

# Anomalie-Segmentierung auf Bilddaten für Autonome Fahrzeuge via Deep Learning

Theo Langer<sup>1</sup>

**Abstract:** Die Fähigkeit unbekannten Objekte zu erkennen ist ein wichtiger Aspekt von Computer Vision Systemen in sicherheitskritischen Anwendungen wie dem autonomen Fahren. Die Verwendung beispielhafter unbekannter Objekte während des Trainings - genannt Outlier Exposure - hat für Klassifizierungsaufgaben in der Vergangenheit zuverlässig die Performanz erhöht. Dieser Ansatz wird hier mit speziellen Loss-Funktionen kombiniert und auf eine Segmentierungsaufgabe angewendet. Ebenso werden unterschiedliche Detektionsmethoden betrachtet. Die Ergebnisse zeigen, dass die Verwendung zusätzlicher Anomalien während des Trainings die Ergebnisse im allgemeinen verbessert, und zwar unabhängig von Loss-Funktion und OOD-Detektionsmethode. Die besten Ergebnisse liefert dabei das Energy-Loss in Kombination mit der Energy-basierten Detektionsmethode.

**Keywords:** Deep Learning; Anomaliedetektion; Segmentierung; Autonomes Fahren

## 1 Einleitung

Die Detektion unbekannter Objekte in Bildern ist ein wesentlicher Sicherheitsaspekt für Machine Learning-Algorithmen, wie sie beispielsweise im autonomen Fahren verwendet werden. Wird etwa ein Objekt auf der Straße fälschlicherweise als Hintergrund detektiert wird (vgl. Abb. 1) kann es schnell zu Unfällen kommen. Das Erkennen unbekannter Objekte wird *Out-of-distribution* (OOD) Detektion genannt.

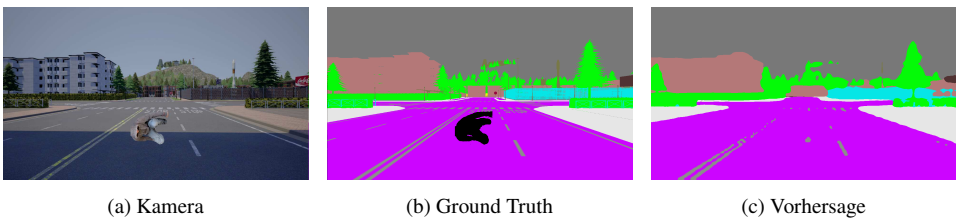


Abb. 1: Bild eines synthetisch eingefügten, seitlich liegenden Menschen auf der Straße. Links das ursprüngliche Bild, in der Mitte die Zielsegmentierung und rechts die Vorhersage. Der Mensch wird nicht erkannt, sondern als Straßenuntergrund bzw. Fahrbahnmarkierung klassifiziert.

---

<sup>1</sup> Otto-von-Guericke-Universität, Fakultät für Informatik, Universitätsplatz 2, 39106 Magdeburg, Deutschland [bluethео.langer@ovgu.de](mailto:bluethео.langer@ovgu.de)

In dieser Arbeit sollen die Auswirkungen von speziellen OOD-Losses und OOD-Detektoren (die bis jetzt meist nur für die OOD Detektion bei der Klassifikation von Bildern genutzt wurden), sowie das Einfügen von zusätzlichen Anomalien während des Trainings bei der OOD Detektion bei der Segmentierung von Bildern untersucht werden.

Der Rest dieser Arbeit ist wie folgt gegliedert: in Bestehende Arbeiten (Abschnitt 2), wo die unterschiedlichen Loss Funktionen und Detektoren vorgestellt werden, Experimente (Abschnitt 3), wo die Experimente vorgestellt und ausgewertet werden sowie Fazit & Ausblick (Abschnitt 4).

