Vorabfragen

- Sie möchten ein CNN zur Erkennung von Verkehrszeichen trainieren. Sie besitzen 300 Trainingsbilder pro Klasse.
  - Ist das ausreichend?
  - Falls nein: Was könnten Sie tun?

- Ein autonomes Fahrzeug erkennt auf einem Bild einen Fußgänger. Welche zusätzliche Information benötigt das Fahrzeug, um sicher bremsen oder ausweichen zu können?

Prof. Dr. Daniel Gaida
Professor für Cyber-Physische Systeme
Fakultät für Informatik und Ingenieurwissenschaften - Institut für Informatik
Technology Arts Sciences TH Köln

Deep Learning, Machine Learning und Künstliche Intelligenz – SS 26

Gelesen von Prof. Dr. Daniel Gaida

Prof. Dr. Daniel Gaida
Professor für Cyber-Physische Systeme
Fakultät für Informatik und Ingenieurwissenschaften - Institut für Informatik

Lernraum III

Inhalt

- Data Augmentation – synthetische Vergrößerung des Trainingsdatensatzes
  - Bilder
  - Text
  - Zeitreihen (future)

- Bildverarbeitung mit KI Algorithmen, die über Bildklassifikation hinausgeht
  - Objektdetektion
  - Objekttracking
  - Open-Vocabulary Objektdetektion
  - Bildsegmentierung
  - Open-Vocabulary Image Segmentation

Prof. Dr. Daniel Gaida
Professor für Cyber-Physische Systeme
Fakultät für Informatik und Ingenieurwissenschaften - Institut für Informatik
Technology Arts Sciences TH Köln

Lernziel

Die Studierenden können Methoden der Data Augmentation für Bilder und Text umsetzen, indem sie

- Code zur Data Augmentation in Python implementieren und verstehen können,

um später besser generalisierende DL/ML-Modelle trainieren zu können.

Die Studierenden können einfache Deep Learning Projekte zur Bilderkennung umsetzen, indem sie

- zwischen Objektdetektion, -tracking und –segmentierung eine geeignete Technik auswählen und
- diese mit einem Deep Learning Modell in Python umsetzen können,

um später verschiedene Bilderkennungsprobleme lösen zu können.

Prof. Dr. Daniel Gaida
Professor für Cyber-Physische Systeme
Fakultät für Informatik und Ingenieurwissenschaften - Institut für Informatik
Technology Arts Sciences TH Köln

Data Augmentation

Ziel

- Erzeugung weiterer Trainingsdaten, um ML/DL-Modelle zu erhalten, die besser generalisieren
- Durch leichte Veränderung der einzelnen Trainingsdatenpunkte werden neue, künstlich generierte, Datenpunkte erzeugt, die dem Trainingsdatensatz hinzugefügt werden.
- Dabei sind die Veränderungen so klein, dass das Label nicht verändert werden muss

- Beispiele:
  - Bilder:
    - Rotation, Translation, Farbänderung, Rauschen, …
  - Text:
    - Satzbau umstellen, Wörter durch Synonyme ersetzen, Rechtschreibfehler einbauen, …
  - Zeitreihen:
    - Rauschen, Frequenzveränderung, Amplitudenveränderung, …

Prof. Dr. Daniel Gaida
Professor für Cyber-Physische Systeme
Fakultät für Informatik und Ingenieurwissenschaften - Institut für Informatik
Technology Arts Sciences TH Köln

Data Augmentation

Weitere Ziele

- Robustheit erhöhen
  - Rechtschreibfehler, Bild- oder Tonrauschen, …
- Klassen balancieren
  - Generiere Daten für Klassen für die man zu wenig Daten hat
- Privatsphäre/Anonymisierung
  - Generiere Daten und trainiere ausschließlich mit den synthetischen Daten bzw. stelle diesen generierten Datensatz öffentlich zur Verfügung

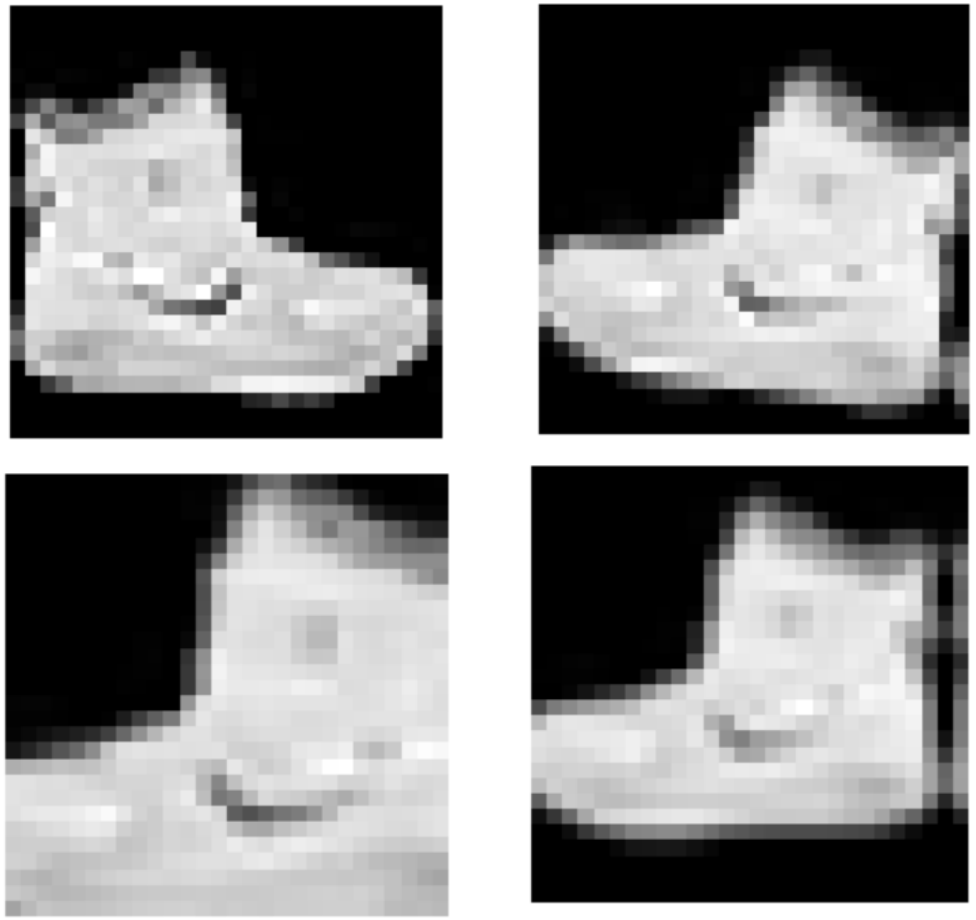
Prof. Dr. Daniel Gaida
Professor für Cyber-Physische Systeme
Fakultät für Informatik und Ingenieurwissenschaften - Institut für Informatik
Technology Arts Sciences TH Köln

Data Augmentation

Bilder

Transformationen

- Spiegeln
- Rotation
- Zoom
- Translation



Original:

Rotation und Translation:

Prof. Dr. Daniel Gaida
Professor für Cyber-Physische Systeme
Fakultät für Informatik und Ingenieurwissenschaften - Institut für Informatik
Technology Arts Sciences TH Köln

Data Augmentation

Bilder

Keras Layer

- `layers.RandomFlip("horizontal")`
- `layers.RandomRotation(0.1)`
- `layers.RandomZoom(0.3)`
- `layers.RandomTranslation(height_factor=0.1, width_factor=0.1)`

```
keras.Sequential([
    layers.RandomRotation(0.1),
    layers.RandomTranslation(height_factor=0.1, width_factor=0.1)
])
```

s. Notebook 01_fminst_data_augmentation.ipynb

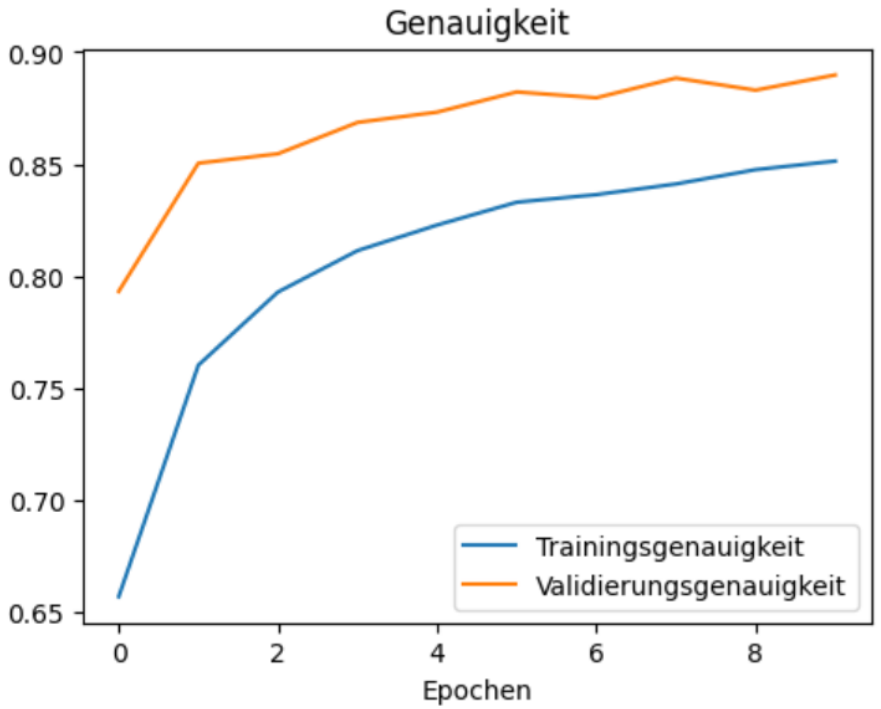
Prof. Dr. Daniel Gaida
Professor für Cyber-Physische Systeme
Fakultät für Informatik und Ingenieurwissenschaften - Institut für Informatik
Technology Arts Sciences TH Köln

Data Augmentation

Bilder - Übung

Training eines CNN mit Data Augmentation Schichten in keras:

- Nutzen Sie eine GPU und fügen Sie weitere Data Augmentation Schichten zu dem CNN hinzu
  - Bspw.:
    - Änderung des Kontrasts
    - Quelle: https://www.tensorflow.org/tutorials/images/data_augmentation
- Bei Bedarf auch ohne Data Augmentation trainieren, um einen Vergleich zu haben
  - Ab Epoche 8 startet leichtes Overfitting
  - Genauigkeit bei Train und Val aber > 91 %
- Auf was muss man bei der Nutzung von Data Augmentation achten?
  - Denken Sie dabei bspw. an den MNIST Datensatz…



Data Augmentation

Bilder

Albumentations, https://albumentations.ai/

- Daten werden Batch für Batch augmentiert. Während der eine Batch für das Training des CNNs genutzt wird, wird der nächste Batch augmentiert und dem CNN zur Verfügung gestellt.
  Das CNN wird nicht verändert.

```
# 3. Albumentations Transformationen definieren
transforms = A.Compose([
    A.HorizontalFlip(p=0.5),
    A.Rotate(limit=10, p=0.5),
    A.RandomBrightnessContrast(p=0.2),
])
```

Prof. Dr. Daniel Gaida
Professor für Cyber-Physische Systeme
Fakultät für Informatik und Ingenieurwissenschaften - Institut für Informatik

Bildquelle: https://albumentations.ai/

Data Augmentation

Bilder – Übung mit Albumentations

Öffnen Sie das Notebook 03_fmnist_albumentations.ipynb

- Welche weiteren Data Augmentation Transformationen können Sie hinzufügen?
  - https://albumentations.ai/docs/3-basic-usage/image-classification/#mnist-style-pipeline-grayscale-28x28

- Lassen Sie sich bei Bedarf diese Zeile von einer LLM erklären:

```
train_ds = train_ds.shuffle(1000).map(process_data,
num_parallel_calls=tf.data.AUTOTUNE).batch(batch_size).prefetch(tf.data.AUTOTUN
E)
```


Prof. Dr. Daniel Gaida
Professor für Cyber-Physische Systeme
Fakultät für Informatik und Ingenieurwissenschaften - Institut für Informatik
Technology Arts Sciences TH Köln

Data Augmentation

Text – Nutzung von LLMs (Stand der Technik bis 2023)

Transformationen

- Back Translation:
  - Übersetzen des Textes in eine andere Sprache und dann zurück in die Originalsprache.

