BeamNG

Not Enough Data: Generating Artificial Datasets

Adam Ivora

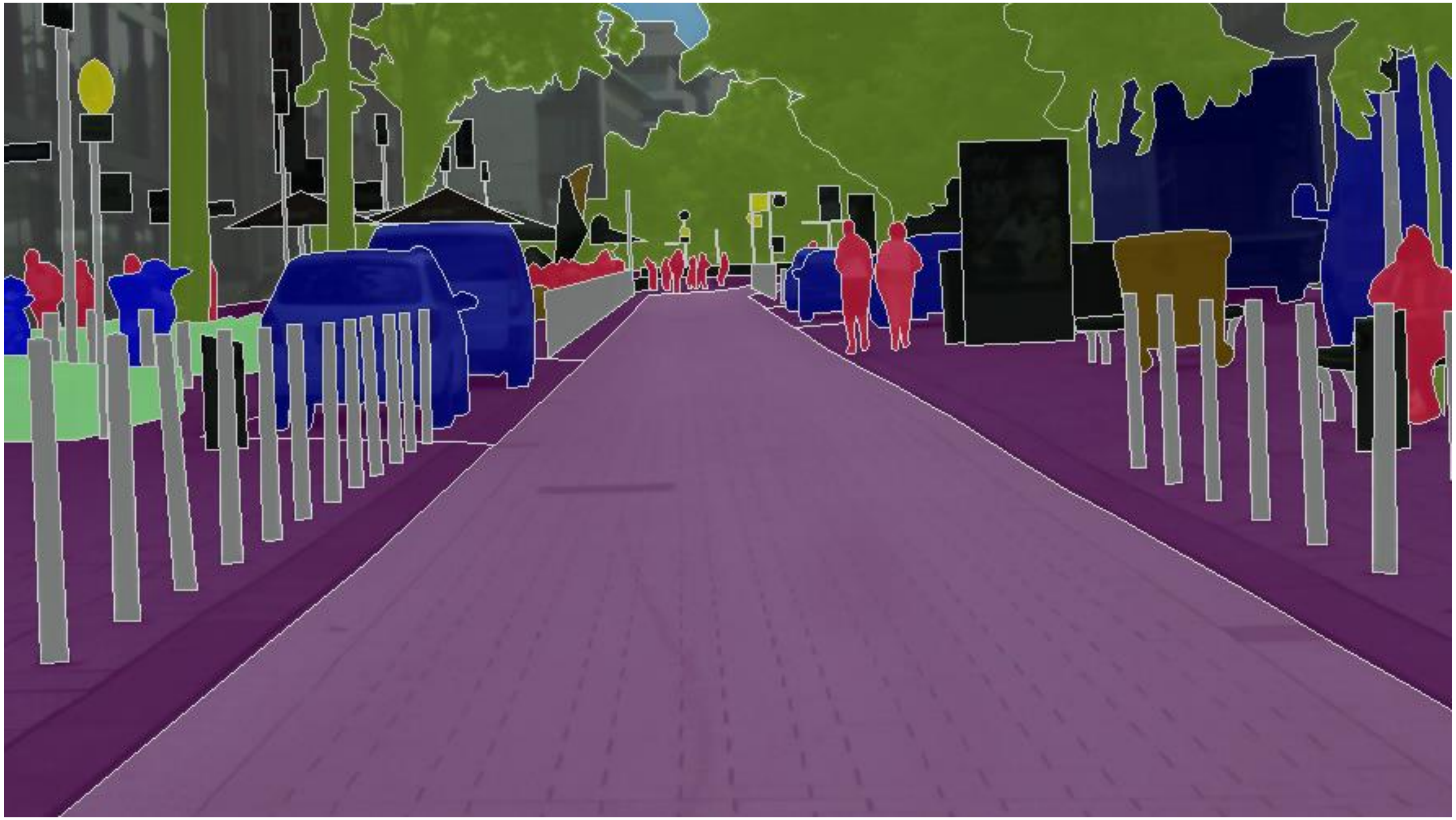
Semantic Annotation in Computer Vision

✓ Each pixel of an image is annotated.

✓ Large datasets are needed for DL models.