## 2 Bestehende Arbeiten

Die meisten Arbeiten bei der OOD-Detektion von Bildern beschäftigen sich mit der Klassifikation und nicht der Segmentierung von Bildern. Für die Klassifikation von Bildern gibt es mehrere Ansätze mit unterschiedlichen OOD-Losses und OOD-Detektoren. Diese gut funktionierenden Ansätze (Energy-Loss [Li20], Outlier Exposure-Loss [HMD18], Entropy-Loss [Li20]) werden in dieser Arbeit auf die Segmentierung von Bildern angewendet. Das Hinzufügen von zusätzlichen Anomalien während des Trainings führte bei einer anderen Arbeit [CRG20] zu einer Verbesserung der Ergebnisse. Diese beiden Ansätze der speziellen OOD-Losses (bislang nur für die Klassifikation von Bildern angewendet) und das Hinzufügen von künstlichen Anomalien werden in dieser Arbeit kombiniert um Bilder zu segmentieren.

### 2.1 OOD-Detektion

Bei der OOD-Detektion wird versucht Pixel zu finden die dem Modell unbekannt sind, also nicht in dem Trainingsdatensatz vorkamen. Pixel von bekannten (in-distribution) Objekten werden im folgenden als IN-Pixel bezeichnet und haben eine positive Zielklasse  $y \geq 0$ , die restlichen (out of distribution) Pixel als OOD-Pixel ( $y < 0$ ).  $f(x)$  steht für die Logitwerte der Pixel nach der Verarbeitung durch das Modell.  $\sigma$  bezeichnet die Softmax-Funktion.

In der Literatur gibt es unterschiedliche Ansätze OOD-Pixel zu detektieren. Die in dieser Arbeit verwendeten OOD-Detektoren sind im folgenden beschrieben:

**MSP** Die “Maximum Softmax Probability”(MSP) nimmt den maximalen Softmax-Wert als OOD-Score [He19]:

$$\text{MSP}(x) = -\max_k \frac{e^{f(x)_k}}{\sum_i e^{f(x)_i}} = -\max_k \sigma(f(x))_k . \quad (1)$$

**MaxLogit** Die MaxLogit [He19] Methode ist definiert als

$$\text{MaxLogit}(x) = -\max_k f(x)_k \quad (2)$$

und eignet sich besser für Anomalieerkennung bei vielen Klassen, da es unabhängig von Anzahl der Klassen ist und unnormalisiert [He19].

**Energy** In [Li20] wird zur OOD-Detektion die Energie wert der Eingabe verwendet, die in Anlehnung an die freie Energie (Helmholtz-Potenzial) [CG87] folgendermaßen definiert wird:

$$\text{Energy}(x) = E(x) = -\log \sum_i e^{f_i(x)} . \quad (3)$$

**Softmax Entropy** Die Entropy eines Pixels berechnet sich aus den Softmax-Wahrscheinlichkeiten  $\sigma(f(x)) \in (0, 1)^q$  der einzelnen Logits des Pixels, wobei  $q = |C| \in \mathbb{N}$  die Anzahl der Klassen in der Menge  $C = \{1, 2, \dots, q\}$  ist [CRG20].

$$\text{Entropy}(x) = H(x) = -\sum_{j \in C} \sigma(f(x))_j \log \sigma(f(x))_j \quad (4)$$

## 2.2 Loss-Funktionen für OOD Detection

Durch die Verwendung von OOD-Daten während des Trainings werden spezielle Lossfunktionen nötig, die sowohl die IN-, als auch OOD-Pixel verarbeiten können (OOD-Klassen besitzen keinen klassischen Ausgabewert in den Logits, wie IN-Klassen).

**Cross-Entropy Loss** Der Cross-Entropy Loss [Co58] für einen Pixel ist wie folgt definiert (wobei die OOD-Pixel ignoriert werden):

$$\mathcal{L}_{\text{NLL}}(x, y) = -\log \sigma(f(x))_y . \quad (5)$$

**Energy Loss** Der Energy Loss verwendet beim Training sowohl IN (Klassenlabel  $y \geq 0$ ), als auch OOD (Klassenlabel  $y < 0$ ) Daten. Er basiert, auf dem Crossentropy-Loss, wird jedoch um einen Energie-basierten regularisierenden Loss  $\mathcal{L}_{\text{ER}}(x, y)$  erweitert [Li20]:

$$\mathcal{L}_{\text{E}}(x, y) = \mathcal{L}_{\text{NLL}}(x, y) + \mathcal{L}_{\text{ER}}(x, y) . \quad (6)$$