- Mask-and-Infill
  - Maskiere Teile des Textes und lasse ein LLM diese ersetzen.
  - Beispiel:
    - Der Kunde war [MASK] zufrieden.
    - → Der Kunde war außerordentlich zufrieden.

- Contextual Word Replacement
  - Wörter werden durch Synonyme ersetzt, allerdings abhängig vom Kontext (ein LLM entscheidet)

Prof. Dr. Daniel Gaida
Professor für Cyber-Physische Systeme
Fakultät für Informatik und Ingenieurwissenschaften - Institut für Informatik

Bildquelle: Wagner, M., Jagayat, J., Kumar, A., Shirazi, A., Alavi, N., & Omrani, M. (2023). Using natural language processing as a scalable mental status evaluation technique. *medRxiv*, 2023-12.

Data Augmentation

Text – Aktueller Stand der Technik

Transformationen

- LLM-basierte Paraphrasierung
  - Prompt:
    - Erzeuge 10 semantisch äquivalente Varianten.
  - Beispiel:
    - Original: „Die Lieferung kam zu spät.“
    - Varianten: „Die Zustellung erfolgte verspätet.“, "Das Paket traf später als erwartet ein.„
- Synthetic Data Generation
  - Statt bestehende Daten zu verändern: LLM erzeugt komplett neue Beispiele.
  - Beispiel:
    - Intent: Passwort zurücksetzen
  - LLM generiert 1000 neue Nutzeranfragen
    - unterschiedliche Schreibweisen
    - unterschiedliche Detailgrade

Prof. Dr. Daniel Gaida
Professor für Cyber-Physische Systeme
Fakultät für Informatik und Ingenieurwissenschaften - Institut für Informatik
Technology Arts Sciences TH Köln

Data Augmentation

Text – Aktueller Stand der Technik

Retrieval-basierte Augmentation

- Idee:
  - Suche ähnliche Dokumente.
- Dann:
  - extrahiere Muster
  - generiere neue Varianten

- Wichtig, damit die generierten Dokumente auch wirklich typisch für die Klasse sind

- Beispiel:
  - Wir haben zu wenig E-Mails für Klasse Christentum
  - Generation von weiteren E-Mails dieser Klasse durch ein LLM durch Übergabe aller Mails der Klasse an das LLM.

Prof. Dr. Daniel Gaida
Professor für Cyber-Physische Systeme
Fakultät für Informatik und Ingenieurwissenschaften - Institut für Informatik
Technology Arts Sciences TH Köln

Data Augmentation

Text - Übung

Öffnen Sie das Notebook: 07_text_data_augmentation.ipynb

Generieren Sie weitere Nachrichten einer Klasse Ihrer Wahl und prüfen Sie die Sinnhaftigkeit und Passgenauigkeit der generierten Nachrichten.

Bei Bedarf:

- Verbessern Sie den Prompt

Data Augmentation

Text – Qualitätskontrolle

Nach der Generierung müssen Sie die Qualität der Sätze prüfen lassen:

- Semantic Similarity Filter
  - Embeddings vergleichen: Cosine Similarity

- NLI-basierte Validierung (Natural Language Inference)
  - Prüft ob der generierte Satz aus dem ursprünglichen Satz folgt (Entailment) oder diese widersprüchlich sind (Contradiction)
    - Widersprüchliche Sätze könnten eine hohe Cosine similarity haben, wenn sie sich nur um ein Wort unterscheiden („nicht“)

- Diversity Filtering
  - Verhindert:
    - Duplikate, Near-Duplicates
  - Typisch: Clustering der Embeddings

Prof. Dr. Daniel Gaida
Professor für Cyber-Physische Systeme
Fakultät für Informatik und Ingenieurwissenschaften - Institut für Informatik
Technology Arts Sciences TH Köln

Übung: Embedding Modell angeben und ausprobieren

- Cosine similarity liefert Werte zwischen -1 und 1 zurück (üblicherweise aber eher zwischen 0 und 1)

```python
# this cell is optional - not needed for the notebook to run
# you can play with this if you want to test how close two embedding vector are
embedding_1 = embed_model.get_text_embedding("Hello, my name is Daniel")
embedding_2 = embed_model.get_text_embedding("How are you?")

# Compute dot product similarity
similarity = sum(a * b for a, b in zip(embedding_1, embedding_2))
print(similarity)
```

```
0.8202128970159881
```

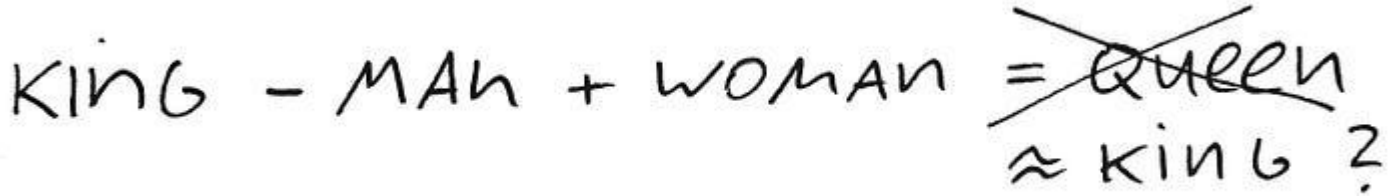
```python
# this cell is optional - not needed for the notebook to run
# you can play with this if you want to test how close two embedding vector are
embedding_1 = embed_model.get_text_embedding("Hello, my name is Daniel")
embedding_2 = embed_model.get_text_embedding("Hello, my name is not Daniel")

# Compute dot product similarity
similarity = sum(a * b for a, b in zip(embedding_1, embedding_2))
print(similarity)
```

```
0.9513346940599776
```


Prof. Dr. Daniel Gaida
Professor für Cyber-Physische Systeme
Fakultät für Informatik und Ingenieurwissenschaften - Institut für Informatik
Technology Arts Sciences TH Köln

Übung: Embedding Modell angeben und ausprobieren

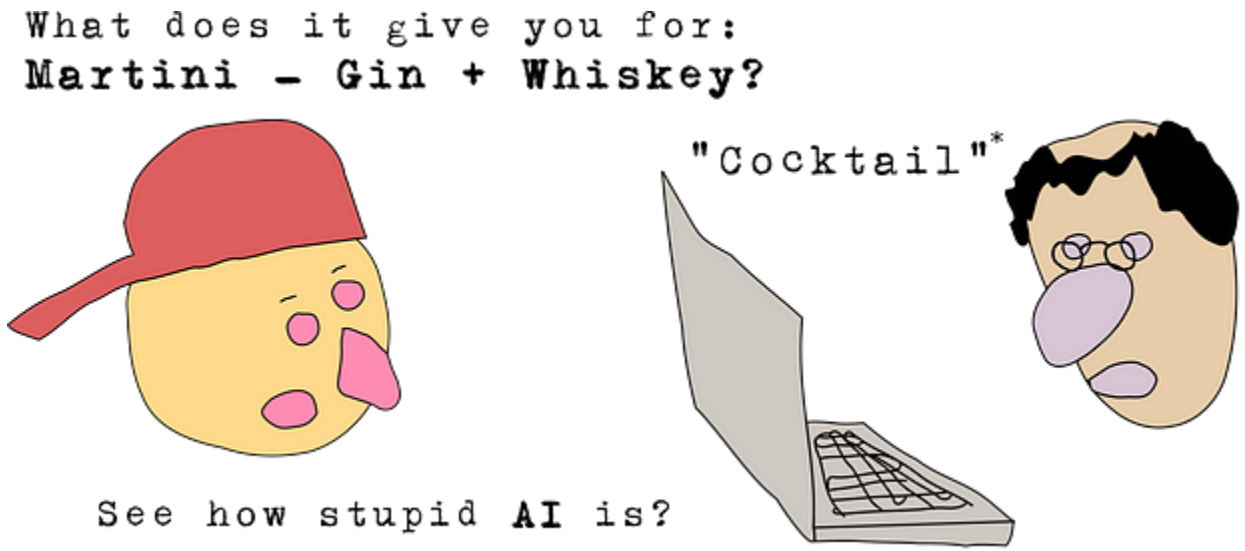


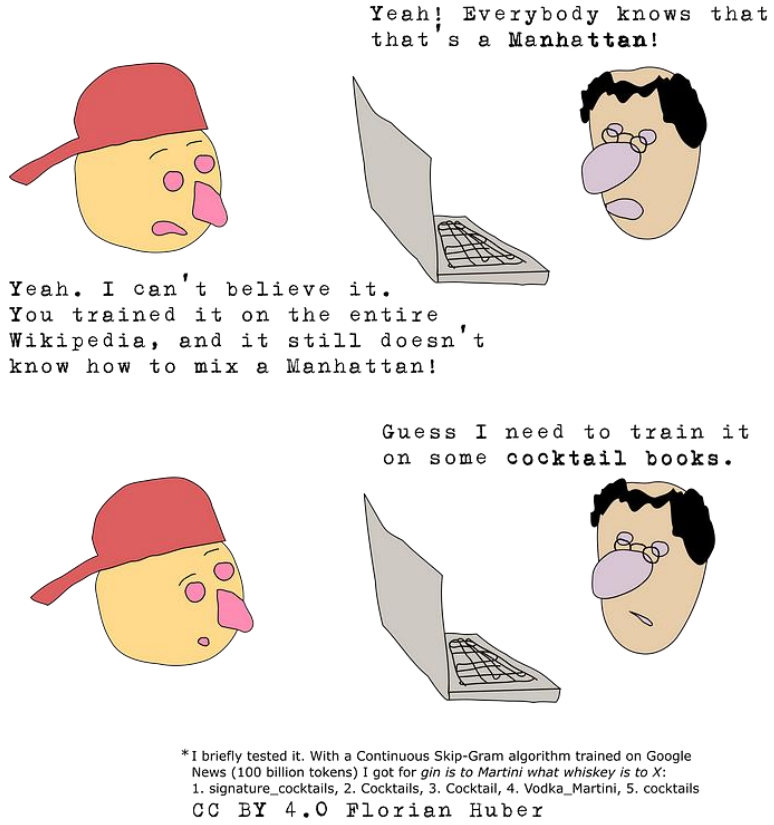

```
import numpy as np

# Convert lists to NumPy arrays before performing arithmetic
embedding_1_np = np.array(embed_model.get_text_embedding("King")) -
np.array(embed_model.get_text_embedding("Man")) +
np.array(embed_model.get_text_embedding("Woman"))
embedding_2_np = np.array(embed_model.get_text_embedding("Queen"))

# Compute dot product similarity using NumPy's dot function
similarity = np.dot(embedding_1_np, embedding_2_np)
print(similarity)
```

Übung: Embedding Modell angeben und ausprobieren





*I briefly tested it. With a Continuous Skip-Gram algorithm trained on Google News (100 billion tokens) I got for *gin is to Martini what whiskey is to X*:
1. signature_cocktails, 2. Cocktails, 3. Cocktail, 4. Vodka_Martini, 5. cocktails

Prof. Dr. Daniel Gaida
Professor für Cyber-Physische Systeme
Fakultät für Informatik und Ingenieurwissenschaften - Institut für Informatik

https://blog.esciencecenter.nl/king-man-woman-king-9a7fd2935a85

Data Augmentation

Text – Qualitätskontrolle - Übung

Öffnen Sie das Notebook 04_HF_EmbeddingModel_test.ipynb

Beachten Sie den „Restart Session“ Button am Ende der Ausgaben der ersten Zelle.

Vergleichen Sie unterschiedliche Texte über die cosine similarity ihrer Embedding Vektoren.

Überprüfen Sie ob Negationen eine hohe similarity haben.

