TECHNISCHE HOCHSCHULE NÜRNBERG
GEORG SIMON OHM

informatiCup
Der GI-Wettbewerb für Studierende

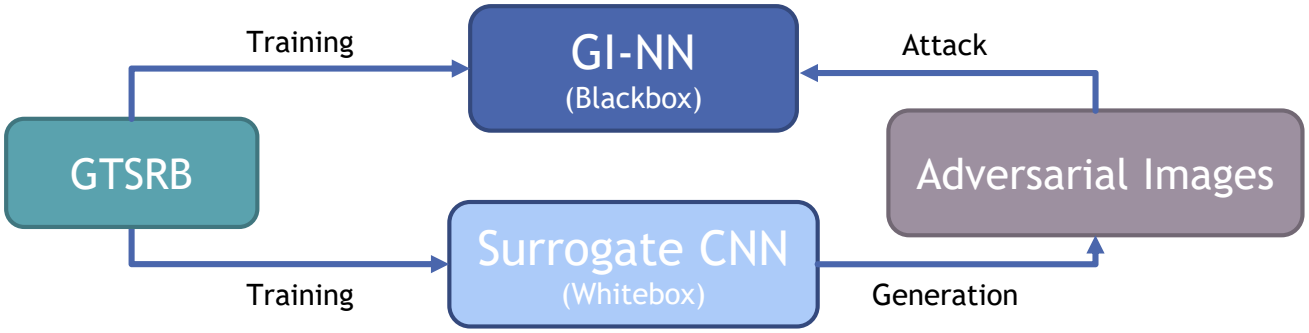
Bilderzeugende Verfahren zum Angriff einer Verkehrsschilder erkennenden KI

Leonhard Applis · Peter Bauer · Andreas Porada · Florian Stöckl

Agenda

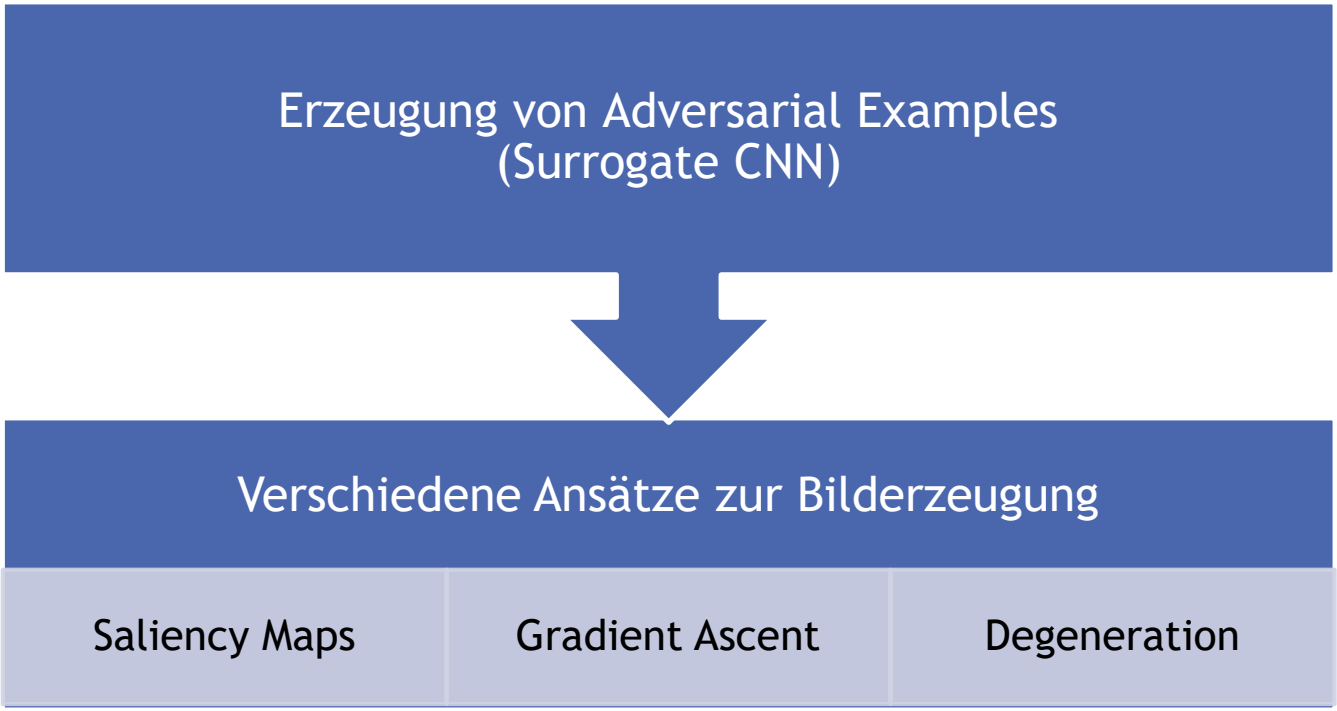
- Herausforderung
- Lösungsansätze
  - Saliency Maps
  - Gradient Ascent
  - Degeneration
- Zusammenfassung

