Explainable AI

Schnuppervorlesung

Prof. Dr. Bernd Heinrich
Lehrstuhl für Wirtschaftsinformatik II
Fakultät für Informatik und Data Science

Universität Regensburg

Zentrale Fragen in diesem Termin

Warum benötigen wir Explainable AI (XAI)?

Wie können Bilder klassifizieren werden?

Wie lassen sich Bildklassifikationen besser verstehen?

Unterlagen verfügbar unter
https://github.com/URWI2/Schnuppervorlesung-XAI/

Skepsis und Befürchtungen gegenüber AI

Die Petition „***Pause Giant AI Experiments***“ forderte in 2023 eine Pause beim Training großer KI-Systeme und fand zahlreiche, auch **prominente Unterstützer** (z.B. Gründer und CEOs verschiedener IT-Unternehmen, darunter Steve Wozniak und Elon Musk sowie Professoren renommierter Universitäten im Bereich AI/Informatik).

“We call on all AI labs to immediately pause for at least 6 months the training of AI systems more powerful than GPT-4.”

“Powerful AI systems should be developed only once we are confident that their effects will be positive and their risks will be manageable.”

Quelle: https://futureoflife.org/open-letter/pause-giant-ai-experiments/

Forderungen der Petition

1) *"…jointly develop and implement a set of **shared safety protocols** for advanced AI design and development that are rigorously **audited** and **overseen** by independent outside experts"*
2) *"new and capable **regulatory authorities** dedicated to AI"*

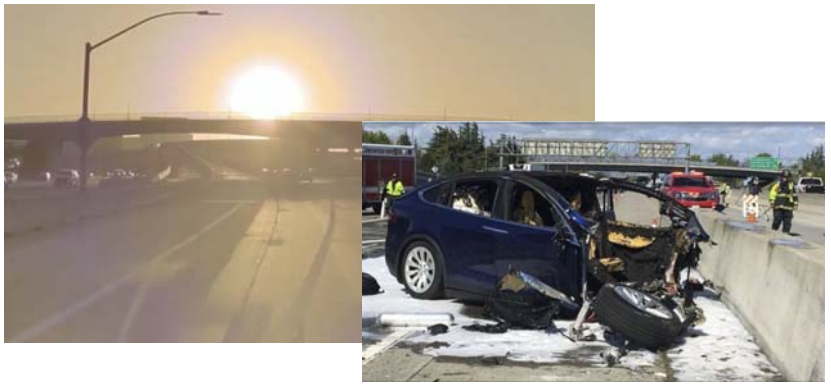
3) *"AI research and development should be refocused on making today's powerful, state-of-the-art systems more accurate, safe, **INTERPRETABLE, TRANSPARENT, ROBUST, ALIGNED, TRUSTWORTHY**, and loyal."*

Quelle: https://futureoflife.org/open-letter/pause-giant-ai-experiments/

Erklärbarkeit als gesellschaftliche Herausforderung

Fragen zur Haftbarkeit für Fehler von KI-Anwendungen

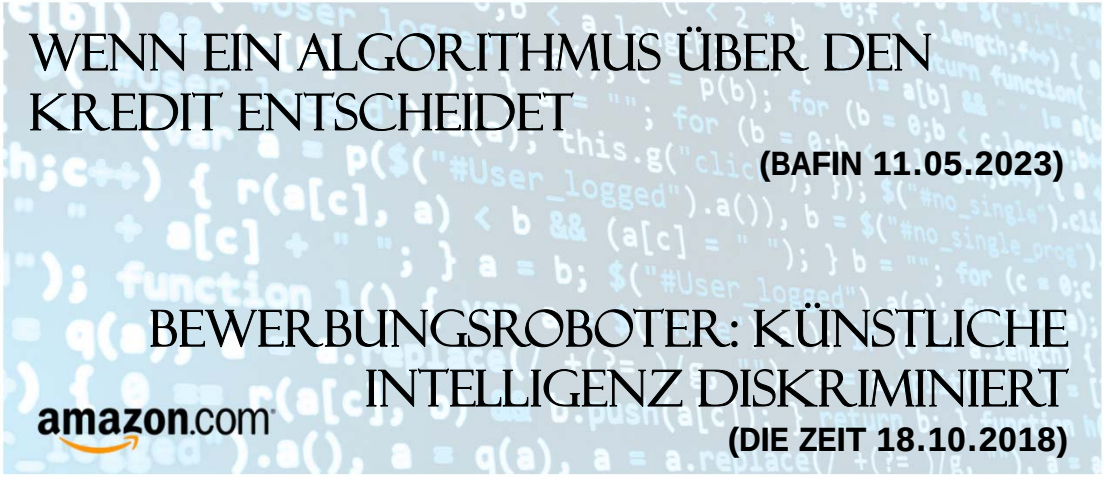
- Können Fehler und Auswirkungen festgestellt werden?
- Hätten diese Fehler von Hersteller-seite erkannt und vermieden werden müssen?