Untersuchen Sie bei Bedarf unterschiedliche Embedding Modelle

Prof. Dr. Daniel Gaida
Professor für Cyber-Physische Systeme
Fakultät für Informatik und Ingenieurwissenschaften - Institut für Informatik
Technology Arts Sciences TH Köln

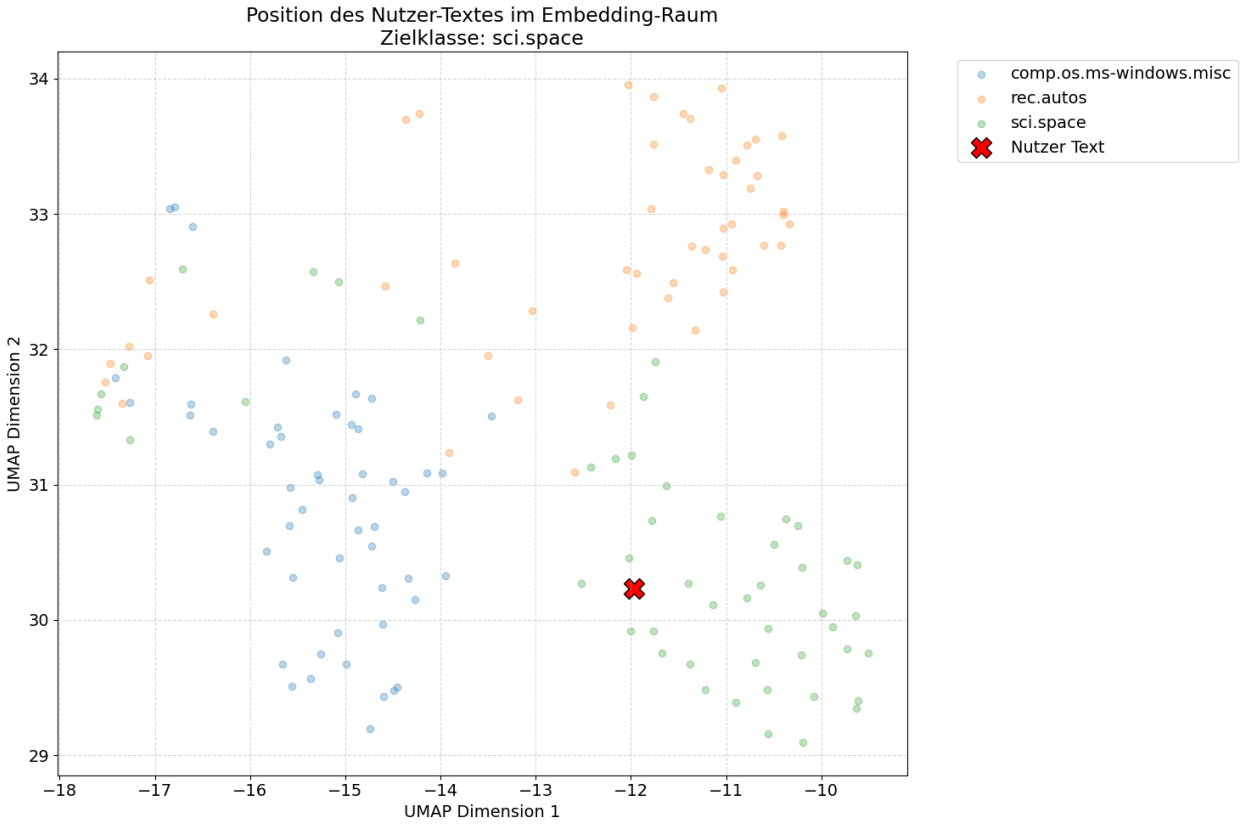
Data Augmentation

Text – Qualitätskontrolle - Übung

Prüfen Sie wie gut mit LLM generierter Text (07_text_data_augmentation.ipynb) zu den Texten der Klasse passt.

Notebook: 08_embedding_visualization.ipynb

Sie können sich das trainierte UMAP Modell und die Embeddings hier herunterladen und in google colab hochladen –
das spart 3-4 Minuten Ausführungszeit des Notebooks.



Prof. Dr. Daniel Gaida
Professor für Cyber-Physische Systeme
Fakultät für Informatik und Ingenieurwissenschaften - Institut für Informatik

Data Augmentation
Zeitreihen

Transformation	Idee
Jittering	kleines Gaußsches Rauschen hinzufügen
Scaling	Werte leicht skalieren
Time Warping	Zeitachse lokal dehnen oder stauchen
Window Slicing	zufälligen Ausschnitt verwenden

- Wichtig: Transformationen müssen die **zeitliche Struktur** erhalten.
- Gute Augmentation erzeugt realistische Variationen, ohne die Bedeutung der Zeitreihe zu verändern.
- Tool: tsaug
  - https://tsaug.readthedocs.io/en/stable/

Prof. Dr. Daniel Gaida
Professor für Cyber-Physische Systeme
Fakultät für Informatik und Ingenieurwissenschaften - Institut für Informatik
Technology Arts Sciences TH Köln

Data Augmentation

Zusammenfassung

- Data Augmentation vergrößert den Trainingsdatensatz durch leichte Variationen der Trainingsdatenpunkte
- Data Augmentation niemals auf Testdaten anwenden!
- Automatisierte Qualitätskontrolle wichtig
  - Prüft, ob der generierte Datenpunkt in der gleichen Klasse wie sein Original liegt

Prof. Dr. Daniel Gaida
Professor für Cyber-Physische Systeme
Fakultät für Informatik und Ingenieurwissenschaften - Institut für Informatik
Technology Arts Sciences TH Köln

Lernraum III

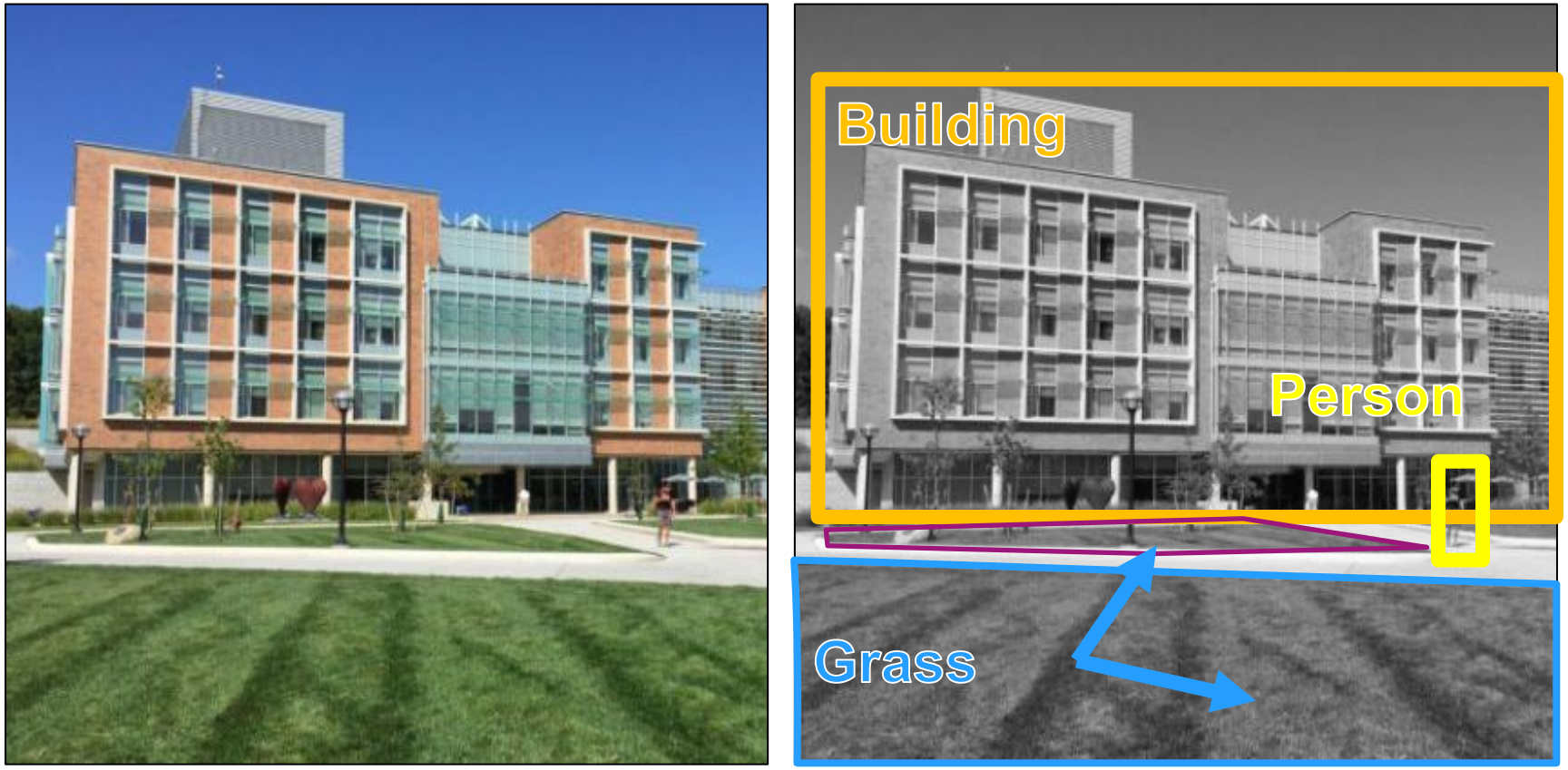
Inhalt

- Data Augmentation – synthetische Vergrößerung des Trainingsdatensatzes
  - Bilder
  - Text
  - Zeitreihen (future)

- **Bildverarbeitung mit KI Algorithmen, die über Bildklassifikation hinausgeht**
  - Objektdetektion
  - Objekttracking
  - Open-Vocabulary Objektdetektion
  - Bildsegmentierung
  - Open-Vocabulary Image Segmentation

Objektdetektion

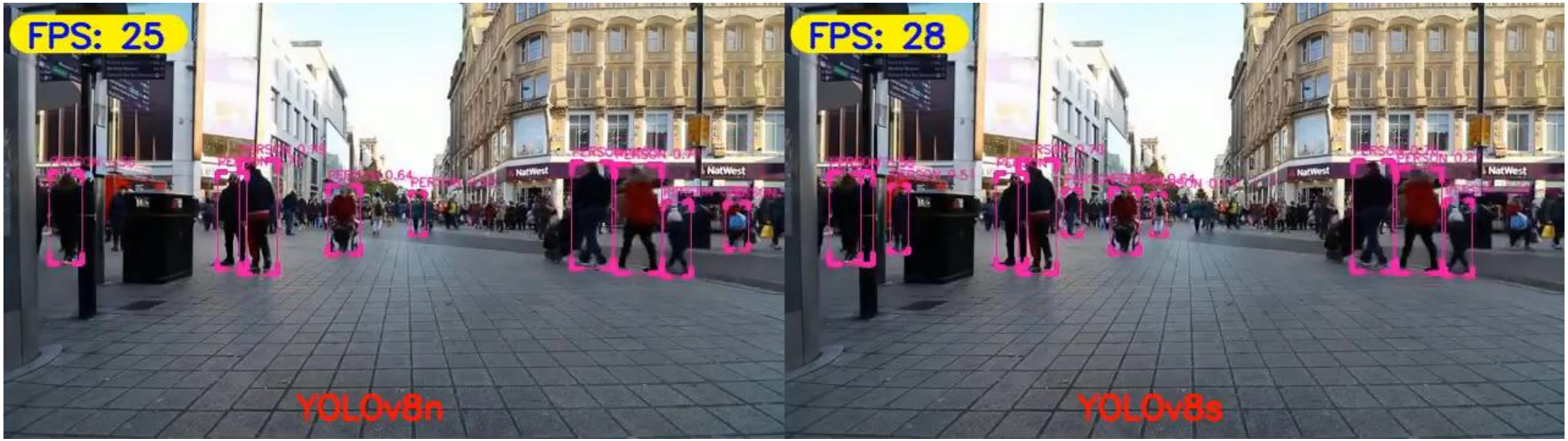
Ziel: Benennung der Objekte in der Szene