Semantic annotation driving datasets:

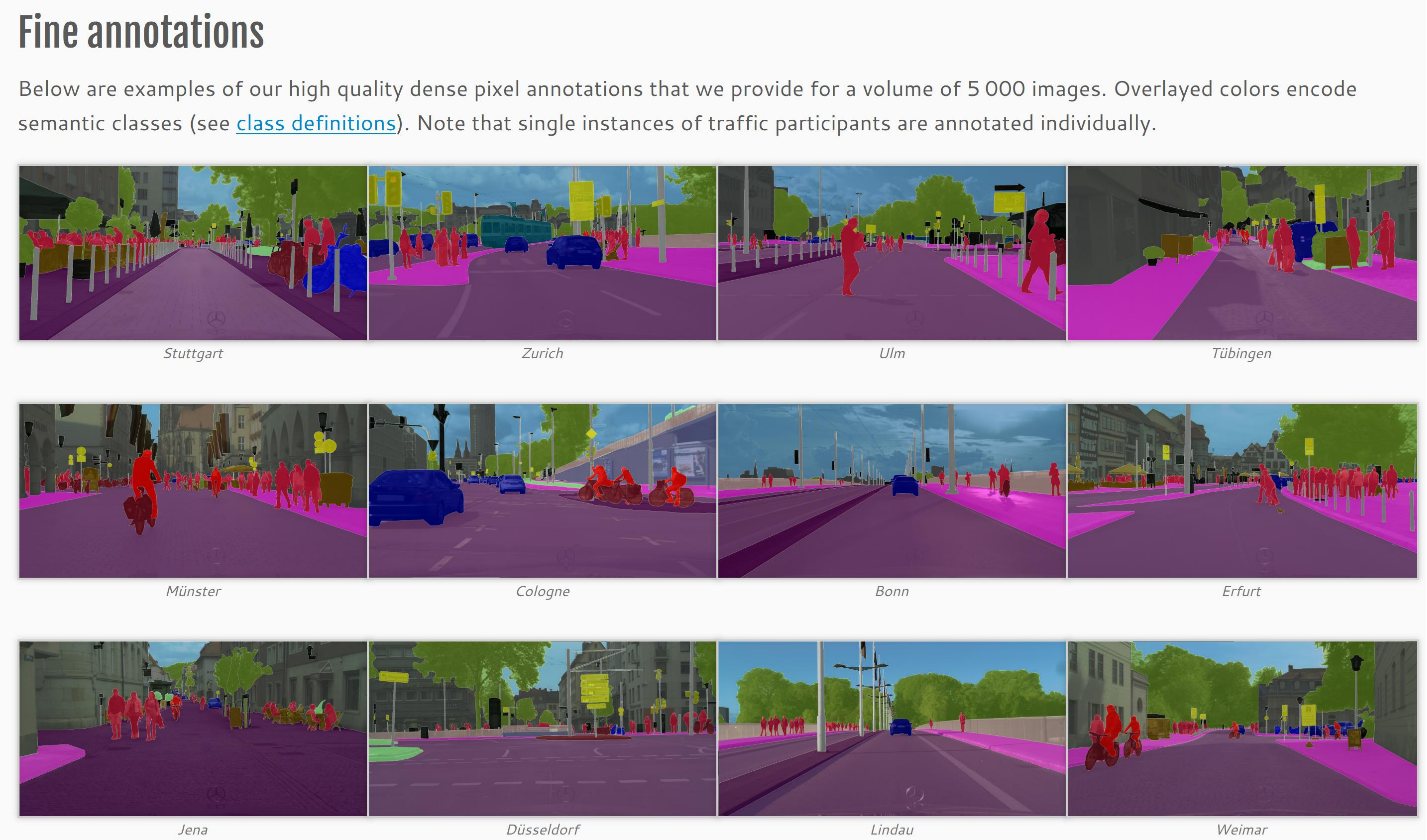
- DSEC-Semantic (10891 frames)
- Cityscapes (5000 fine + 20000 coarse)
- KITTI (400 images)
- BDD100K (10000 images)



source: cityscapes.org

Look at Cityscapes

source: cityscapes.org



Fine annotations

Below are examples of our high quality dense pixel annotations that we provide for a volume of 5 000 images. Overlayed colors encode semantic classes (see class definitions). Note that single instances of traffic participants are annotated individually.