Der Energie basierte Loss  $\mathcal{L}_{\text{ER}}$  setzt sich aus mehreren zu wählenden Hyperparametern zusammen:  $m_{\text{in}}$ ,  $m_{\text{out}}$  und  $\alpha$  (Wichtungsfaktor)

$$\mathcal{L}_{\text{ER}}(x, y) = \alpha \begin{cases} \max\{0, (E(x) - m_{\text{in}})\}^2 & \text{wenn } y \geq 0, \\ \max\{0, (m_{\text{out}} - E(x))\}^2 & \text{sonst.} \end{cases} \quad (7)$$

**Outlier Exposure Loss** Der Outlier Exposure Loss [HMD18] besteht aus dem Crossentropy Loss für IN-Pixel und einem extra Term für die OOD-Pixel und ist wie folgt definiert:

$$\mathcal{L}_{OE}(x, y) = \begin{cases} -\log \sigma_y(f(x)) & \text{wenn } y \geq 0, \\ \alpha \left( \sum_{j \in C} f(x)_j - \log \left( \sum_{j \in C} e^{f(x)_j} \right) \right) & \text{sonst.} \end{cases} \quad (8)$$

IN-Pixel werden vom Outlier Exposure Loss also durch die Crossentropy bewertet und OOD-Pixel nur durch einen separaten Ausdruck.

**Entropy Loss** Der Entropy-Loss [CRG20] arbeitet beim Training mit IN und OOD Daten. Er setzt sich aus dem  $\mathcal{L}_{in}$ -Loss für IN-Pixel ( $y \geq 0$ ) und dem  $\mathcal{L}_{out}$ -Loss für OOD-Pixel ( $y < 0$ ) zusammen. Das Verhältnis dieser Losse wird durch  $\lambda$  gewichtet.

$$\mathcal{L}_H(x, y) = (1 - \lambda) \mathcal{L}_{in}(x, y) + \lambda \mathcal{L}_{out}(x) \quad (9)$$

Für IN-Pixel berücksichtigt der Cross-Entropy-Loss den Wert der Zielklasse  $y$ . Bei OOD-Pixeln werden die Werte aller Klassen berücksichtigt.  $q$  ist die Anzahl der Klassen  $C$  und  $\mathbb{1}_{j=y}$  ist definiert als 1, wenn  $j = y$ , und 0 sonst.

$$\mathcal{L}_{in}(x, y) = - \sum_{j \in C} \mathbb{1}_{j=y} \log(\sigma_y(f(x))) \quad (10)$$

$$\mathcal{L}_{out}(x, y) = - \sum_{j \in C} \frac{1}{q} \log(\sigma_y(f(x))) \quad (11)$$

Hierbei ist Minimierung der Entropy-Loss Funktion  $\mathcal{L}_H$ , äquivalent zur Maximierung der Softmax Entropy Gleichung (4) [CRG20].

## 2.3 Datensatz

**StreetHazard** Der StreetHazard Datensatz [He19] ist ein künstlicher Datensatz der durch die Unreal Engine mit der CARLA Simulationsumgebung [Do17] erzeugt wurde. Dabei werden unter anderem uneinheitliche chromatische Verhältnisse, Randeffekte sowie Veränderung der Beleuchtung verhindert, sodass sich anomale Objekte nicht durch triviale Hinweise erkennen lassen. Der Datensatz ist ein Anomalie-Segmentierungsdatensatz, der mithilfe einer Simulation realistisch wirkende anomale Objekte in in realistischen Szenarien einfügt.

Im Trainings-(5125 Bilder) und Evaluationsdatensatz (1031 Bilder) sind keine Anomalien enthalten (vgl. Tab. 1). Im Testdatensatz (1500 Bilder) ist auf jedem Bild mindestens ein

OOD-Objekt. Die Bilder für jeden Teildatensatz wurden in einer anderen virtuellen Stadt der CARLA Simulation erzeugt.