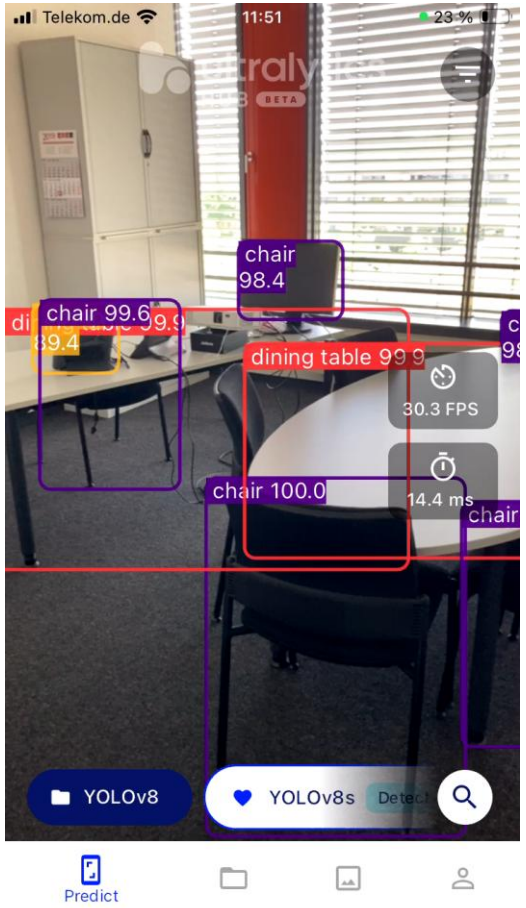
Prof. Dr. Daniel Gaida
Professor für Cyber-Physische Systeme
Fakultät für Informatik und Ingenieurwissenschaften - Institut für Informatik

Quelle: Computer Vision (EECS 442), Uni Michigan, Winter 21

Objektdetektion



Objektdetektion



- YOLOv8
- Erkennt nur 80 verschiedene Objektkategorien (Autos, Fahrräder, Tiere, Regenschirme, …)

https://github.com/ultralytics/ultralytics

Prof. Dr. Daniel Gaida
Professor für Cyber-Physische Systeme
Fakultät für Informatik und Ingenieurwissenschaften - Institut für Informatik

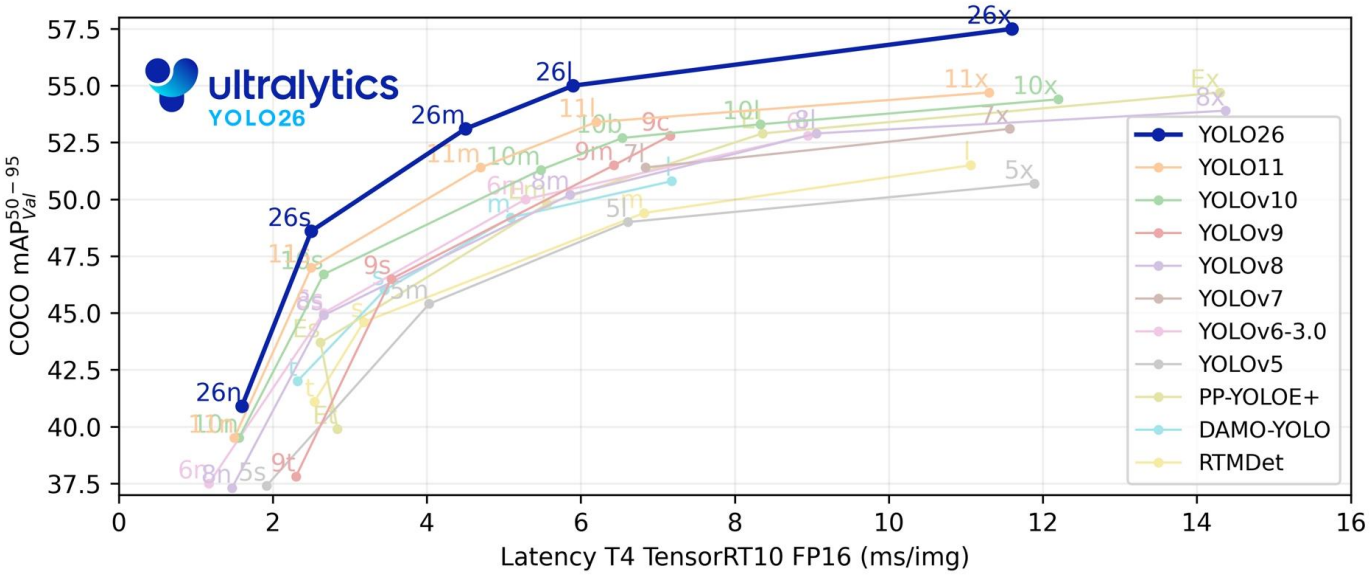
Objektdetektion

Deep Learning Modell

- YOLO – You only look once
- Redmon, J., & Farhadi, A. (2015). You only look once: Unified, real-time object detection. In Proc. IEEE CVPR (pp. 779-788).

- YOLO26 – plug & play
  https://github.com/ultralytics/ultralytics

- Fertig trainierte Modelle in unterschiedlichen Größen herunterladbar
- Trainiert an COCO Datensatz
  - 80 verschiedene Objektkategorien (Autos, Fahrräder, Tiere, Regenschirme, …)



Prof. Dr. Daniel Gaida
Professor für Cyber-Physische Systeme
Fakultät für Informatik und Ingenieurwissenschaften - Institut für Informatik

Bild: https://docs.ultralytics.com/models/yolo26#overview

Objektdetektion

Bevor Sie ein Deep Learning Modell selbst erstellen und trainieren:

- Es gibt sehr viele an sehr großen Datensätzen trainierte Modelle, die Sie herunterladen und direkt nutzen können
- Sie können dieses Modell auch als initiales Modell nutzen, welches Sie mit Ihren eigenen Daten verbessern können
  - Transfer Learning
  - Knowledge Distillation

- YOLO26 als Beispiel

- https://huggingface.co/models
- https://modelzoo.co/

Objektdetektion

Trainingsdatensatz

```
0 0.0414228 0.659696 0.0828456 0.299007
11 0.0199709 0.268372 0.0397828 0.0651804
0 0.349451 0.584807 0.152251 0.418149
0 0.913659 0.58147 0.172332 0.45871
```

Trainingsdaten

- Bilder

Markieren der Bilder (Labeling)

- Um jedes zu detektierende Objekt ist ein Rahmen zu zeichnen
- Pro Bild können mehrere Objekte markiert werden

Label

- Koordinaten des Rahmens
- Klasse des Objekts
- Für jedes Bild stehen Label in einer txt-Datei

https://www.edgeimpulse.com/blog/3-ways-to-do-ai-assisted-labeling-for-object-detection

https://www.youtube.com/watch?v=gRAyOPjQ9_s
Sehr gutes Tutorial zu YOLOv8 Training und Inferenz (15 Min.)

Objektdetektion mit Deep Learning

Kostenlose Tools, die das Labeling beschleunigen

- https://www.cvat.ai/
- https://roboflow.com/#annotate
- https://pypi.org/project/labelImg/
- …

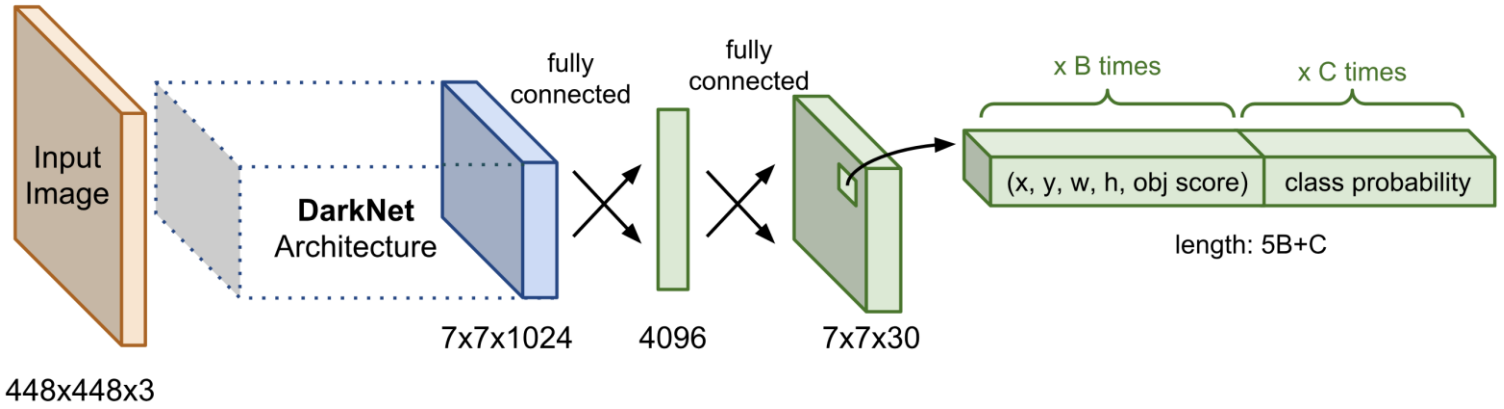
Bereits gelabelte Datensätze (Erster Schritt):

- https://paperswithcode.com/datasets?task=object-detection&page=1 (233 Object Det. Datensätze)
- Kaggle, roboflow, …
- https://datasetsearch.research.google.com/
- Wenn nicht genau der richtige Datensatz dabei ist, hilft es meistens dennoch das Modell auf dem ähnlichsten Datensatz zu trainieren und das Modell dann auf einem eigenen Datensatz weiter zu trainieren (Transfer Learning, finetuning)

Objektdetektion mit Deep Learning

Training – sehr viele Bilder und deren Label

0 0.0414228 0.659696 0.0828456 0.299007
11 0.0199709 0.268372 0.0397828 0.0651804
0 0.349451 0.584807 0.152251 0.418149
0 0.913659 0.58147 0.172332 0.45871

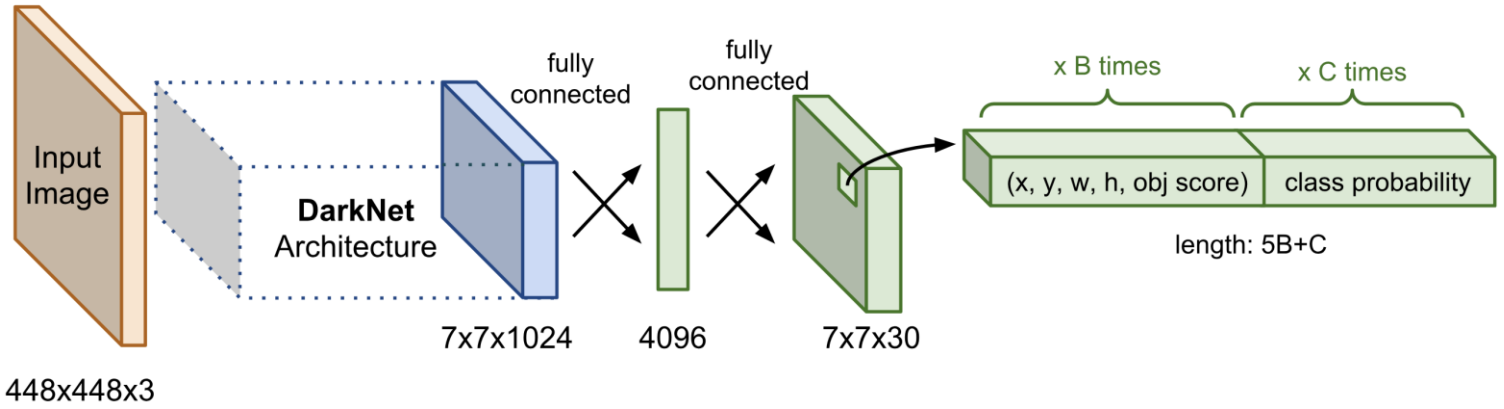


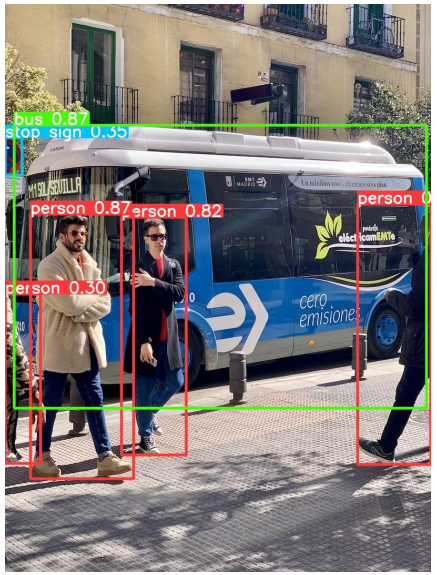
Prof. Dr. Daniel Gaida
Professor für Cyber-Physische Systeme
Fakultät für Informatik und Ingenieurwissenschaften - Institut für Informatik