Herausforderung



- InformatiCup:
  - Architektur des GI-NN ist unbekannt
  - Trainingsdatensatz ist bekannt (GTSRB)
- Stand der Wissenschaft:
  - Transferierbarkeit von Angriffen zwischen NNs [1]
    - Unterschiedliche NN-Architektur
    - Selber Trainingsdatensatz

Lösungsansätze



- Vorteile:
  - Direkter Zugriff auf die NN-Architektur und -Parameter
  - Umgehung der Web-Schnittstelle (Schnelligkeit)

Saliency Maps

- Topografische Darstellung von klassentypischen, markanten Bildmerkmalen (High-Level Features), die das trainierte CNN zu Eingabebildern „gelernt“ hat. [2]

Beispielbild

Saliency Map

- Hypothese:
  - Hohe Erkennungsrate bei CNNs, jedoch nur abstrakte, schemenhafte Wahrnehmung beim Menschen möglich

Saliency Maps: Implementierung [3-6]

- Ungeglättete und geglättete Verfahren
  - (Geglättete) Guided Backpropagation
  - (Geglättete) Integrated Gradient
  - (Geglättete) Vanilla

Vorverarbeitung

- Surrogat CNN „Aphrodite“: Implementierung, Training
- Bildklassifikation durch „Aphrodite“
- Bildselektion: Konfidenz = 100% (1.933/12.630)

Bilderzeugung

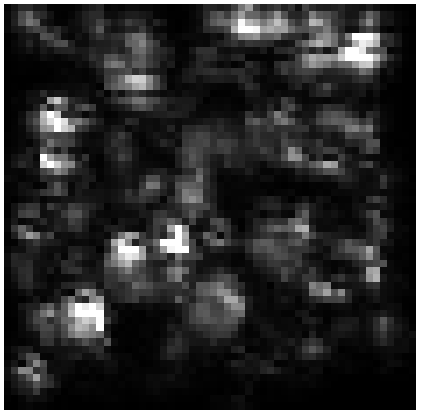
- Saliency Map:
  - Ungeglättete Verfahren
  - Geglättete Verfahren

Remote Evaluierung

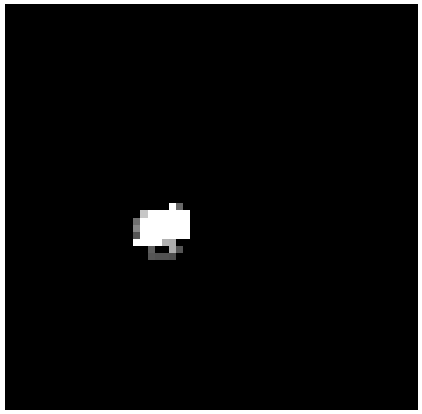
- Bildselektion: Konfidenz > 90%

Saliency Maps: Ergebnisse

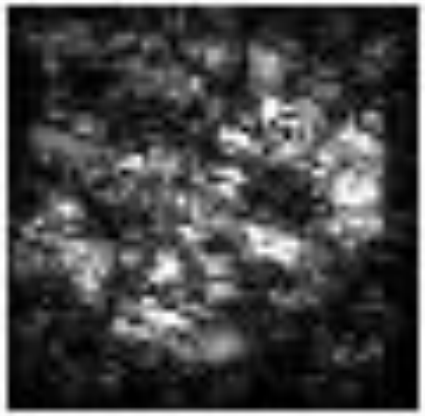
- Ungeglättete Verfahren (Erfolge):
  - Je Verfahren 0/1933 (0,00%)
- Geglättete Verfahren (Erfolge):
  - Guided Backpropagation: 7/1933 (0,36%)
  - Integrated Gradient: 3/1933 (0,16%)
  - Vanilla Saliency: 3/1933 (0,16%)