→ **Erklärbarkeit** kann zum Auffinden solcher Fehler genutzt werden

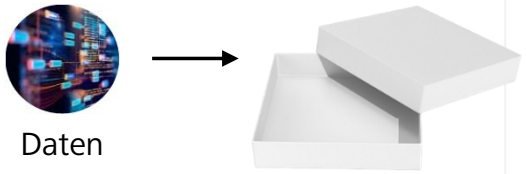
Algorithmische Diskriminierung

- **Erklärbarkeit** ermöglicht es, Entscheidungen eines AI-Systems besser zu prüfen, um z.B. **Bias** in den **Daten** erkennen zu können

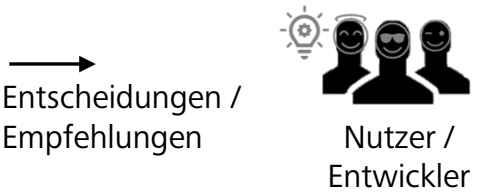


Erklärbarkeit von AI-Modellen

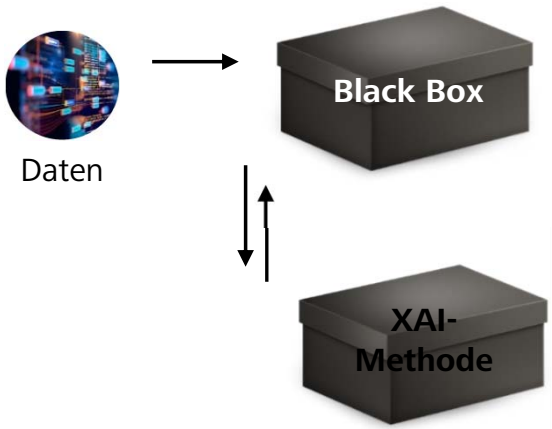
Intrinsisch



- Nutzung von für Menschen direkt **interpretierbaren ML-Modellen** („White Box“)
- Ergebnisse **verständlich** → White Box für Menschen **vollständig interpretierbar**

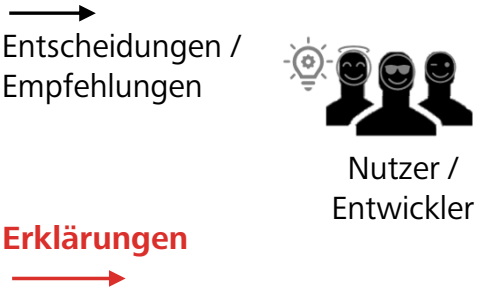


Post-Hoc



- Nutzung von für Menschen **nicht** direkt interpretierbaren ML-Modellen („Black Box“)

- **Nachträgliche Erweiterung/ Veränderung** des Modells um eine **Erklärkomponente**
- Erzeugung möglicher **Erklärungen** für Ergebnisse des ML-Modells