https://lilianweng.github.io/posts/2018-12-27-object-recognition-part-4/

Objektdetektion mit Deep Learning

Inferenz





YOLOv11 Objektdetektion testen

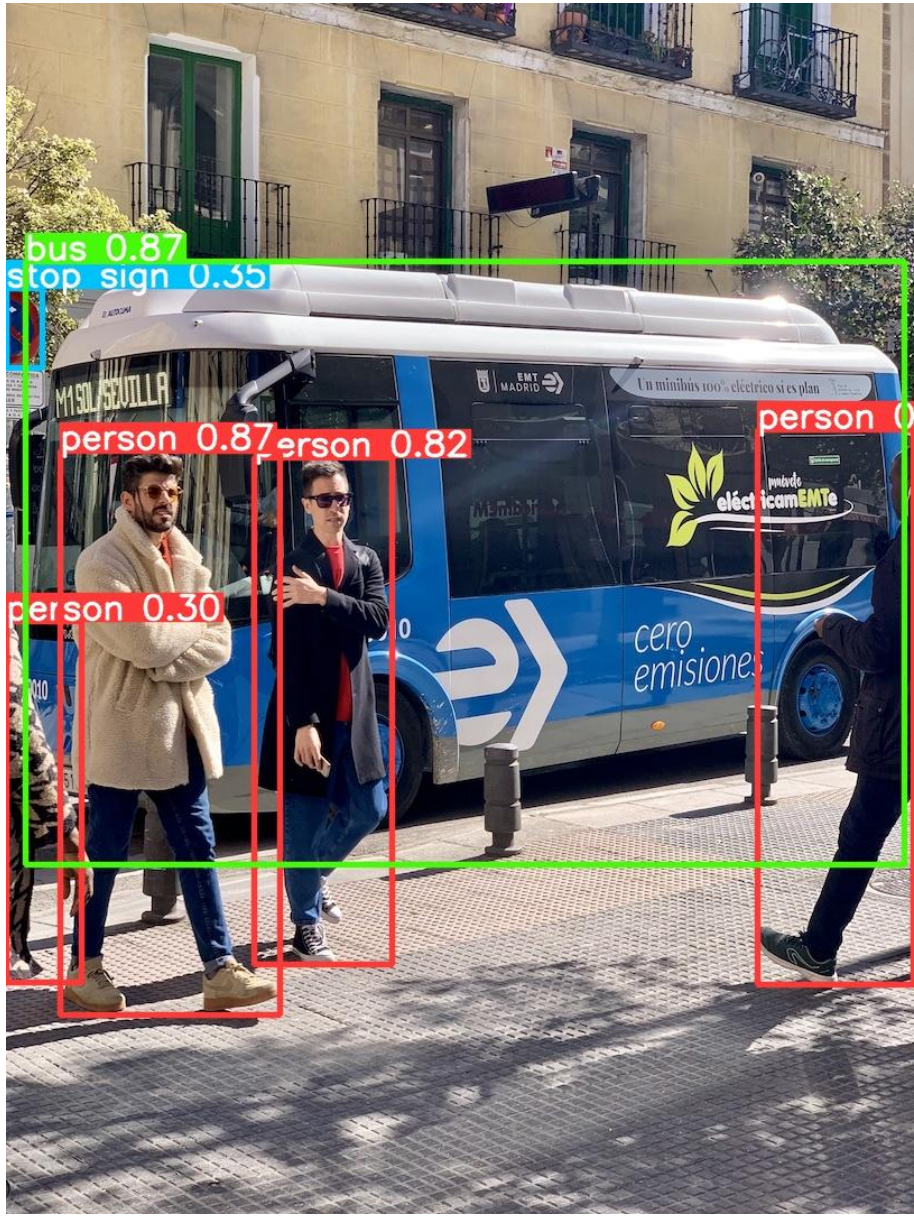
pip install ultralytics

from ultralytics import YOLO

model = YOLO("yolo26n.pt") # load an official model
# Predict with the model
results = model("https://ultralytics.com/images/bus.jpg")

https://docs.ultralytics.com/tasks/detect/

Siehe Notebook: 06_objektdetektion_yolo26.ipynb



Prof. Dr. Daniel Gaida
Professor für Cyber-Physische Systeme
Fakultät für Informatik und Ingenieurwissenschaften - Institut für Informatik
Technology Arts Sciences TH Köln

Lernraum III

Inhalt

- Data Augmentation – synthetische Vergrößerung des Trainingsdatensatzes
  - Bilder
  - Text
  - Zeitreihen (future)

- **Bildverarbeitung mit KI Algorithmen, die über Bildklassifikation hinausgeht**
  - Objektdetektion
  - **Objekttracking**
  - Open-Vocabulary Objektdetektion
  - Bildsegmentierung
  - Open-Vocabulary Image Segmentation

Prof. Dr. Daniel Gaida
Professor für Cyber-Physische Systeme
Fakultät für Informatik und Ingenieurwissenschaften - Institut für Informatik
Technology Arts Sciences TH Köln

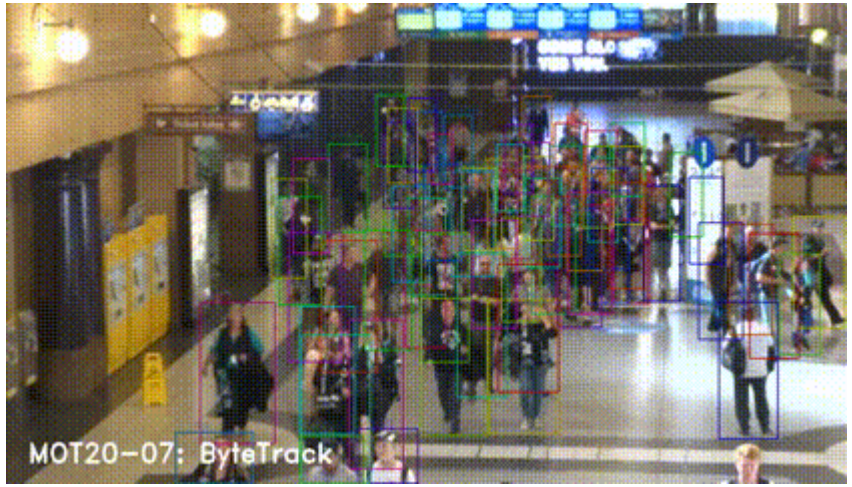
Objekt Tracking

Objekt Tracking

- Eindeutige Objekt Detektion von beweglichen Objekten über eine Sequenz von Frames

YOLO26 hat Objekt Tracking integriert

https://docs.ultralytics.com/modes/track/



https://github.com/ifzhang/ByteTrack

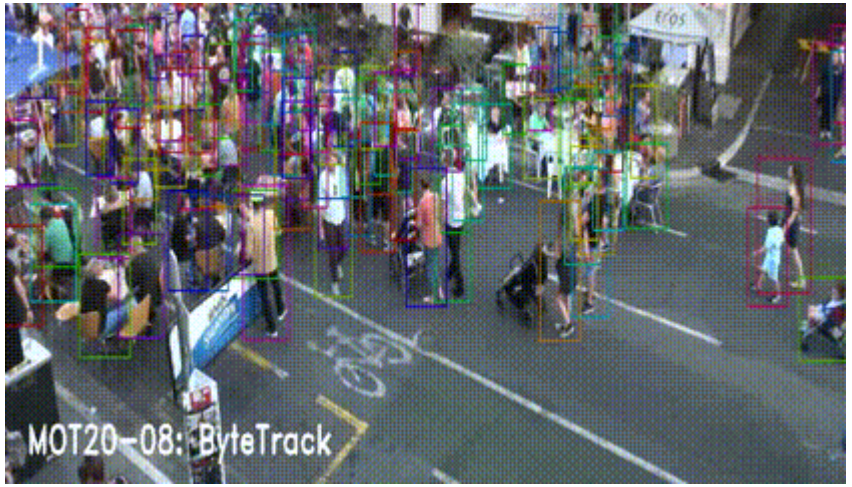
Sehr gutes Tutorial von Roboflow zu Objekt Tracking mit YOLOv8 (14 Min.):

https://www.youtube.com/watch?v=Mi9iHFd0_Bo

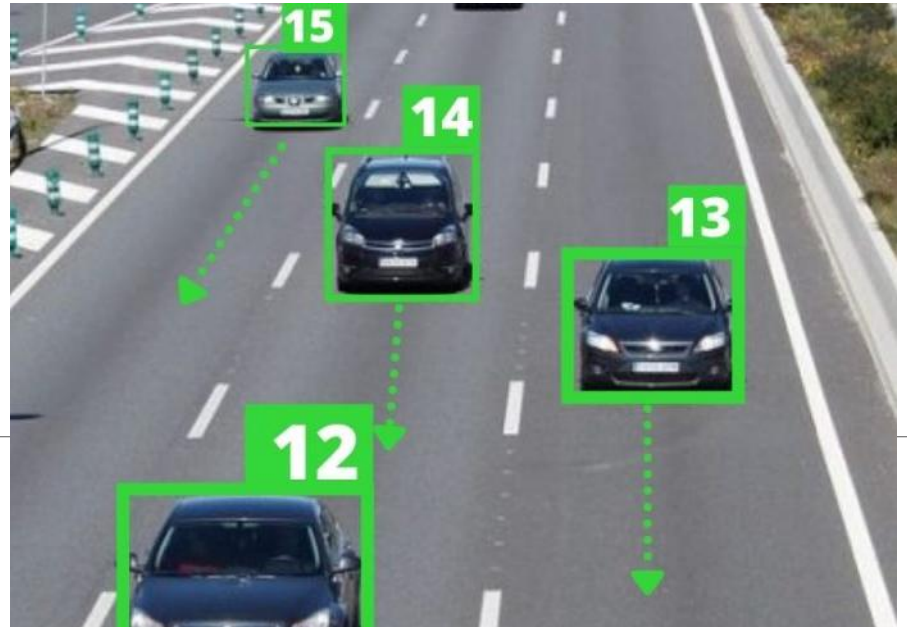
Objekt Tracking

Vorgehensweise:

- Tracking-by-Detection
  - Objekt Detektion (bspw. mit YOLO)
  - Tracking der detektierten Objekte
    - Kalman Filter prädiziert wo Objekt in nächstem Frame sein könnte. In dieser Region wird dann für das nächste Frame Objekt Detektion ausgeführt.



https://github.com/ifzhang/ByteTrack

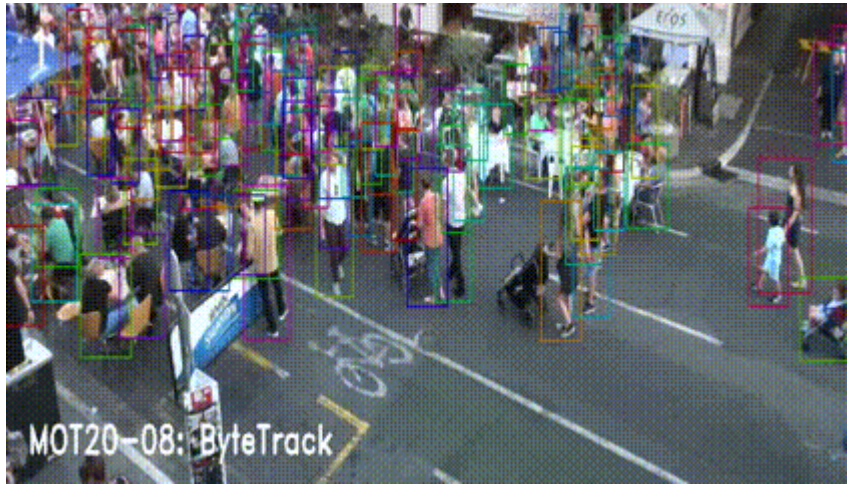


https://pysource.com/2021/11/02/kalman-filter-predict-the-trajectory-of-an-object/