Tab. 1: Übersicht über die Aufteilung der Daten im StreetHazard Datensatz

StreetHazard	Training	Evaluation	Test
Bilder	5125	1031	1500
CarlaTown	1,2,3	4	5,6
Anomalie	Nein	Nein	min. 1 pro Bild

**Erweiterter StreetHazard** Um den Energy-, Entropy- so wie den Outlier-Exposure Loss effektiv zu verwenden, müssen auch im Trainings und Evaluationsdatensatz OOD-Objekte enthalten sein. So wurde in 5% aller Trainings und Evaluationsbilder ein OOD Objekt (Hindernis) durch Pixel-Ersetzung eingefügt, ohne die Szene nachträglich zu bearbeiten (vgl. Abb. 2). Diese neu eingefügten Objekte (Hindernisse) unterscheiden sich dabei von den OOD-Objekten im Testdatensatz.

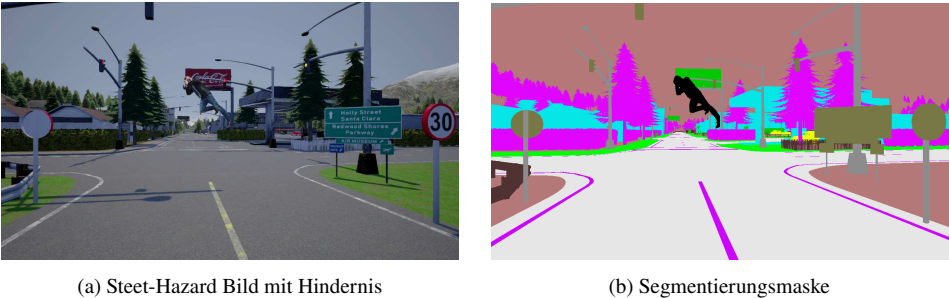


Abb. 2: StreetHazard Datensatz mit einem Hindernis und unten die zugehörige Segmentierungsmaske. Das Hindernis ist hier mit schwarz dargestellt.

Tab. 2: Übersicht über die Aufteilung der Daten im StreetHazard Datensatz

Erweiterter StreetHazard	Training	Evaluation	Test
Bilder	5125	1031	1500
CarlaTown	1, 2, 3	4	5, 6
Anomalie	5% aller Bilder mit Hindernis	5% aller Bilder mit Hindernis	min. 1 pro Bild

In dem Trainingsdatensatz des Erweiterten Streethazard Datensatzes (vgl. Tab. 2) wurde auf 270 Bildern genau ein Hindernis eingefügt, bei dem Validationsdatensatz auf 58 Bildern.<sup>2</sup> Dabei wurden 81 unterschiedliche Hindernisse verwendet, die sich in 20 Klassen (z.B. Fahrräder, Dreiräder, Kinder, Menschen, Tiere) aufteilen.

Vor dem Einfügen wurden die Hindernisse zufällig rotiert (von 0° bis 359°) und skaliert

<sup>2</sup> Die Bilder der Hindernisse entstammen [cdn.pixabay.com](https://cdn.pixabay.com).

(0, 2 bis 0, 5 der ursprünglichen Größe). Die Position der Hindernisse in den Bildern wurde zufällig gewählt. Die Annotation für die eingefügten Pixel wurde mit  $-1$  als OOD gewählt.

3 Experimente

Um die unterschiedlichen Losses und OOD-Detektionsmethoden zu vergleichen werden vier Experimente (vgl. Tab. 3) durchgeführt. Jedes dieses Experimente wird mit den vier in Abschnitt 2.1 beschriebenen OOD-Detektionsverfahren evaluiert, sodass insgesamt 16 unterschiedliche Versuchsreihen vorliegen. Die Implementierung erfolgte mithilfe der PyTorch- [Pa19] und der PyTorch-OOD [KFO22]-Bibliothek.

**Trainingssetup** Es wurde dasselbe Trainingssetup wie im Paper von Hendricks et al. [He19] verwendet: Das Modell enthält Teile aus dem vortrainierten ImageNet [De09] und ein Pyramid-Pooling-Module [Zh16].