Zul. Höchstgeschw. 50
99,95%



Zul. Höchstgeschw. 30
92,83%

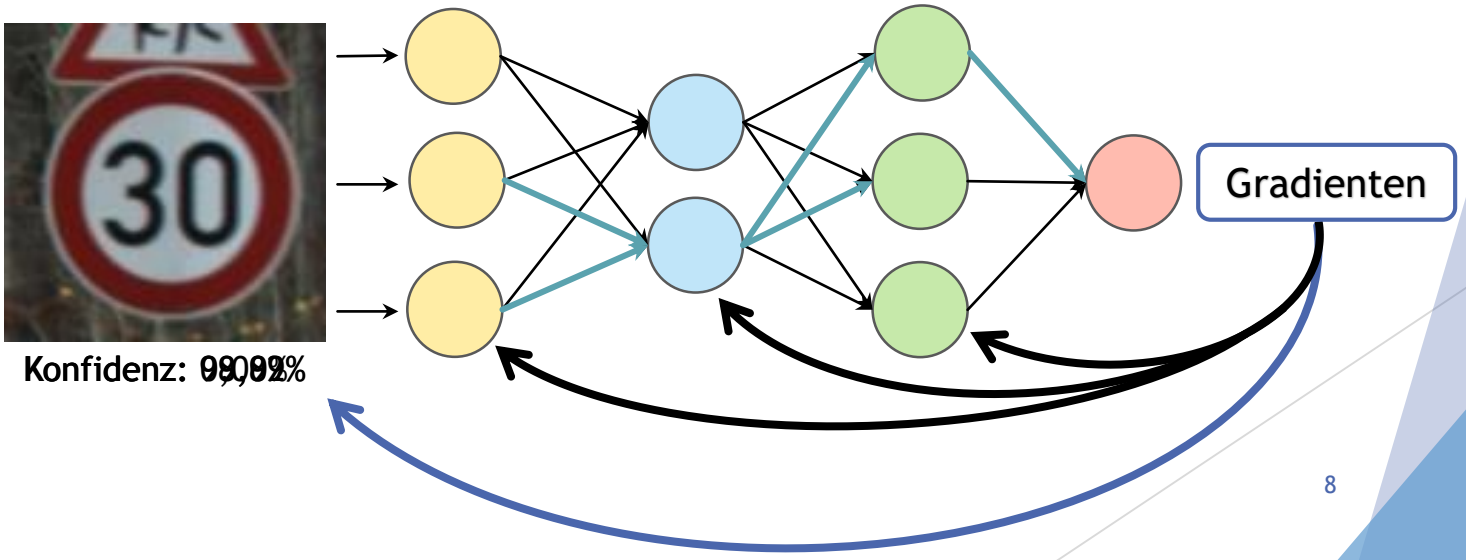


Baustelle
99,99%

Gradient Ascent

- Gradient Descent [7]
  - Berechnung von Gradienten durch Backpropagation
  - Gradient Descent optimiert **Parameter** des NN

- Gradient Ascent: Targeted Backpropagation [6]
  - „optimiert“ **das Bild** iterativ mithilfe von Backpropagation



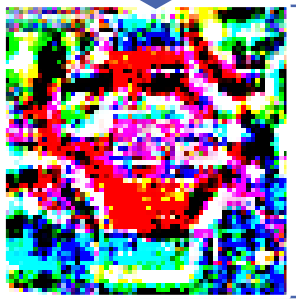
Gradient Ascent: Implementierung

Vorverarbeitung

- Implementierung und Training des Surrogat CNNs „AlexNet“
- Spezifikation der Zielklasse und Zufallsbilderzeugung