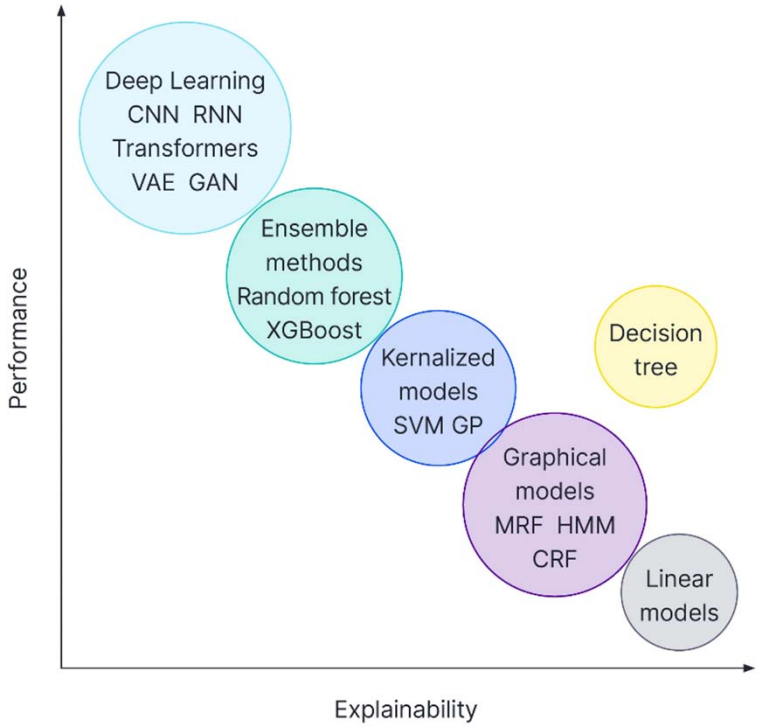


Erklärbarkeit vs. Güte von AI-Modellen

Warum werden nicht ausschließlich **White-Box-Modelle** verwendet?

- Viele **reale Problemstellungen** erfordern ML-Modelle, welche nicht direkt interpretierbar sind.

- **Trade-off** zwischen **Erklärbarkeit** und **Güte** des Modells

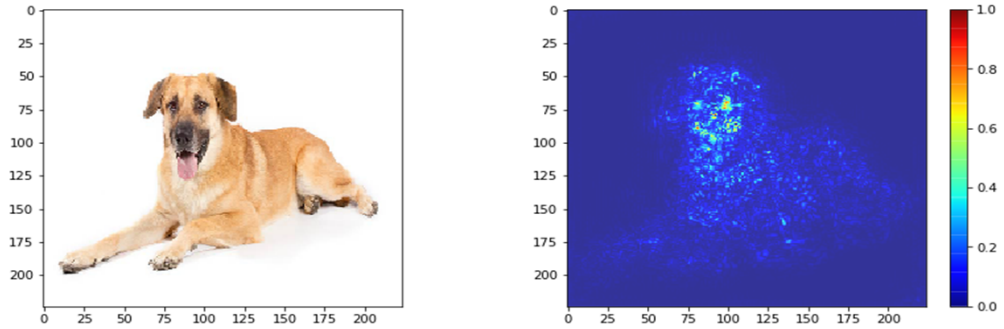


Quelle: https://aiplanet.com/learn/explainable-ai/model-interpretation-strategies/567/model-interpretation-strategies

Focus of Explanation

Erklärung Data Processing

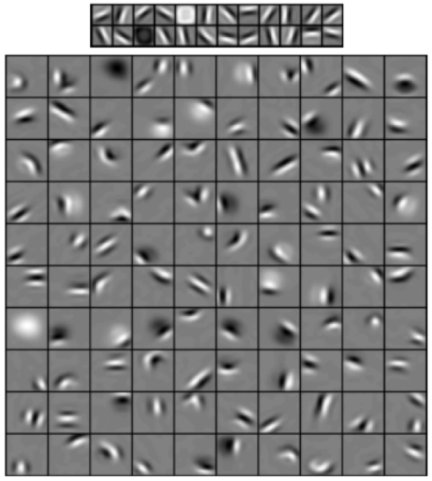
- Warum führt ein bestimmter Input zu einem bestimmten Output?
- **Transparenz über die Daten-verarbeitung im Modell**



Beispiel: Saliency Map

Erklärung Data Representation

- Welche Informationen lernt/speichert ein ML-Modell (aus den Trainingsdaten)
- **Transparenz über die Funktion einzelner Elemente des ML-Modells**
  (z.B. Parameter/Split-Attribute eines Entscheidungsbaums; Parameter/Neuronen/Schichten eines Neuronalen Netzes)



Beispiel: Learned Features

Zentrale Fragen in dieser Vorlesung

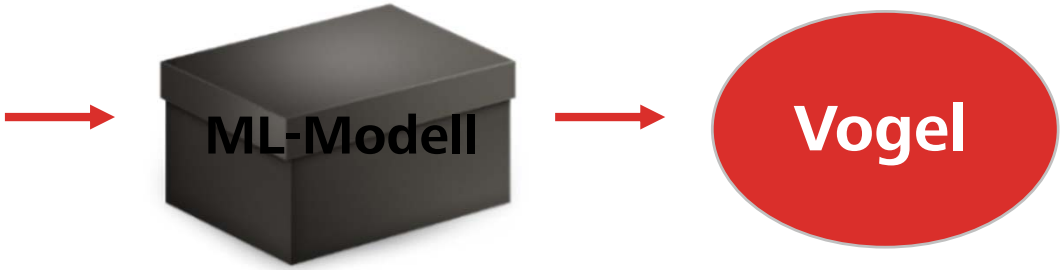
? **Warum benötigen wir Explainable AI (XAI)?**