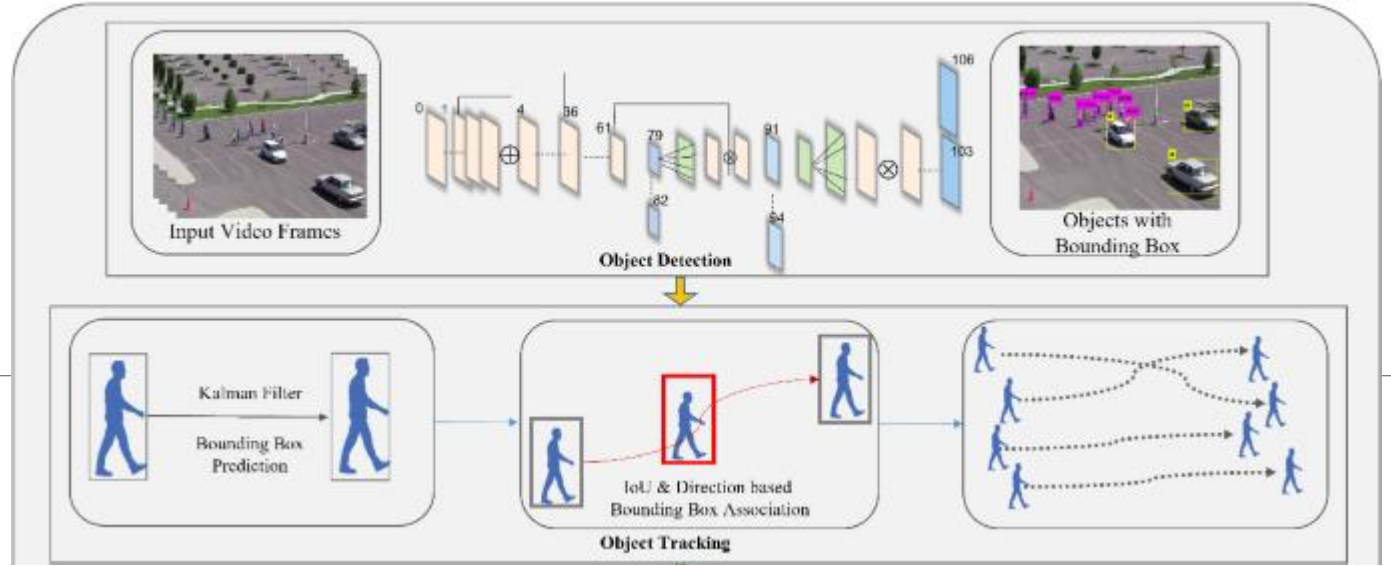
Objekt Tracking

Vorgehensweise:

- Tracking-by-Detection
  - Objekt Detektion (bspw. mit YOLO)
  - Tracking der detektierten Objekte
    - Kalman Filter prädiziert wo Objekt in nächstem Frame sein könnte. In dieser Region wird dann für das nächste Frame Objekt Detektion ausgeführt.
  - Re-Identifikation (ist das Objekt noch das gleiche?)



https://github.com/ifzhang/ByteTrack

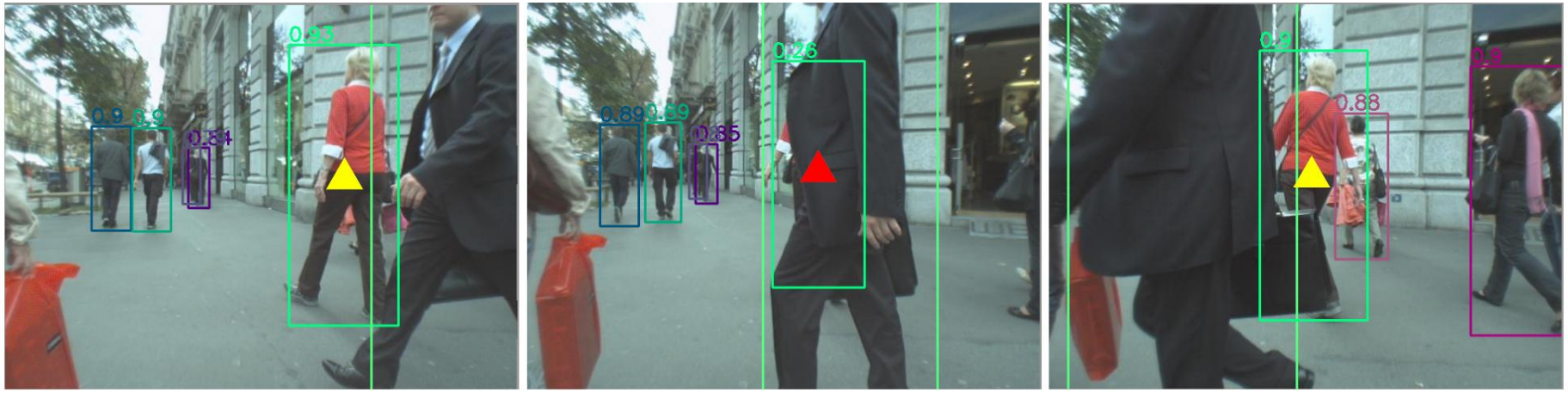


A. Patel et al., Motion-compensated online object tracking for activity detection and crowd behavior analysis, The visual Computer, 2022

Objekt Tracking

In YOLO26 integriert („State of the Art“ des Objekt Trackings):

- ByteTrack, https://github.com/ifzhang/ByteTrack
- Beibehalten von Bounding Boxen, die temporär nur einen geringen Score erhalten



Y. Zhang et al., ByteTrack: Multi-Object Tracking by Associating Every Detection Box, 2022

- BoT-SORT, https://github.com/NirAharon/BoT-SORT
  - Erweiterung von ByteTrack um Kompensation der Bewegung der Kamera, verbesserter Zustandsvektor des Kalman Filters, …

YOLO26 Objekttracking testen

```
from ultralytics import YOLO

model = YOLO("yolo26n.pt")  # Load an official Detect model

# Perform tracking with the model with ByteTrack
results = model.track("https://youtu.be/LNwODJXcvt4", show=True, tracker="bytetrack.yaml")
```

https://docs.ultralytics.com/modes/track/

s. Notebook: 09_objekttracking_yolo26.ipynb

Lernraum III

Inhalt

- Data Augmentation – synthetische Vergrößerung des Trainingsdatensatzes
  - Bilder
  - Text
  - Zeitreihen (future)

- **Bildverarbeitung mit KI Algorithmen, die über Bildklassifikation hinausgeht**
  - Objektdetektion
  - Objekttracking
  - **Open-Vocabulary Objektdetektion**
  - Bildsegmentierung
  - Open-Vocabulary Image Segmentation

Prof. Dr. Daniel Gaida
Professor für Cyber-Physische Systeme
Fakultät für Informatik und Ingenieurwissenschaften - Institut für Informatik
Technology Arts Sciences TH Köln

Objekt Detektion von beliebigen Objektkategorien

- Algorithmen der Objekt Detektion funktionieren nur für die Objektkategorien für die sie trainiert wurden
- YOLO wurde trainiert mit COCO Datensatz
  - 80 verschiedene Objektkategorien (Autos, Fahrräder, Tiere, Regenschirme, …)
- Mit Trainingsdaten von weiteren Objektkategorien können Sie auch diese detektieren
- Was ist, wenn Sie „beliebige“ Objektkategorien detektieren möchten?
  - Open-Vocabulary Objekt Detektion
  - Trainiert mit Millionen von Bildern aus dem Internet und deren Bildbeschriftungen
  - YOLOE - Real-Time Seeing Anything
    - https://docs.ultralytics.com/models/yoloe

Objekt Detektion von beliebigen Objektkategorien

- Open-Vocabulary Objekt Detektion
- https://github.com/ailab-cvc/yolo-world

Objekt Detektion von beliebigen Objektkategorien

```
from ultralytics import YOLO

model = YOLO("yoloe-11s-seg.pt")                    # Initialize a model
model.set_classes(["person", "bus"])                # Define custom classes

results = model.predict("path/to/image.jpg")        # Run inference to detect your custom classes
results[0].show()                                   # Show results
```

s. Notebook 05_AppObjectDetection.ipynb

Prof. Dr. Daniel Gaida
Professor für Cyber-Physische Systeme
Fakultät für Informatik und Ingenieurwissenschaften - Institut für Informatik
https://docs.ultralytics.com/models/yolo-world/

Lernraum III

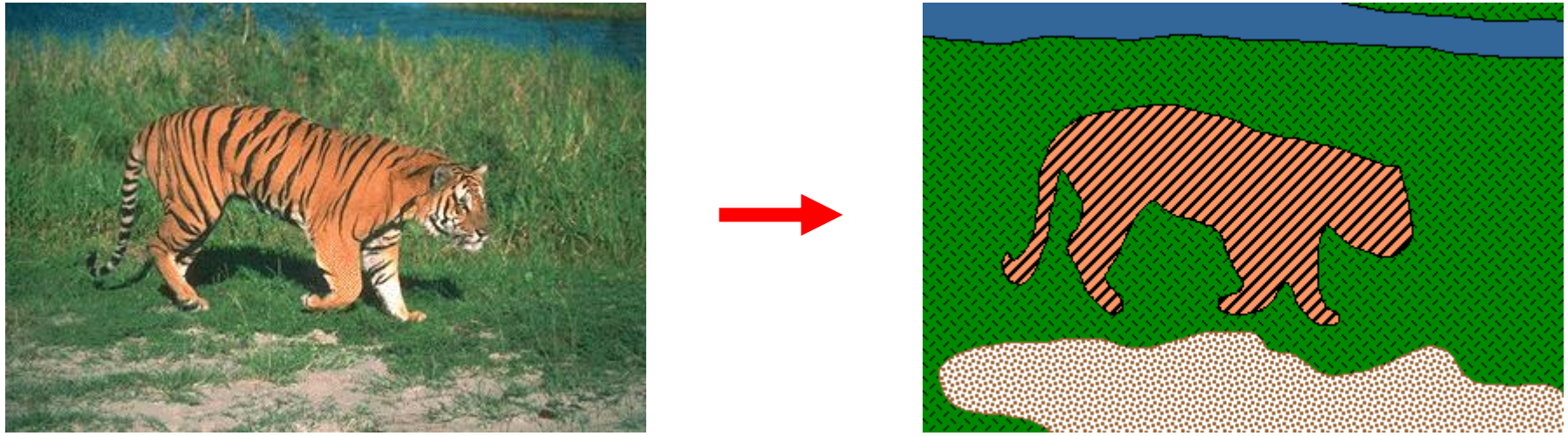
Inhalt

- Data Augmentation – synthetische Vergrößerung des Trainingsdatensatzes
  - Bilder
  - Text
  - Zeitreihen (future)

- **Bildverarbeitung mit KI Algorithmen, die über Bildklassifikation hinausgeht**
  - Objektdetektion
  - Objekttracking
  - Open-Vocabulary Objektdetektion
  - **Bildsegmentierung**
  - Open-Vocabulary Image Segmentation