Stuttgart

Zurich

Ulm

Tübingen

Münster

Cologne

Bonn

Erfurt

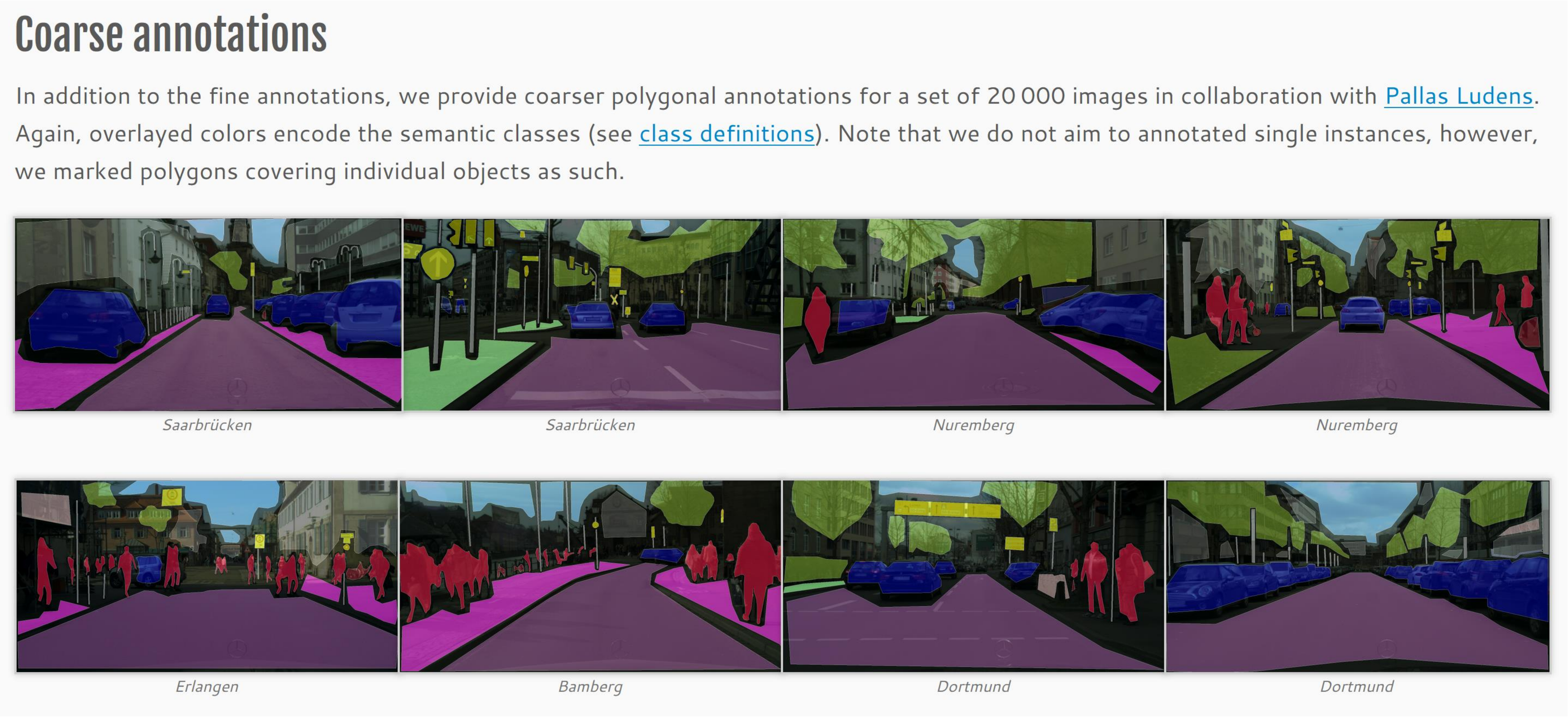
Jena

Düsseldorf

Lindau

Weimar

source: cityscapes.org



Coarse annotations

In addition to the fine annotations, we provide coarser polygonal annotations for a set of 20 000 images in collaboration with Pallas Ludens. Again, overlayed colors encode the semantic classes (see class definitions). Note that we do not aim to annotated single instances, however, we marked polygons covering individual objects as such.