? **Wie können Bilder klassifizieren werden?**

? **Wie lassen sich Bildklassifikationen besser verstehen?**

Bildklassifikation

Ein Machine-Learning-Modell erhält ein Bild und soll das darin gezeigte Motiv klassifizieren:

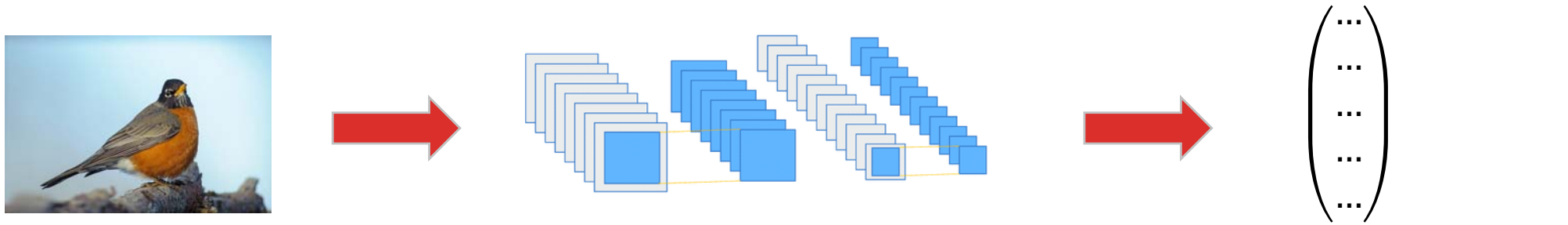


? Wie erfolgt eine solche **Prognose**?

Convolutional Neural Networks

Kurze Einführung

- **Convolutional Neural Networks (CNNs)** können zur **Klassifikation von Bildern** genutzt werden.
- Ein Bild ist ein **zweidimensionales Array**, dessen Einträge die **Werte der Pixel** (z.B. Helligkeitswerte und/oder Farbwerte/Channel) sind.
- Das Bild wird im **CNN** durch verschiedene Schichten **propagiert**
- **Erhalte** am Ende einen **Vektor**, welcher zur **Klassifikation** (z.B. Klassen Vogel, Fisch, Hund, usw.) genutzt wird.