Loss	Parameter	Datensatz
Cross-Entropy Loss		StreetHazard
Energy Loss	$\alpha = 0.1, m_{in} = m_{out} = -10$	Erweiterter Streethazard
Outlier Loss	$\alpha = 0.5$	Erweiterter Streethazard
Entropy Loss	$\lambda = 0.25$	Erweiterter Streethazard

Tab. 3: Übersicht über die Experimente

**Parameterwahl Energy-Loss** Der Energy-Loss benötigt die Hyperparameter  $\alpha$ ,  $m_{in}$  und  $m_{out}$ . Um bestmögliche Ergebnisse zu erzielen sollten  $m_{in}$  und  $m_{out}$  spezifisch gewählt werden.  $m_{in}$  wird so gewählt, dass möglichst viele IN-Pixel eine geringe Energie haben als  $m_{in}$  und  $m_{out}$  so, dass möglichst viele OOD-Pixel eine höhere Energie haben [Li20].

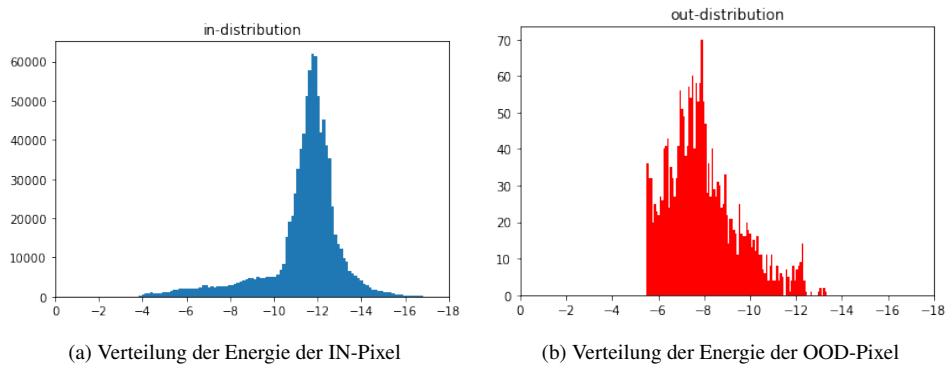


Abb. 3: Die Verteilungen der Energien für IN und OOD-Pixel auf dem Validationsdatensatz

Anhand von Abb. 3 wurde  $m_{in} = m_{out} = -10$  gewählt.

### 3.1 Metriken

Um die Qualität der Ergebnisse zu bemessen, werden die AUROC-, AUPR- und FPR-Metrik verwendet. Diese Metriken werden auch von anderen Arbeiten verwendet [Li20] [HMD18].

**AUROC** Die **area under the receiver operating characteristic** gibt die Fläche unter der ROC Krue an. Diese Metrik kann als Wahrscheinlichkeit interpretiert werden, dass ein OOD-Pixel eine höheren OOD-Score erhält als ein IN-Pixel. Je höher dieser Wert ist, desto besser ist das Modell.

**AUPR** Die **area under the precision-recall curve** gibt die Fläche unter der Precision-Recall-Kurve an. Diese Metrik eignet sich besser für unbalancierte Klassenverteilungen mit wenigen Anomalien [MS99].

**FPR** Die FPR Metrik gibt hier die falsch positiv Rate bei 95% richtig positiv Rate an. Diese Metrik korreliert mit der AUROC Metrik.

3.2 Ergebnisse

Tab. 4: Ergebnisse der Experimente. Die **fett** hervorgehobenen Werte, sind die besten Werte eines Experiments je Metrik. Sämtliche Werte sind in Prozent angegeben. ↑ zeigt an dass höhere Werte besser sind, ↓ das Gegenteil.

Loss	OOD	IoU ↑	Acc ↑	AUROC ↑	AUPR-IN ↑	FPR95 ↓
Cross-Entropy	MSP	51.49	90.12	89.19	6.97	26.34
	MaxLogit			89.26	10.27	33.84
	Energy			89.32	<b>10.50</b>	33.81
	Entropy			<b>89.65</b>	8.58	<b>26.30</b>
Energy	MSP	51.76	89.96	89.98	8.22	24.23
	MaxLogit			92.36	14.54	24.37
	Energy			<b>92.50</b>	<b>15.49</b>	24.18
	Entropy			90.67	10.86	<b>23.99</b>
Outlier Exposure	MSP	51.69	90.17	90.01	9.37	25.32
	MaxLogit			91.64	15.65	27.58
	Energy			<b>91.72</b>	<b>16.42</b>	27.47
	Entropy			90.64	12.41	<b>25.13</b>
Entropy	MSP	51.71	90.01	90.06	9.90	24.52
	MaxLogit			91.49	16.64	28.09
	Energy			<b>91.62</b>	<b>17.49</b>	27.92
	Entropy			90.68	13.05	<b>24.39</b>