Saarbrücken *Saarbrücken* *Nuremberg* *Nuremberg*

Erlangen *Bamberg* *Dortmund* *Dortmund*

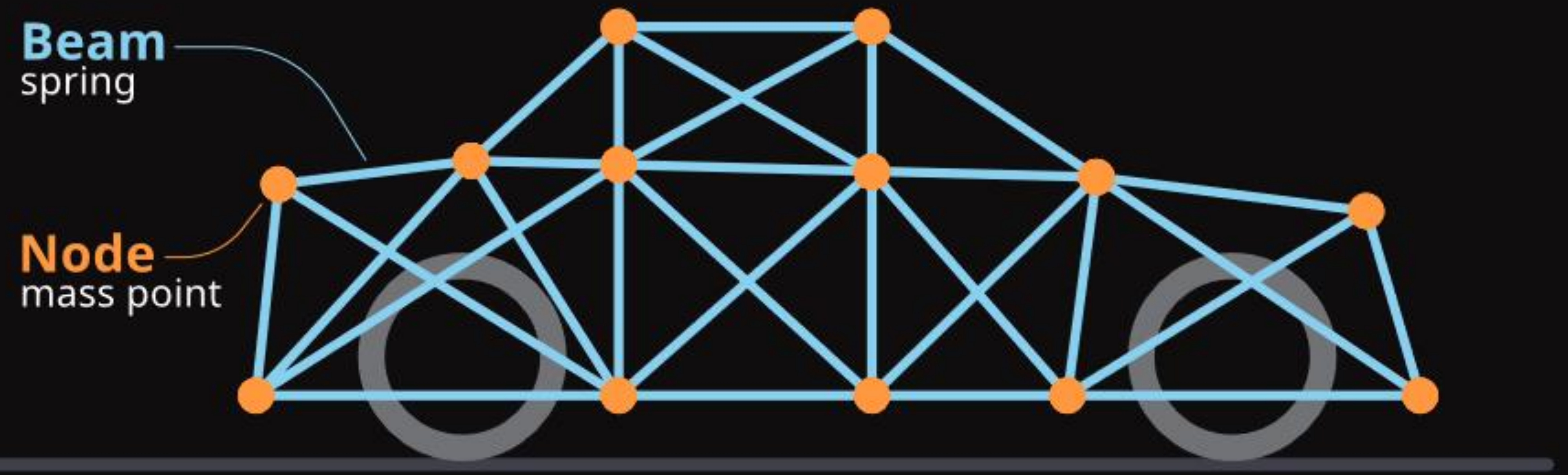
Motivation

Why even bother trying

- ✓ segmentation datasets are limited in size and quality
- ✓ synthetic datasets have unlimited size and have “perfect” annotations
- ✓ research if a synthetic dataset can be used in the field of driving semantic segmentation datasets

ABOUT US

BeamNG



BeamNG.*drive*

BeamNG.drive is a popular realistic driving simulator in the gaming sector where the finely-tuned physics model and highly developed graphics create an immersive driving experience in a sandbox environment.