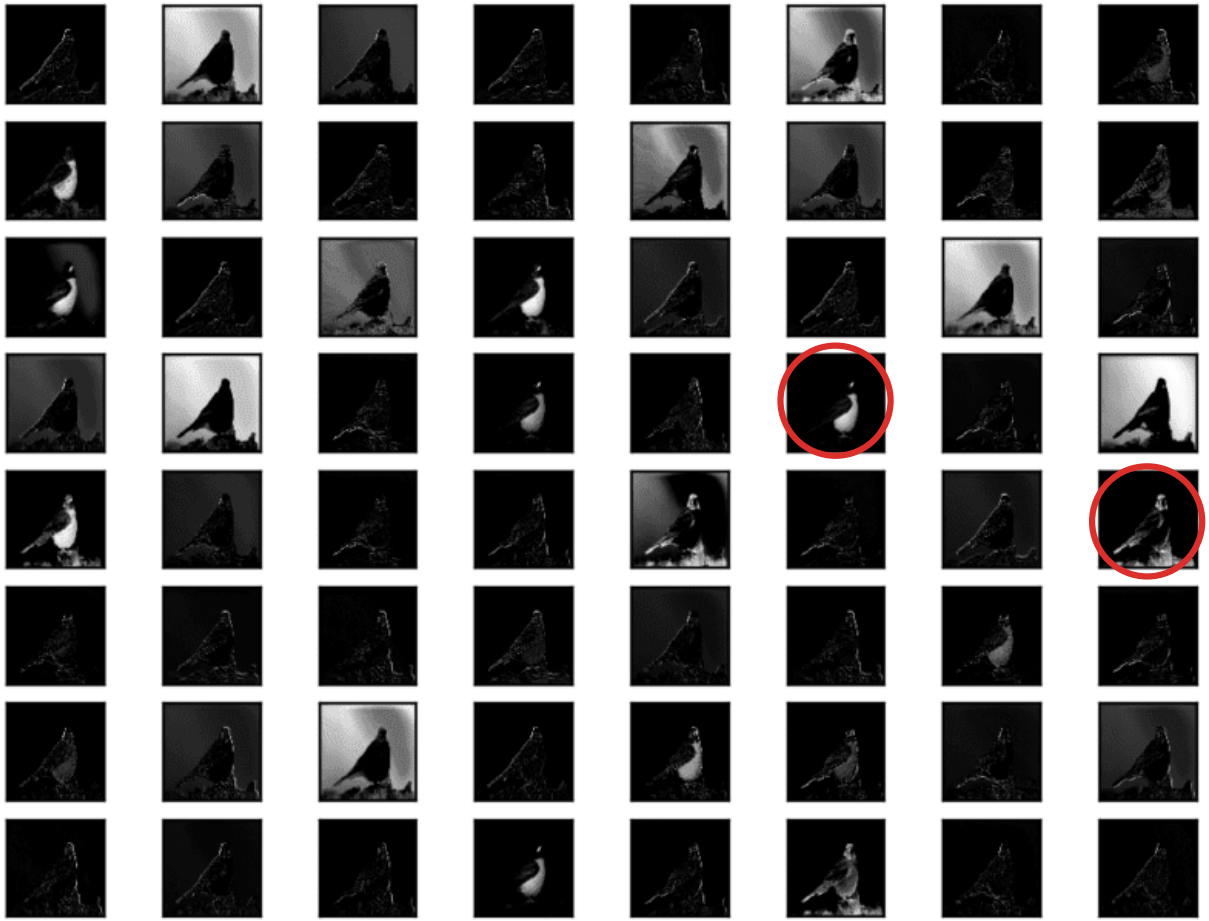


Input | Schichten mit bearbeiteten Bildern | Klassifikationsvektor

Convolutional Neural Networks

Kurze Einführung

Bilder in der <u>ersten</u> Schicht des Netzwerks:



Input

Durch Verarbeitung im CNN entstehen **mehrere Bilder** aus dem Input-Bild: **verschiedene Aspekte** des originalen Bildes werden hervorgehoben

Erkennen von **Kanten und Formen**

Convolutional Neural Networks

Kurze Einführung

- Erhalte als **Output** nach der letzten Abbildung einen **Vektor**
- Jeder Eintrag dieses Vektors ist ein „**(Punkte)Wert**“ für eine der möglichen **Klassen**
- Die Klasse mit dem **höchsten** „**(Punkte)Wert**“ ist die **prognostizierte Klasse**

$$\begin{pmatrix} s_{Klasse_1} \\ s_{Klasse\ 2} \\ s_{Klasse\ 3} \\ \dots \\ s_{Klasse\ Hund} \\ \dots \\ \dots \\ s_{Klasse\ N} \end{pmatrix}$$

Zentrale Fragen in dieser Vorlesung

? **Warum benötigen wir Explainable AI (XAI)?**

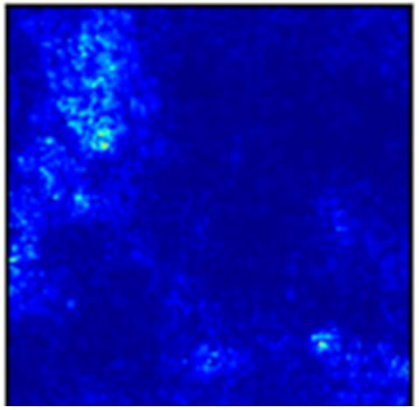
? **Wie können Bilder klassifizieren werden?**

? **Wie lassen sich Bildklassifikationen besser verstehen?**

Motivation

? Wie viel tragen die **einzelnen Pixel** des Bildes zur Modellprognose bei?

XAI-Methode



Wichtige Pixel des Bildes heller dargestellt

Pixel Attribution → dafür existieren verschiedene Verfahren!

Aufgabenstellung und Notation

- Ein Bild $I = (I_1, \dots, I_p)$ mit p Pixeln wird in ein CNN gegeben und einer Klasse $i \in \{1, \dots, n\}$ zugeteilt
- Basierend auf dem Output-Vektor $s(I) = (s_1(I), s_2(I), \dots, s_n(I))$ wird die Klasse i mit dem höchsten Eintrag $s_i(I)$ prognostiziert
- Jeder Eintrag $s_i(I)$ des Vektors hängt dabei von dem gesamten Input-Bild I ab, d.h. von jedem Pixel I_j (mit $j = 1, \dots, p$) des Bilds I
- Ein **Pixel-Attribution-Verfahren** gibt für jeden Vektoreintrag s_i einen Vektor $r_i = (r_{i1}, \dots, r_{ip})$ zurück
- Eintrag r_{ij} gibt an, welchen Einfluss der j-te Pixel des Bilds I auf den i-ten Eintrag des Output-Vektors $s(I)$ und damit für die Klasse i hat