**Evaluationsmetriken** Die Ergebnisse der OOD-Detektoren sind mit spezifischen Losses besser, als mit dem Cross-Entropy-Loss. Energy- und Entropie-basierte Loss-Funktionen erzielten dabei die besten Ergebnisse (vgl. Tab. 4).

Um die Ergebnisse der Experimente zu vergleichen wurde der Wilcoxon-Signed-Rank Test [RN11] verwendet. Dabei werden für jedes einzelne Bild die Werte der jeweiligen Metrik bestimmt und diese Werte anschließend miteinander verglichen. Als Nullhypothese wird  $i$  ist signifikant größer als  $j$  gewählt, wenn der Mittelwert aller Bilder von  $i$  größer als der Mittelwert aller Bilder von  $j$  ist, sonst wird  $i$  ist signifikant kleiner als  $j$  als Nullhypothese gewählt. Der Signifikanztest brachte folgende Ergebnisse bezüglich der Metriken:

**AUROC** Für die AUROC-Metrik ist die Versuchsreihe mit Energie-Loss und Energie Detektor signifikant besser als alle anderen Versuchsreihen. Die Experimente mit spezifischen Losses sind signifikant besser als das Cross-Entropy-Experiment.



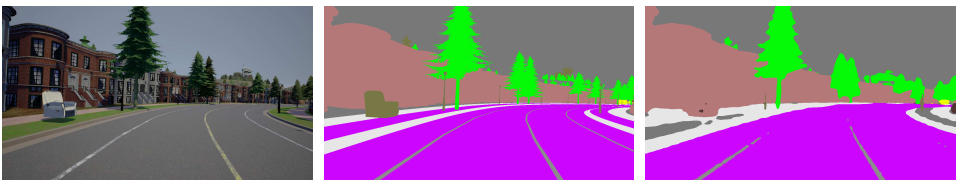
**AUPR** Für die AUPR-Metrik sind die Versuchsreihen mit dem Energie und Entropie-Loss und der Energie-Detektion signifikant besser als alle anderen Versuchsreihen und voneinander nicht signifikant verschieden

**FPR** Für die FPR-Metrik ist die Versuchsreihe mit Energie-Loss und Energie-Detektion signifikant besser als alle anderen Versuchsreihen (bzw. nicht signifikant unterschiedlich zu Energie Loss mit Entropy Detektor). Die Ergebnisse der OOD Detektoren mit den spezifischen Losses sind signifikant besser (oder nicht signifikant unterschiedlich), als das Cross-Entropy-Experiment.

**Beobachtungen** Durch die Signifikanztests wird deutlich, dass die Versuchsreihe mit dem *Energy Loss* und dem *Energie-Detektor* signifikant besser (bei AUPR gibt es keinen signifikanten Unterschied zum Entropy Loss und Energie-Detektor) ist als alle anderen Versuchsreihen. Vergleicht man die OOD-Detektion innerhalb eines Experiments, so ist die Energie-Detektion meist auch signifikant besser als die anderen OOD-Detektionen.

### 3.3 Qualitative Ergebnisse

In Abb. 4 ist Bild 101 aus dem Testdatensatz mit korrekter Segmentierung und der vorhergesagten Segmentierung dargestellt. Diese vorhergesagte Segmentierung wurde von dem Modell getroffen, welches mit dem *Energy Loss* trainiert wurde. Die Anomalien hier sind der Sessel links im Bild, das Schlagzeug über den Bäumen, sowie Spielzeugkreisel über dem Haus.



(a) Eingabebild

(b) korrekte Segmentierung

(c) Vorhergesagte Segmentierung

Abb. 4: Bild 101: Eingabe, korrekte Segmentierung, sowie vorhergesagte Segmentierung des mit dem Energy Loss trainierten Modells.