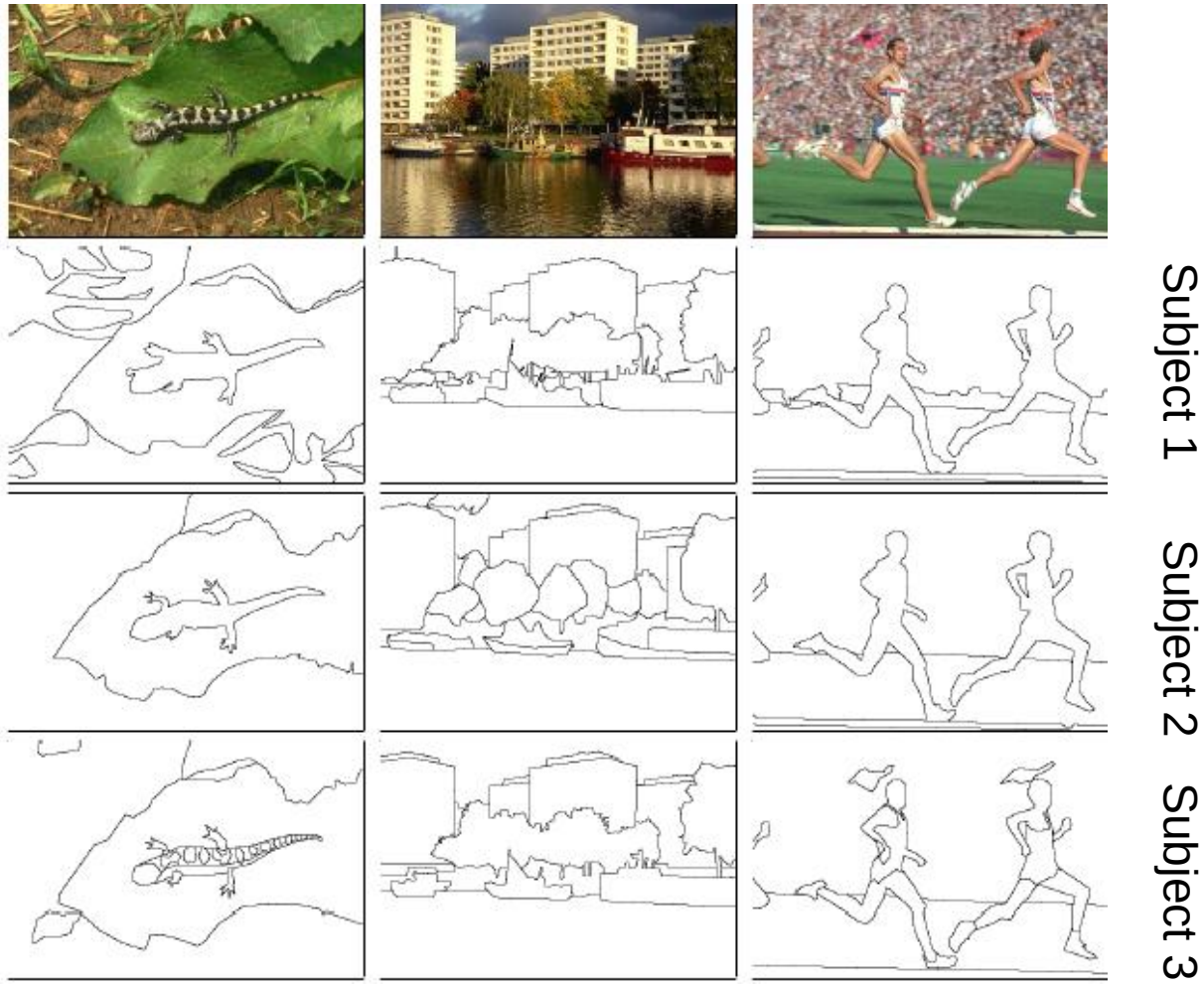
Prof. Dr. Daniel Gaida
Professor für Cyber-Physische Systeme
Fakultät für Informatik und Ingenieurwissenschaften - Institut für Informatik
Technology Arts Sciences TH Köln

Bildsegmentierung

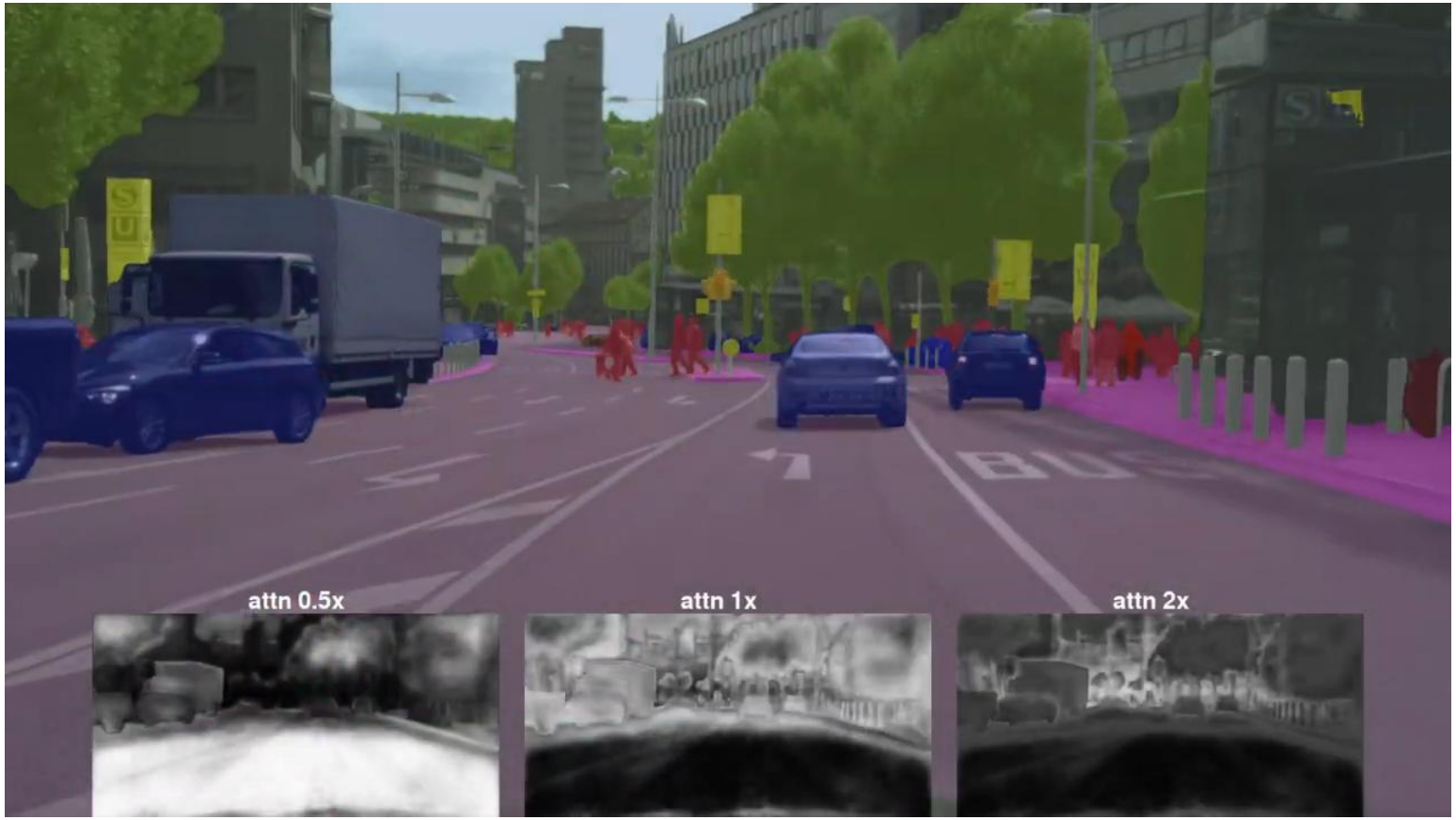


Bildsegmentierung

- Es gibt keine objektive Definition einer „richtigen“ Segmentierung!
- Segmentierung wird durch Menschen vorgegeben



Bildsegmentierung



github.com/NVIDIA/semantic-segmentation

A. Tao et al., Hierarchical Multi-Scale Attention for Semantic Segmentation, 2020

Prof. Dr. Daniel Gaida
Professor für Cyber-Physische Systeme
Fakultät für Informatik und Ingenieurwissenschaften - Institut für Informatik

Bildsegmentierung

Trainingsdaten

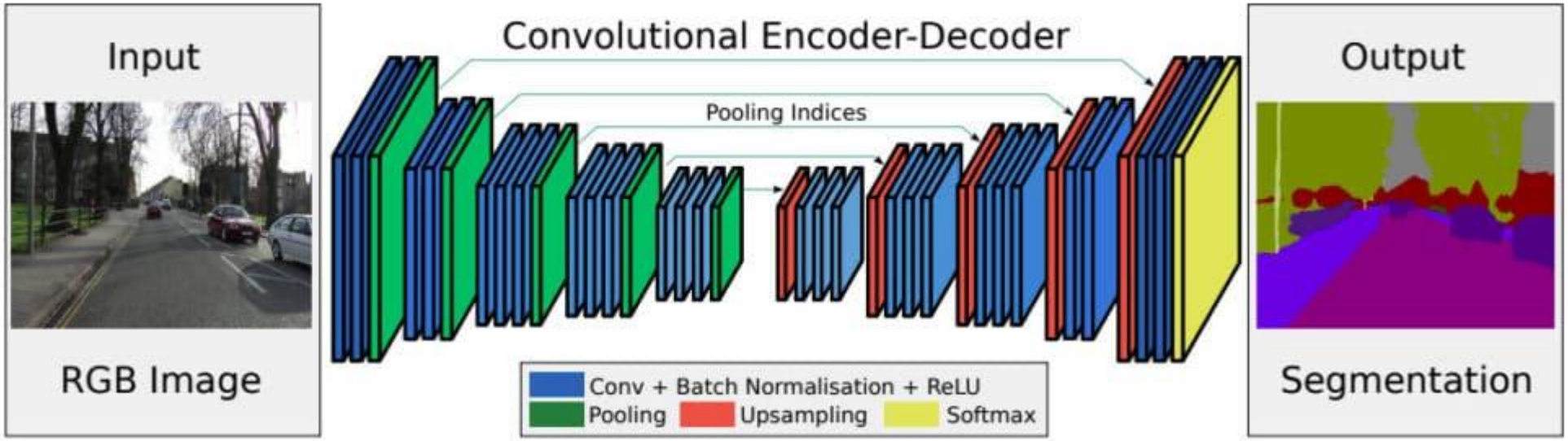
Eingangsdaten

Ausgangsdaten

Quelle für Trainingsdaten: https://paperswithcode.com/datasets?task=semantic-segmentation (261 Datensätze)

Bildsegmentierung mit Deep Learning

Training des Deep Learning Modells (bspw. Autoencoder)



V. Badrinarayanan et al., SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation, 2016

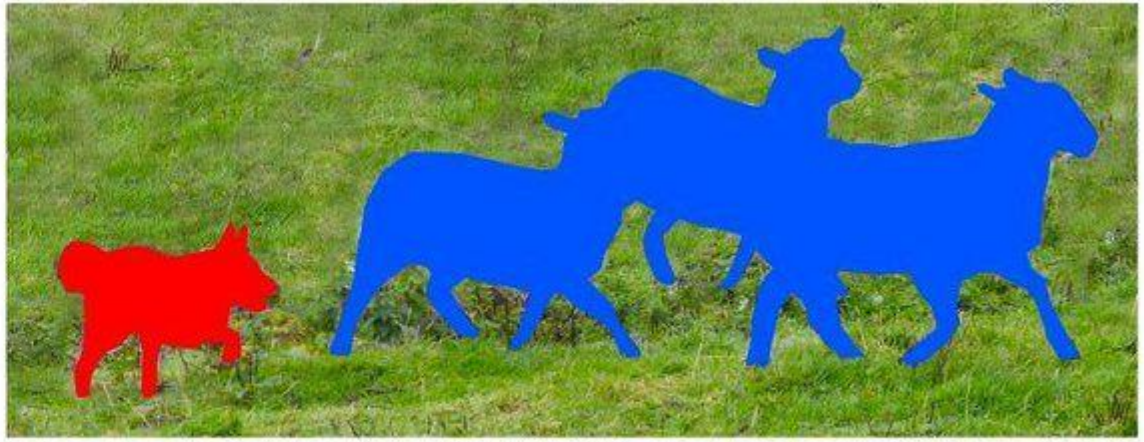
Prof. Dr. Daniel Gaida
Professor für Cyber-Physische Systeme
Fakultät für Informatik und Ingenieurwissenschaften - Institut für Informatik