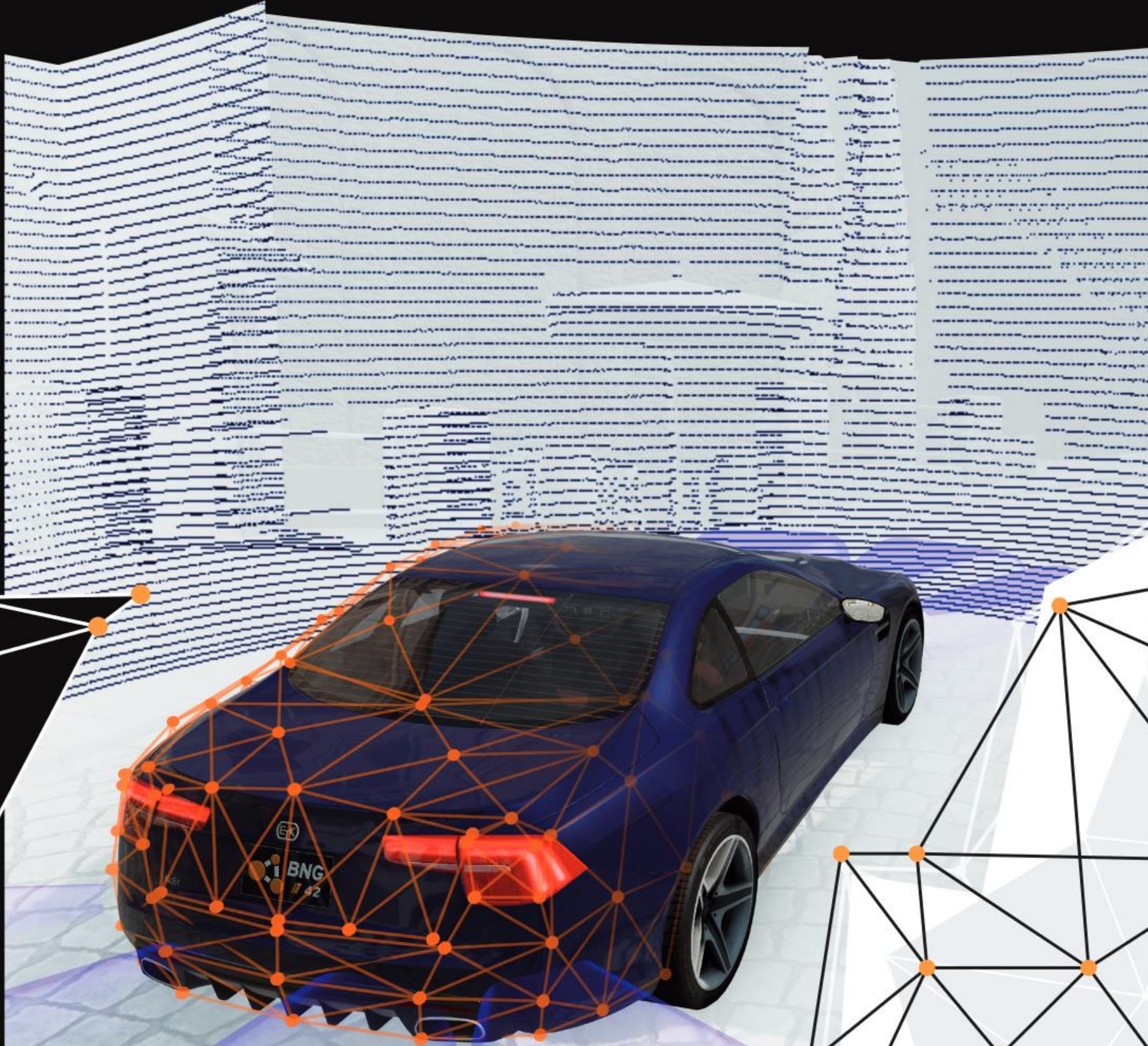
The BeamNG physics engine is at the core of the most detailed and authentic vehicle simulation you've ever seen in a game.

Every component of a vehicle is simulated in real-time using nodes (mass points) and beams (springs). Crashes feel visceral, as the game uses an incredibly accurate damage model.

BeamNG.*tech*

BeamNG.tech expands the capabilities of BeamNG.drive by supporting automated data generation, providing various sensor models that are commonly used in the autonomous driving sector, and allowing more parametrization of the overall software.

Combining this with our support for industrial and academic projects, BeamNG.tech proves to be a versatile tool for ADAS development, data generation and a strong foundation for any driver training application.