Gradient Ascent

- Berechnung der Gradienten
- Veränderung des Eingabebildes bis Zielklasse mit Konfidenz >90% angenähert wird



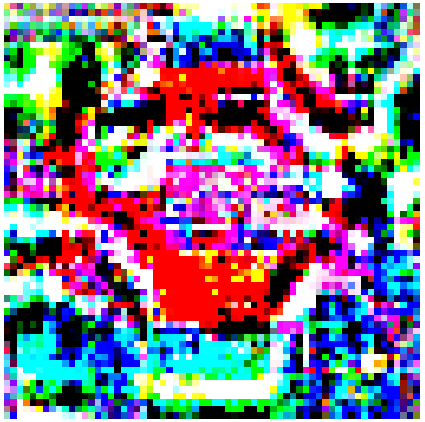
Remote Evaluierung

- Bildselektion: Konfidenz >90%

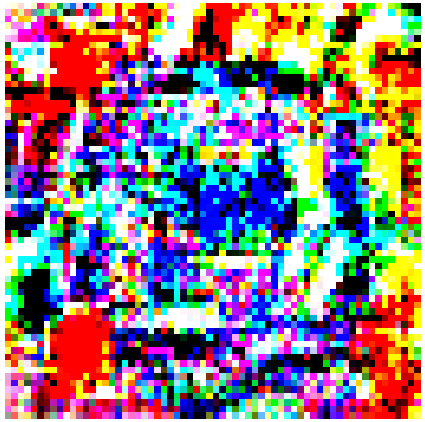
Gradient Ascent: Ergebnisse

- 43 Ergebnisbilder entsprechend der im GTSRB-Datensatz vorhandenen Klassen

- Erfolge: 20 Bilder (46,51%) mit Konfidenz >90%
  - Nur bei 4 Bildern Übereinstimmung der Ursprungs- und Zielklasse



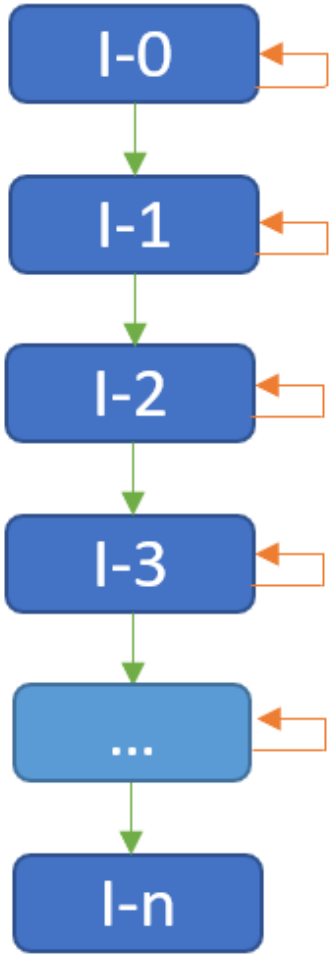
Vorfahrt gewähren
99,99%



Kreisverkehr
98,68%

Allgem. Überholverbot
99,99%

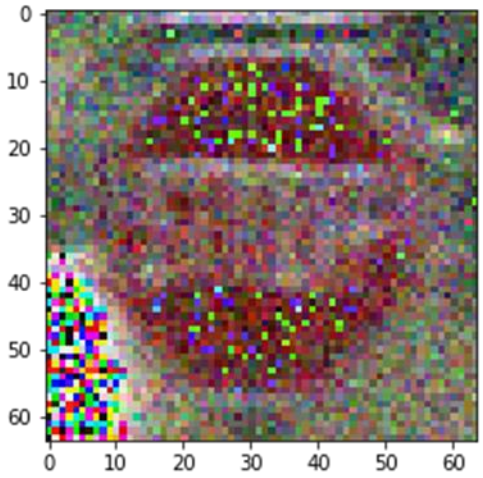
Degeneration



1. Eingabe eines Ausgangsbildes: **Echtes Verkehrsschild** (I-0)
2. Verrauschen des Schildes
3. Senden an Schnittstelle
   a) Weiterhin als Verkehrsschild erkannt?
   Weiter benutzen (Grüner Pfeil)
   b) Zu niedrige Konfidenz?
   Entferne Rauschen
   und wdh. ab Schritt 2 (Oranger Pfeil)
4. Wiederhole bis n-Wiederholungen erreicht