Bildsegmentierung mit Deep Learning

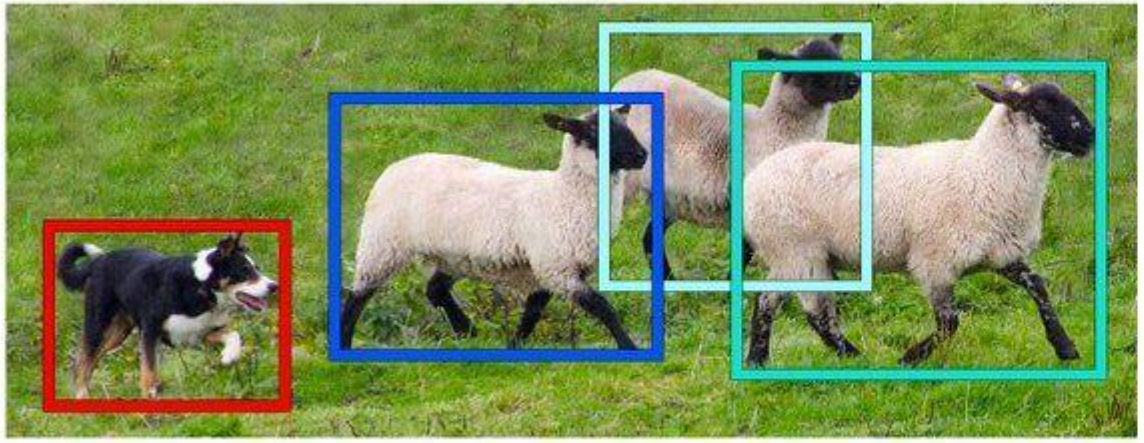
Instance Segmentation vs. Semantic Segmentation

https://blog.roboflow.com/difference-semantic-segmentation-instance-segmentation/

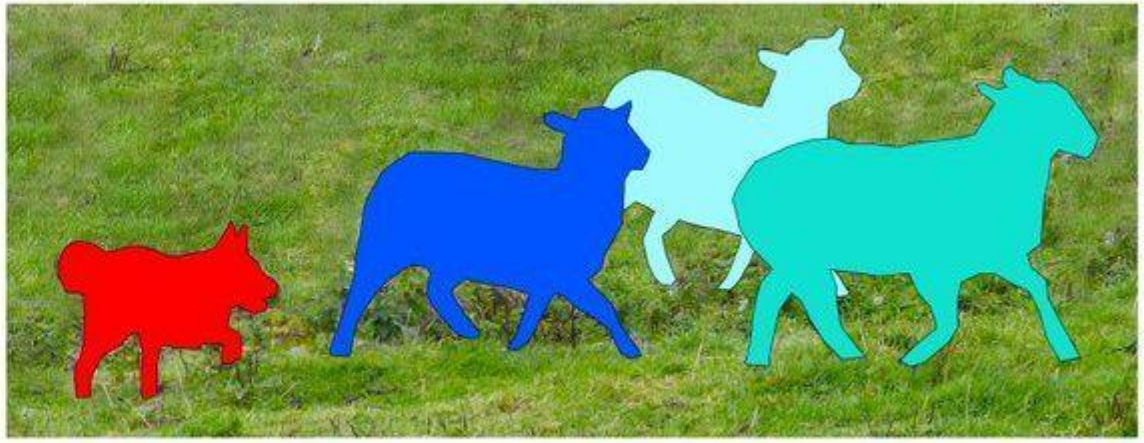
Image Recognition



Semantic Segmentation



Object Detection



Instance Segmentation

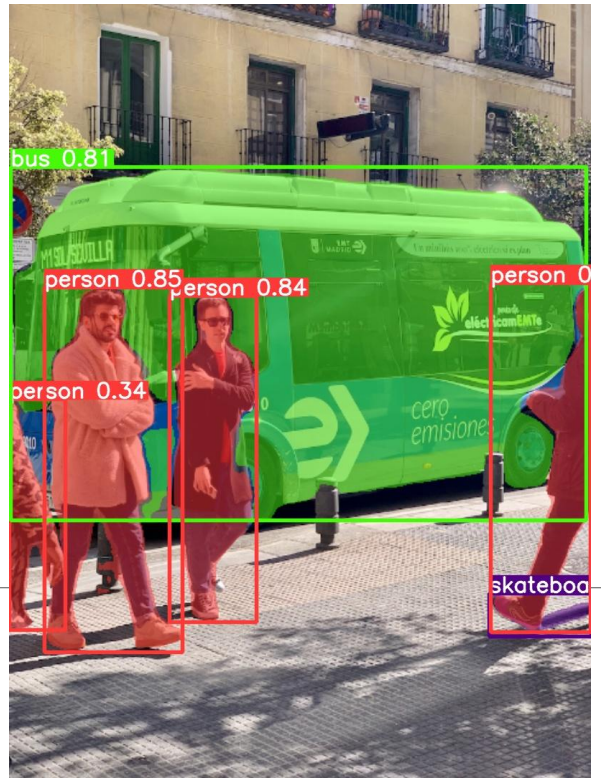
Bildsegmentierung mit Deep Learning

YOLO26 kann auch Instance Segmentation (https://docs.ultralytics.com/tasks/segment/)

Gutes Tutorial zu YOLOv8 zu Image Segmentation und Objekt Detektion, nur Inferenz (13 Min.)
https://www.youtube.com/watch?v=75LI9MI9eEo

```
from ultralytics import YOLO
model = YOLO("yolo26n-seg.pt")  # load a model
# Predict with the model
results = model("https://ultralytics.com/images/bus.jpg")
```

s. Notebook: 06_objektdetektion_yolo26.ipynb



Bildsegmentierung

Instance Segmentation mit YOLOv8

Lernraum III

Inhalt

- Data Augmentation – synthetische Vergrößerung des Trainingsdatensatzes
  - Bilder
  - Text
  - Zeitreihen (future)

- **Bildverarbeitung mit KI Algorithmen, die über Bildklassifikation hinausgeht**
  - Objektdetektion
  - Objekttracking
  - Open-Vocabulary Objektdetektion
  - Bildsegmentierung
  - **Open-Vocabulary Image Segmentation**

Prof. Dr. Daniel Gaida
Professor für Cyber-Physische Systeme
Fakultät für Informatik und Ingenieurwissenschaften - Institut für Informatik
Technology Arts Sciences TH Köln

Open-Vocabulary Image Segmentation

Segment Anything Modell (SAM 3)

- github.com/facebookresearch/segment-anything
- https://docs.ultralytics.com/models/sam-3
- Foundation Modell für Bildsegmentierung; Promptable

- FastSAM (https://docs.ultralytics.com/models/fast-sam/)
  - Yolov8 Segmentierungsmodell und anschließend Fokus auf die gewünschten Begriffe

- Grounded Segment Anything
  - Grounding DINO (Open-Vocabulary Objekt Detektion) – lokalisiert die Objekte; geg. Prompt
    - https://github.com/IDEA-Research/Grounding-DINO-1.5-API
  - Segment Anything Modell – segmentiert die lokalisierten Objekte
  - https://github.com/IDEA-Research/Grounded-SAM-2

Prof. Dr. Daniel Gaida
Professor für Cyber-Physische Systeme
Fakultät für Informatik und Ingenieurwissenschaften - Institut für Informatik

Bildquelle:
https://huggingface.co/docs/transformers/v4.46.0/en/model_doc/sam

Open-Vocabulary Image Segmentation

```
from ultralytics import FastSAM

# Define an inference source
source = "path/to/bus.jpg"
# Create a FastSAM model
model = FastSAM("FastSAM-s.pt")  # or FastSAM-x.pt
# Run inference on an image
everything_results = model(source, device="cpu", retina_masks=True, imgsz=1024,
conf=0.4, iou=0.9)

# Run inference with texts prompt
results = model(source, texts="a photo of a dog")
```


Prof. Dr. Daniel Gaida
Professor für Cyber-Physische Systeme
Fakultät für Informatik und Ingenieurwissenschaften - Institut für Informatik
https://docs.ultralytics.com/models/fast-sam/

YOLOv8 testen – auf dem Smartphone

Prof. Dr. Daniel Gaida
Professor für Cyber-Physische Systeme
Fakultät für Informatik und Ingenieurwissenschaften - Institut für Informatik
Technology Arts Sciences TH Köln

Entwicklung der Bilderkennung mit Deep Learning Modellen

Generation	Fähigkeit
CNN	Bildklassifikation
YOLO	Objektdetektion
ByteTrack	Objekttracking
YOLO-Seg	Segmentierung
SAM 3	Prompt-basierte Segmentierung
Vision-Language-Modelle	Offene Objekterkennung (GPT-4o, …)
Generative Vision Models	Bildverstehen und Bildgenerierung

Prof. Dr. Daniel Gaida
Professor für Cyber-Physische Systeme
Fakultät für Informatik und Ingenieurwissenschaften - Institut für Informatik
Technology Arts Sciences TH Köln

Generative Vision Models - Vision Banana

Bildgenerierung und -verstehen

- Gabeur, V. et al., Image Generators are Generalist Vision Learners, **2026**

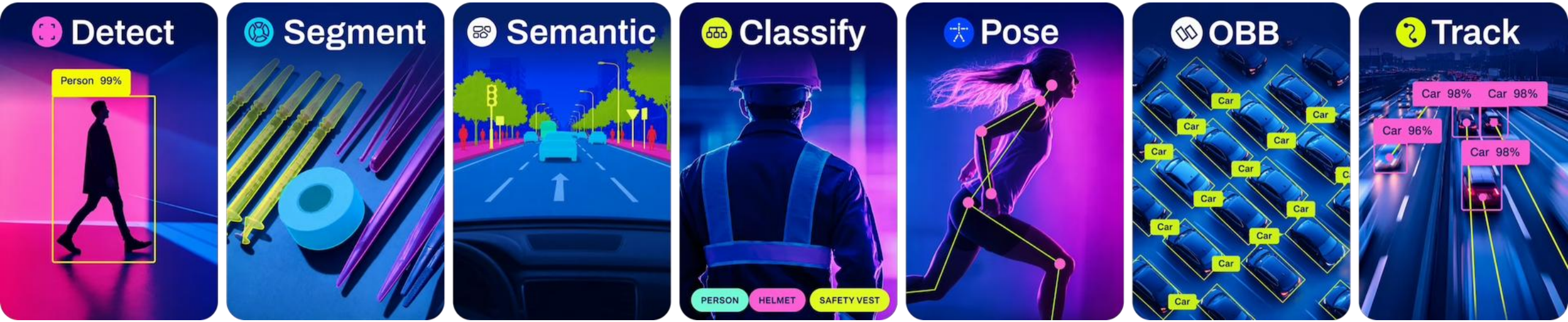
- “Vision Banana is a SOTA unified model for both image understanding and generation.”
- “Generative vision pretraining is an effective paradigm for visual understanding.”
- “Image generation serves as a universal interface for diverse vision tasks.”

- S. Beispiele in https://vision-banana.github.io/#capabilities

Objektdetekion, -tracking und -segmentierung

Zusammenfassung

- Mit YOLO-Modellen von Ultralytics kann man viele Bilderkennungs-Anwendungen recht einfach starten



Prof. Dr. Daniel Gaida
Professor für Cyber-Physische Systeme
Fakultät für Informatik und Ingenieurwissenschaften - Institut für Informatik

Bildquelle:
https://docs.ultralytics.com/tasks

Nächste Vorlesung und Fragen

- Trainingsstrategien (Strategy for DL Troubleshooting)
- Fragen?

Prof. Dr. Daniel Gaida
Professor für Cyber-Physische Systeme
Fakultät für Informatik und Ingenieurwissenschaften - Institut für Informatik