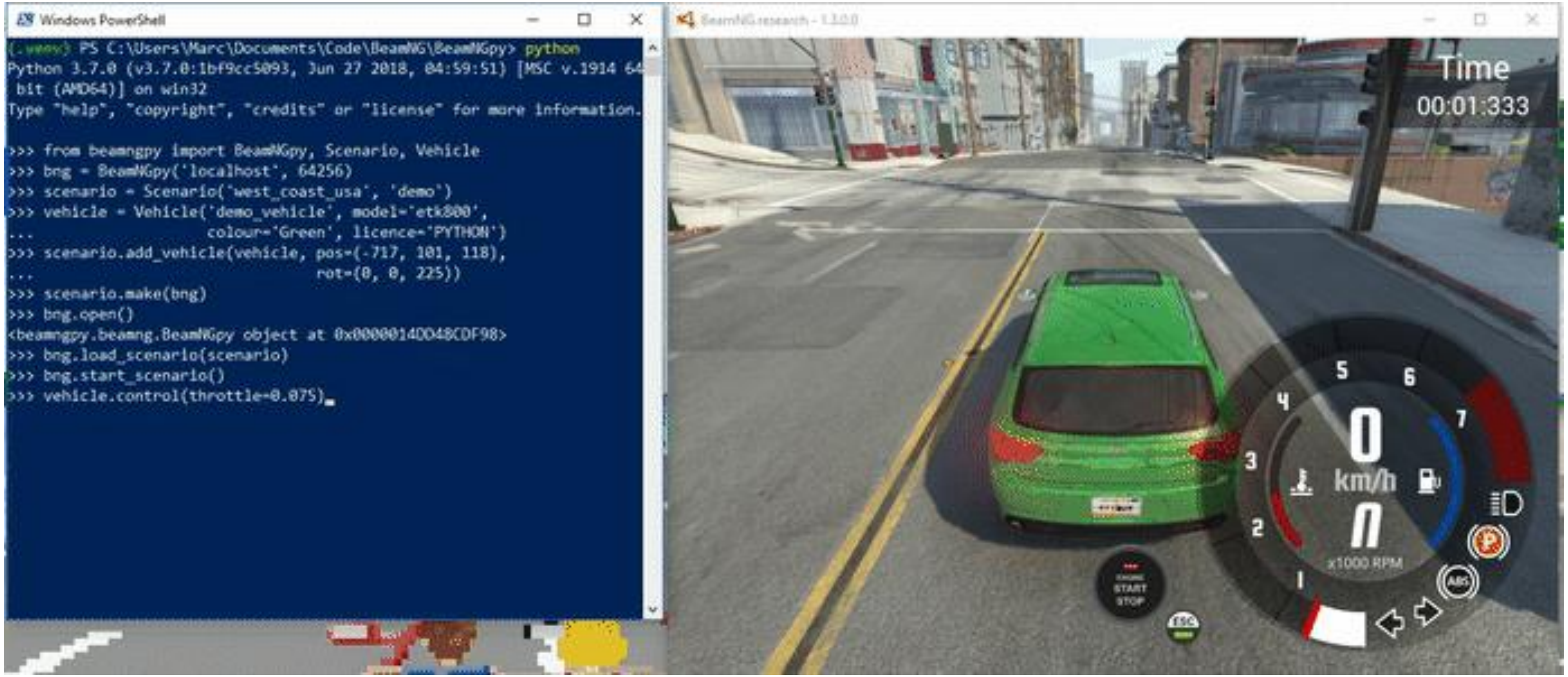


Annotations: OFF

BeamNGpy – Python API for BeamNG.tech

https://github.com/BeamNG/BeamNGpy

Simple interface exposing the functions of BeamNG.tech – vehicle/environment control, sensor readings (Camera, LiDAR, Ultrasonic, Accelerometer), annotations