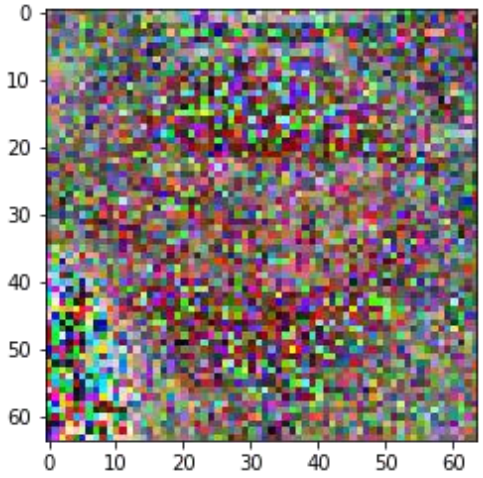
Degeneration: Ergebnisse

Tiefe 500



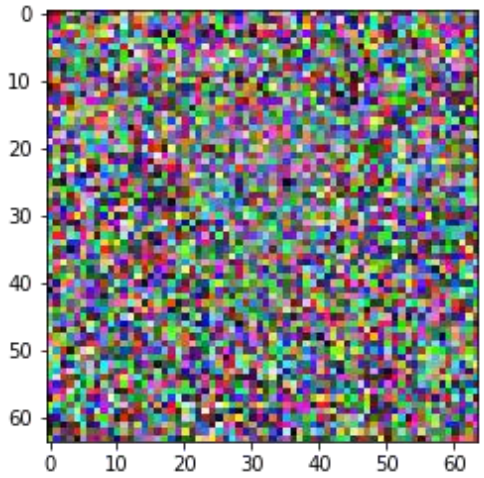
~97%

Tiefe 2500



~95%

Tiefe 5000



~92%

Degeneration: Vor- und Nachteile

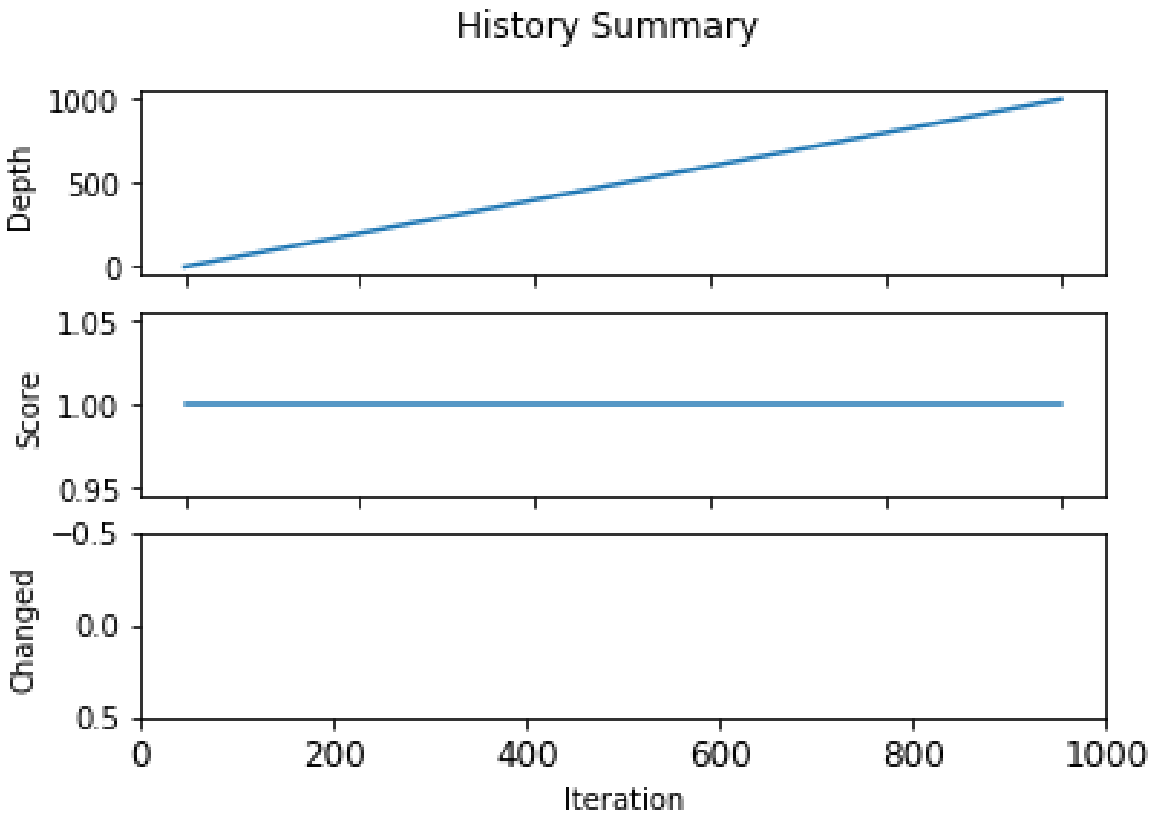
Vorteile:

