-X. DOI: 10.1109/VISUAL.1998.745291. URL: http://ieeexplore.ieee.org/document/745291/.

[Le+23] Dung Le u. a. “Fast Approximation of the Generalized Sliced-Wasserstein Distance”. In: *ICML Workshop on New Frontiers in Learning, Control, and Dynamical Systems*. Okt. 2023. URL: http://arxiv.org/abs/2210.10268.

[Mol22] Christoph Molnar. *Interpretable Machine Learning*. 2. Aufl. 2022. URL: https://christophm.github.io/interpretable-ml-book/.

[MOS] MOSEK ApS. *MOSEK*. URL: https://www.mosek.com/. Letzter Zugriff am: 03.07.2024.

[Nad+21] Kimia Nadjahi u. a. “Fast Approximation of the Sliced-Wasserstein Distance Using Concentration of Random Projections”. In: *Neural Information Processing Systems*. Juni 2021. URL: http://arxiv.org/abs/2106.15427.

[PC19] Gabriel Peyré und Marco Cuturi. “Computational Optimal Transport”. In: *Foundations and Trends in Machine Learning* 11 (2019), S. 355–607. URL: http://arxiv.org/abs/1803.00567.

[Rab+11] Julien Rabin u. a. “Wasserstein Barycenter and Its Application to Texture Mixing”. In: *Scale Space and Variational Methods in Computer Vision*. 2011, S. 435–446. DOI: 10.1007/978-3-642-24785-9{_}37. URL: http://link.springer.com/10.1007/978-3-642-24785-9_37.

[RGC15] Aaditya Ramdas, Nicolas Garcia und Marco Cuturi. “On Wasserstein Two Sample Testing and Related Families of Nonparametric Tests”. In: *Entropy* 19 (Sep. 2015). URL: http://arxiv.org/abs/1509.02237.

[RKV20] Matthias Rosynski, Frank Kirchner und Matias Valdenegro-Toro. “Are Gradient-based Saliency Maps Useful in Deep Reinforcement Learning?” In: *CoRR* (Dez. 2020). URL: http://arxiv.org/abs/2012.01281.

[ROF92] Leonid I Rudin, Stanley Osher und Emad Fatemi. *Nonlinear total variation based noise removal algorithms*. Techn. Ber. 1992, S. 259–268.

[RTG00] Yossi Rubner, Carlo Tomasi und Leonidas J Guibas. “The Earth Mover’s Distance as a Metric for Image Retrieval”. In: *International Journal of Computer Vision 40* (2000), S. 99–121. DOI: 10.1023/A:1026543900054.

[RTG98] Y. Rubner, C. Tomasi und L.J. Guibas. “A metric for distributions with applications to image databases”. In: *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*. Narosa Publishing House, 1998, S. 59–66. ISBN: 81-7319-221-9. DOI: 10.1109/ICCV.1998.710701. URL: http://ieeexplore.ieee.org/document/710701/.

[Sam+19] Wojciech Samek u. a. *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. Hrsg. von Wojciech Samek u. a. Bd. 11700. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2019. ISBN: 978-3-030-28953-9. DOI: 10.1007/978-3-030-28954-6. URL: http://link.springer.com/10.1007/978-3-030-28954-6.

[Sel+16a] Ramprasaath R Selvaraju u. a. “Grad-CAM: Why did you say that?” In: *ArXiv* abs/1611.07450 (Nov. 2016). URL: http://arxiv.org/abs/1611.07450.

[Sel+16b] Ramprasaath R. Selvaraju u. a. “Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization”. In: *International Journal of Computer Vision* (Okt. 2016), S. 336–359. DOI: 10.1007/s11263-019-01228-7. URL: http://arxiv.org/abs/1610.02391%20http://dx.doi.org/10.1007/s11263-019-01228-7.

[Ser+22] Mathieu Serrurier u. a. “On the explainable properties of 1-Lipschitz Neural Networks: An Optimal Transport Perspective”. In: *Neural Information Processing Systems* (Juni 2022). URL: http://arxiv.org/abs/2206.06854.

[SPV18] Alexandru Constantin Serban, Erik Poll und Joost Visser. *Adversarial Examples - A Complete Characterisation of the Phenomenon*. Techn. Ber. Radbound University, Okt. 2018. URL: http://arxiv.org/abs/1810.01185.

[SVZ13] Karen Simonyan, Andrea Vedaldi und Andrew Zisserman. “Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps”. In: *CoRR* (Dez. 2013). URL: http://arxiv.org/abs/1312.6034.

[Sze+14] Christian Szegedy u. a. *Intriguing properties of neural networks*. Techn. Ber. Google Inc., Facebook Inc., New York University, University of Montreal, Feb. 2014. URL: http://arxiv.org/abs/1312.6199.

[Tom+20] Richard Tomsett u. a. “Sanity Checks for Saliency Metrics”. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 34.04 (Apr. 2020), S. 6021–6029. ISSN: 2374-3468. DOI: 10.1609/aaai.v34i04.6064. URL: https://ojs.aaai.org/index.php/AAAI/article/view/6064.

[Vir+20] Pauli Virtanen u. a. “SciPy 1.0: fundamental algorithms for scientific computing in Python”. In: *Nature Methods* 17.3 (März 2020), S. 261–272. ISSN: 1548-7091. DOI: 10.1038/s41592-019-0686-2. URL: https://rdcu.be/b08Wh%20https://scipy.org/.

[Wan+20] Haofan Wang u. a. “Score-CAM: Score-Weighted Visual Explanations for Convolutional Neural Networks”. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2020, S. 111–119. DOI: 10.1109/CVPRW50498.2020.00020.

[Zho+15] Bolei Zhou u. a. “Learning Deep Features for Discriminative Localization”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (Dez. 2015), S. 2921–2929. URL: http://arxiv.org/abs/1512.04150.