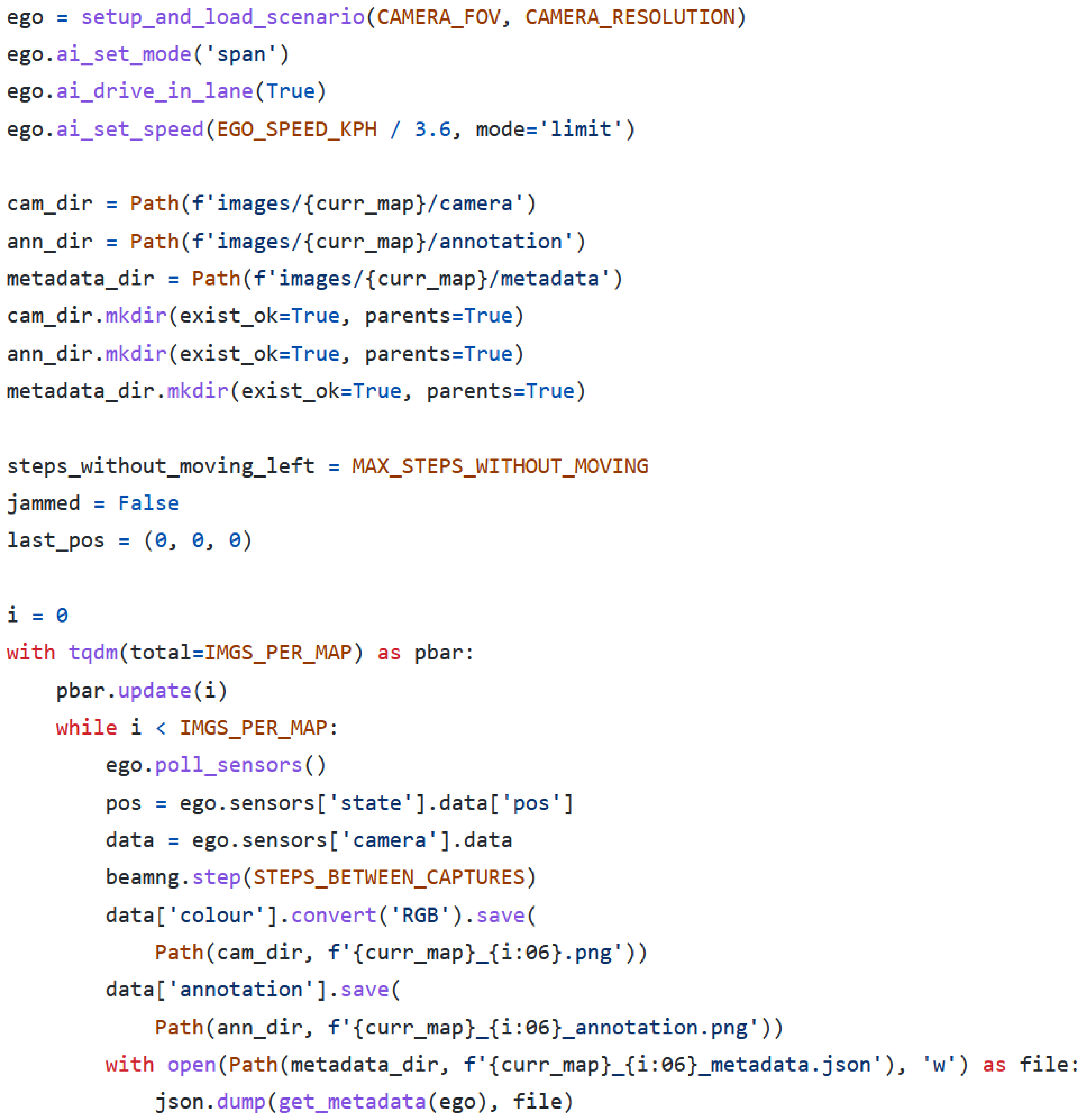
```
(.venv) PS C:\Users\Marc\Documents\Code\BeamNG\BeamNGpy> python
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC v.1914 64
 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.

>>> from beamngpy import BeamNGpy, Scenario, Vehicle
>>> bng = BeamNGpy('localhost', 64256)
>>> scenario = Scenario('west_coast_usa', 'demo')
>>> vehicle = Vehicle('demo_vehicle', model='etk800',
...                   colour='Green', licence='PYTHON')
>>> scenario.add_vehicle(vehicle, pos=(-717, 101, 118),
...                               rot=(0, 0, 225))
>>> scenario.make(bng)
>>> bng.open()
<beamngpy.beamng.BeamNGpy object at 0x0000014DD48CDF98>
>>> bng.load_scenario(scenario)
>>> bng.start_scenario()
>>> vehicle.control(throttle=0.075)
```