- Einfache Implementierung (ca. 100 Zeilen Code)
- Modellunabhängig
- Ebenfalls für Random Forests, SVMs, etc.
- Ergebnisse beliebig gut je nach Zeitaufwand
- Zwischenergebnisse wieder aufgreifbar

Nachteile:

- Lange Laufzeit (Lokal ca. 3 Minuten pro Bild, Remote 1-2 Stunden)
- Längere Laufzeit bei besseren Netzen
- Lokal ggfs. längere Laufzeiten auf schlechter Hardware
- Feintuning der Rauschfunktion erforderlich

Degeneration: Verbesserung durch Batch-Processing



Zusammenfassung I

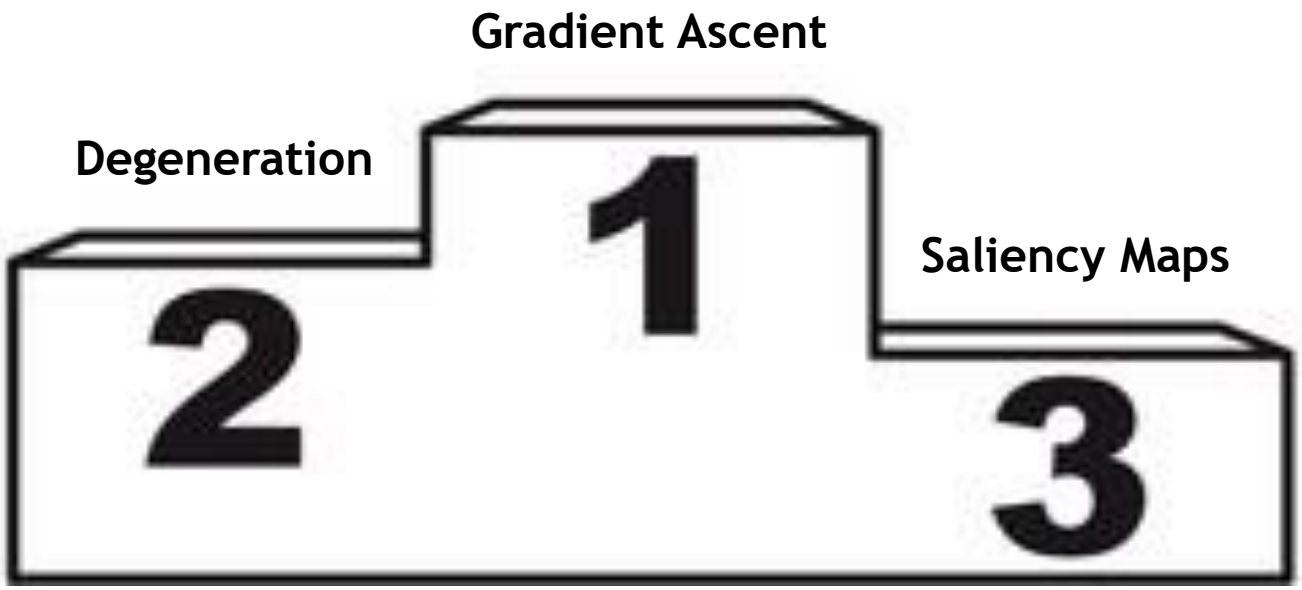
- Erfolge pro Zeiteinheit bei Gradient Ascent am höchsten (46,51%)
  - Geringe Laufzeit, zielgerichtete Bilderzeugung
- Degeneration bietet schnelle Erfolge, „guter erster Ansatz“
  - Sehr lange Laufzeit, zielgerichtet und mit „Erfolgsgarantie“ (Erfolg der Bilder ist zu jeder Zeit bekannt)
- Saliency Map kann Grundlage bieten für Adversarial Attacks, weitere Optimierung erforderlich
  - Lange Laufzeit, keine zielgerichtete Bilderzeugung möglich (Brute Force; Stichprobe aus Testdatensatz)
- Bemerkbare Einschränkung durch geringe Auflösung der Bilder (Mehr Pixel → Höhere Entropie der einzelnen Pixel)