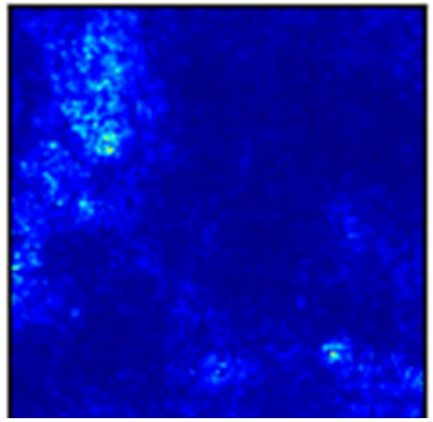
? **Wie** kann ein solcher **Vektor r_i** bestimmt werden?

Saliency Maps
Vorgehensweise

Vorgehensweise

1. Wähle eine Klasse $i \in \{1, \dots, n\}$, die betrachtet werden soll und verwende $s_i(I)$ als Outputwert der Klasse.
2. Berechne für alle Pixel I_j, $j = 1, \dots, p$, die Ableitung
$$\frac{\delta s_i(I)}{\delta I_j}$$
3. Ersetze den Wert jedes Pixels I_j durch $\frac{\delta s_i(I)}{\delta I_j}$ und zeichne das entstandene Bild. Erhalte ein Bild derselben Größe wie I, welches die Einflüsse der Pixel I_j auf $s_i(I)$ visualisiert ($\rightarrow$ **Saliency Map**)

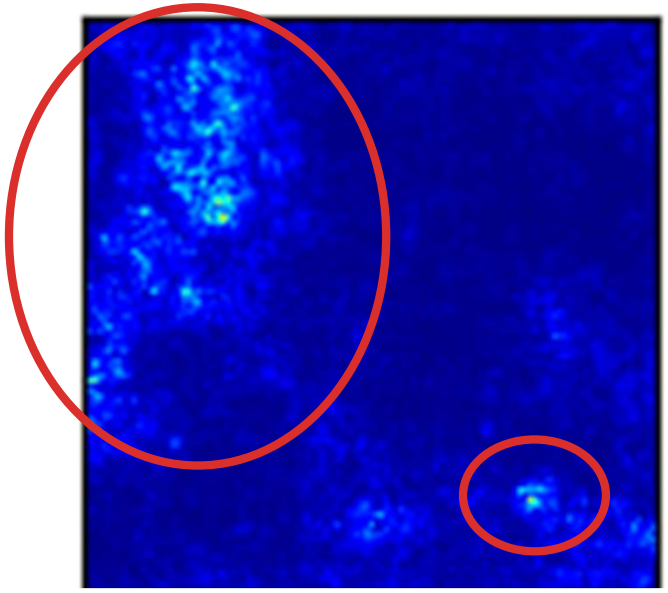
Klasse: Greyhound



Idee der Saliency Maps: Der Einfluss von I_j auf $s_i(I)$ wird durch **Wert der Ableitung** gemessen!

Interpretation Saliency Map

Vergleich zwischen **Bild** und **Saliency Map** zur **prognostizierten Klasse** „Greyhound“



Pixel, welche den Hund zeigen, waren besonders einflussreich auf die Klassifikation als Greyhound!

Hohe Gradienten auch für einzelne andere Pixel

Aufgabenstellung

Aufgabe

Laden Sie den Code unter https://github.com/URWI2/Schnuppervorlesung-XAI/ herunter. Dieser Code trainiert ein CNN auf dem häufig verwendeten **MNIST-Datensatz.** Versuchen Sie nun, die Methode **Saliency Maps** zu implementieren.

Datensatz

Der **MNIST-Datensatz** besteht aus handgeschriebenen Ziffern. Der Datensatz enthält insges. 70.000 Bilder, die jeweils in Graustufen vorliegen und eine Auflösung von 28x28 Pixeln haben. Jedes Bild zeigt eine einzelne handgeschriebene Ziffer von 0 bis 9.

Hinweis: In dem verwendeten **ML-Framework *„PyTorch“*** können Ableitungen mit der Funktion **.backward()** auf dem Output berechnet und mit **.grad.data** auf dem Input abgerufen werden.