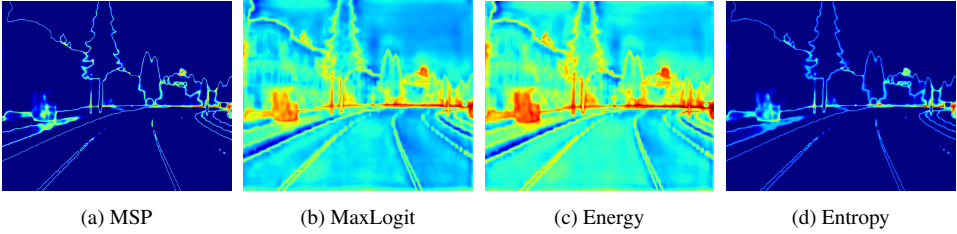


Abb. 5: Bild 101: Ergebnisse der unterschiedlichen OOD-Detektoren auf Basis des mit dem Energy Loss trainierten Modells. Je röter, höher ist der OOD-Score, desto unsicherer ist die Vorhersage.

In Abb. 5 wird anhand der deutlichen roten Färbung erkennbar, dass der Energydetektor die drei Anomalien am besten detektieren kann.

In Abb. 6c) erkennt der Energy Detektor (vgl. Abb. 7) ebenfalls am besten die Anomalie (ein Schlagzeug über den Bäumen).



Abb. 6: Bild 175: Eingabe, korrekte Segmentierung, sowie vorhergesagte Segmentierung des mit dem Energy Loss trainierten Modells.

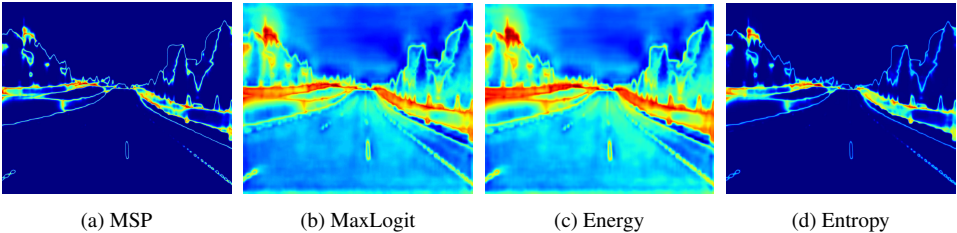


Abb. 7: Bild 175: Ergebnisse der unterschiedlichen OOD-Detektoren auf Basis des mit dem Energy Loss trainierten Modells. Je röter, höher ist der OOD-Score, desto unsicherer ist die Vorhersage.

## 4 Fazit & Ausblick

**Fazit** Das Hinzufügen von zusätzlichen Anomalien während des Trainings, sowie die Verwendung von speziellen OOD-Losses führte zu besseren Ergebnissen bei der OOD-Detektion, als bei dem Cross-Entropy-Experiment ohne zusätzlichen Hindernissen.

Der in diesem Setup beste Loss war der Energie Loss und der beste OOD-Detektor der

Energie-Detektor. Die ursprünglich für die Klassifikation entwickelten Lossfunktionen liefern alle bessere Ergebnisse als der Cross-Entropy-Loss.

**Ausblick** Das hier angewendete Verfahren mit einer Erweiterung des Datensatzes und den speziellen Loss-Funktionen kann auf weiteren Datensätzen z.B. dem BDD-Anomaly [Yu18] (einem Segmentationsdatensatz mit realen Bildern) verwendet werden, um die Ergebnisse zu validieren.

Daten Augmentierung verbessert meist die Performance von neuronalen Netzen [SK19]. So könnte eine Augmentation der Original bzw Hindernisbilder in Form von z.B. “noise injection“ [Mo18] oder “color space transformation“ [Ch14] die Performance des Netzes verbessern, was zu besseren Ergebnissen bei der OOD-Detektion führen könnte.