Zusammenfassung II

	Remote Degen.	Local Degen.	Gradient Ascent	Smoothed Vanilla	Smoothed Integrated Gradient	Smoothed Guided Backprop.
Bilder (Anz.)	5	5	43	1933	1933	1933
Dauer (min)	309:10	18:30	0:13	36:02	36:05	41:26
Bilder/s	/	/	3,30	0,89	0,89	0,77
Erfolge (abs.)	5	0	20	3	3	7
Erfolge (rel.)	100%	0,00%	46,51%	0,16%	0,15%	0,36%
Laufzeit	---	+	++	--	--	--

Nicht gelistet: Saliency Map Standard Verfahren, Batch-Degeneration

Zusammenfassung III



Kontakt

- **Leonhard Applis**
  applisle74858@th-nuernberg.de
- **Peter Bauer**
  bauerpe72692@th-nuernberg.de
- **Andreas Porada**
  poradaan60975@th-nuernberg.de
- **Florian Stöckl**
  stoecklfl75458@th-nuernberg.de

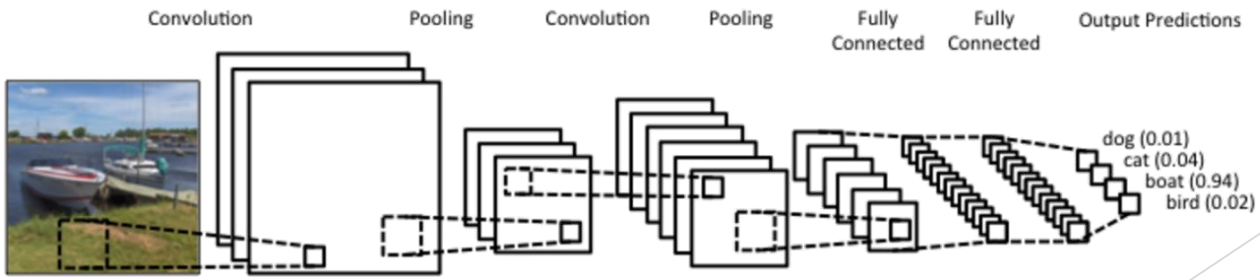
Anhang

Anhang – Aphrodite CNN

- 10 Layer
- Topologischer Aufbau:
  - Conv + Pooling
  - ReLU
  - Classifier
- Accuracy: 96,5%

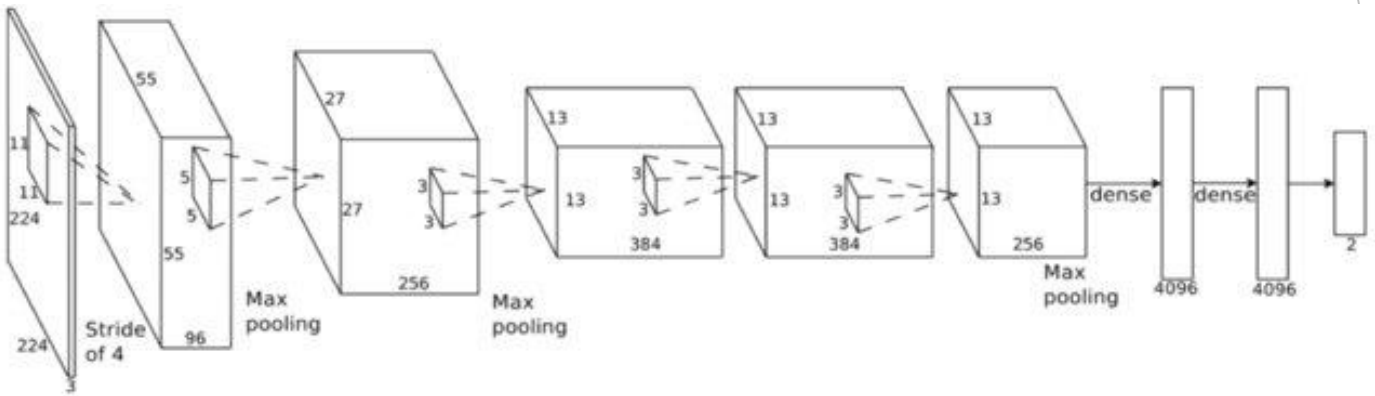
Layer (type)	Output Shape	Param #
conv2d_30 (Conv2D)	(None, 64, 64, 32)	896
conv2d_31 (Conv2D)	(None, 62, 62, 32)	9248
max_pooling2d_15 (MaxPooling	(None, 31, 31, 32)	0
dropout_15 (Dropout)	(None, 31, 31, 32)	0
conv2d_32 (Conv2D)	(None, 31, 31, 64)	18496
conv2d_33 (Conv2D)	(None, 29, 29, 128)	73856
max_pooling2d_16 (MaxPooling	(None, 14, 14, 128)	0
dropout_16 (Dropout)	(None, 14, 14, 128)	0
flatten_6 (Flatten)	(None, 25088)	0
dense_18 (Dense)	(None, 128)	3211392
dense_19 (Dense)	(None, 128)	16512
dense_20 (Dense)	(None, 43)	5547

Total params: 3,335,947
Trainable params: 3,335,947
Non-trainable params: 0



Anhang – AlexNet CNN

- Der Aufgabenstellung angepasstes „State-of-the-Art“ CNN [8]



- Modifikationen:
  - Eingabelayer 64 x 64 x 3 (Statt 224 x 224 x 3)
  - Ausgabelayer 43 (Statt 2)

- Accuracy: 89%

Anhang - Literatur

[1] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z. Berkay Celik, and Ananthram Swami. Practical Black-box Attacks Against Machine Learning.pdf. *arXiv:1602.02697 [cs]*, February 2016. arXiv: 1602.02697.

[2] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention forrapidsceneanalysis. *IEEETransactionsonPatternAnalysisandMachine Intelligence*, 20(11):1254–1259, November 1998.

[3] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. *arXiv:1312.6034 [cs]*, December 2013. arXiv: 1312.6034.

[4] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for Simplicity: The All Convolutional Net. *arXiv:1412.6806 [cs]*, December 2014. arXiv: 1412.6806.

[5] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic Attribution for Deep Networks. *arXiv:1703.01365 [cs]*, March 2017. arXiv: 1703.01365.

[6] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. SmoothGrad: removing noise by adding noise. *arXiv:1706.03825 [cs, stat]*, June 2017. arXiv: 1706.03825.

[7] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into Transferable Adversarial Examples and Black-box Attacks. arXiv:1611.02770 [cs], November 2016. arXiv: 1611.02770

[8] XueFei Zhou. Understanding the Convolutional Neural Networks with Gradient Descent and Backpropagation. Journal of Physics: Conference Series, 1004(1):012028, 2018.

[9] Datasets, Transforms and Models specific to Computer Vision: pytorch/vision. https://github.com/pytorch/vision, January 2019. original-date: 2016-11-09T23:11:43Z.