## Literaturverzeichnis

- [CG87] Cohen, E. Richard; Giacomo, Pierre: Symbols, units, nomenclature and fundamental constants in physics. *Physica* 25. North-Holland, Netherland, 1987. 1987 Revision. – Reprinted from *PHYSICA* 146A(1987), S. 1-68.
- [Ch14] Chatfield, Ken; Simonyan, Karen; Vedaldi, Andrea; Zisserman, Andrew: Return of the Devil in the Details: Delving Deep into Convolutional Nets. Bericht, November 2014. arXiv:1405.3531 [cs] type: article.
- [Co58] Cox, D. R.: The Regression Analysis of Binary Sequences. *Journal of the Royal Statistical Society. Series B (Methodological)*, 20(2):215–242, 1958.
- [CRG20] Chan, Robin; Rottmann, Matthias; Gottschalk, Hanno: Entropy Maximization and Meta Classification for Out-Of-Distribution Detection in Semantic Segmentation. Dezember 2020.
- [De09] Deng, Jia; Dong, Wei; Socher, Richard; Li, Li-Jia; Li, Kai; Fei-Fei, Li: ImageNet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, jun 2009.
- [Do17] Dosovitskiy, Alexey; Ros, German; Codevilla, Felipe; Lopez, Antonio; Koltun, Vladlen: CARLA: An Open Urban Driving Simulator. November 2017.
- [He19] Hendrycks, Dan; Basart, Steven; Mazeika, Mantas; Zou, Andy; Kwon, Joe; Mostajabi, Mohammadreza; Steinhardt, Jacob; Song, Dawn: Scaling Out-of-Distribution Detection for Real-World Settings. November 2019.
- [HMD18] Hendrycks, Dan; Mazeika, Mantas; Dietterich, Thomas: Deep Anomaly Detection with Outlier Exposure. Dezember 2018.
- [KFO22] Kirchheim, Konstantin; Filax, Marco; Ortmeier, Frank: PyTorch-OOD: A Library for Out-of-Distribution Detection based on PyTorch. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). S. 4350–4359, Juni 2022. ISSN: 2160-7516.

- [Li20] Liu, Weitang; Wang, Xiaoyun; Owens, John D.; Li, Yixuan: Energy-based Out-of-distribution Detection. Oktober 2020.
- [Mo18] Moreno-Barea, Francisco J.; Strazzera, Fiammetta; Jerez, Jose M.; Urda, Daniel; Franco, Leonardo: Forward Noise Adjustment Scheme for Data Augmentation. 2018 IEEE Symposium Series on Computational Intelligence (SSCI), S. 728–734, November 2018.
- [MS99] Manning, Christopher; Schütze, Hinrich: Foundations of Statistical Natural Language Processing. MIT Press, Mai 1999.
- [Pa19] Paszke, Adam; Gross, Sam; Massa, Francisco; Lerer, Adam; Bradbury, James; Chanan, Gregory; Killeen, Trevor; Lin, Zeming; Gimeshein, Natalia; Antiga, Luca; Desmaison, Alban; Köpf, Andreas; Yang, Edward; DeVito, Zach; Raison, Martin; Tejani, Alykhan; Chilamkurthy, Sasank; Steiner, Benoit; Fang, Lu; Bai, Junjie; Chintala, Soumith: PyTorch: An Imperative Style, High-Performance Deep Learning Library. Bericht, Dezember 2019. arXiv:1912.01703 [cs, stat] type: article.
- [RN11] Rey, Denise; Neuhäuser, Markus: Wilcoxon-Signed-Rank Test. In: International Encyclopedia of Statistical Science, S. 1658–1659. Springer Berlin Heidelberg, 2011.
- [SK19] Shorten, Connor; Khoshgoftaar, Taghi M: A survey on Image Data Augmentation for Deep Learning. Journal of Big Data, 6(1):60, Juli 2019.
- [Yu18] Yu, Fisher; Chen, Haofeng; Wang, Xin; Xian, Wenqi; Chen, Yingying; Liu, Fangchen; Madhavan, Vashisht; Darrell, Trevor: BDD100K: A Diverse Driving Dataset for Heterogeneous Multitask Learning. Mai 2018.
- [Zh16] Zhao, Hengshuang; Shi, Jianping; Qi, Xiaojuan; Wang, Xiaogang; Jia, Jiaya: Pyramid Scene Parsing Network. Dezember 2016.