Generation Code

https://github.com/aivora-beamng/bngpy_generate_annotated_dataset

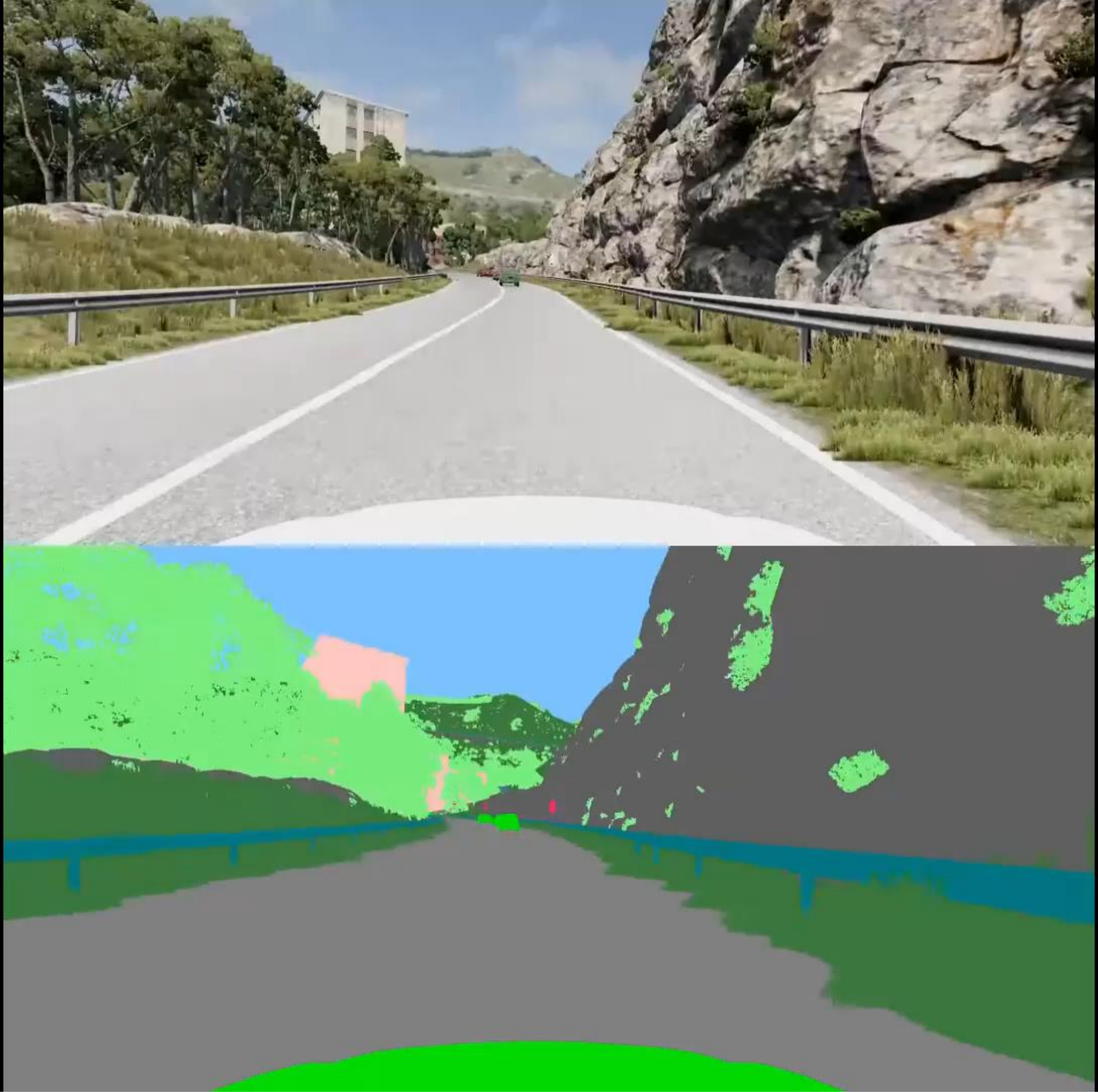
under 200 lines of BeamNGpy code:

- spawn a vehicle with the Camera sensor
- load the map
- spawn traffic
- capture data from the Camera every 1.8 seconds, save it
- some extra code (traffic jams happen ¯_(ツ)_/¯)



```
ego = setup_and_load_scenario(CAMERA_FOV, CAMERA_RESOLUTION)
ego.ai_set_mode('span')
ego.ai_drive_in_lane(True)
ego.ai_set_speed(EGO_SPEED_KPH / 3.6, mode='limit')

cam_dir = Path(f'images/{curr_map}/camera')
ann_dir = Path(f'images/{curr_map}/annotation')
metadata_dir = Path(f'images/{curr_map}/metadata')
cam_dir.mkdir(exist_ok=True, parents=True)
ann_dir.mkdir(exist_ok=True, parents=True)
metadata_dir.mkdir(exist_ok=True, parents=True)

steps_without_moving_left = MAX_STEPS_WITHOUT_MOVING
jammed = False
last_pos = (0, 0, 0)

i = 0
with tqdm(total=IMGS_PER_MAP) as pbar:
    pbar.update(i)
    while i < IMGS_PER_MAP:
        ego.poll_sensors()
        pos = ego.sensors['state'].data['pos']
        data = ego.sensors['camera'].data
        beamng.step(STEPS_BETWEEN_CAPTURES)
        data['colour'].convert('RGB').save(
            Path(cam_dir, f'{curr_map}_{i:06}.png'))
        data['annotation'].save(
            Path(ann_dir, f'{curr_map}_{i:06}_annotation.png'))
        with open(Path(metadata_dir, f'{curr_map}_{i:06}_metadata.json'), 'w') as file:
            json.dump(get_metadata(ego), file)
```

Cool timelapse



The Dataset In Numbers

49758 images

~150 GB

2048x1024

Resolution of the camera images

18 classes (13 from Cityscapes)

SKY, NATURE, BUILDINGS, POLE, OBSTACLES, TRAFFIC_SIGNALS, CAR, SIDEWALK, ASPHALT, GRASS, ROCK, GUARD_RAIL, WATER, MUD, SAND, COBBLESTONE

Additional metadata

Such as vehicle position, rotation, throttle, and others.

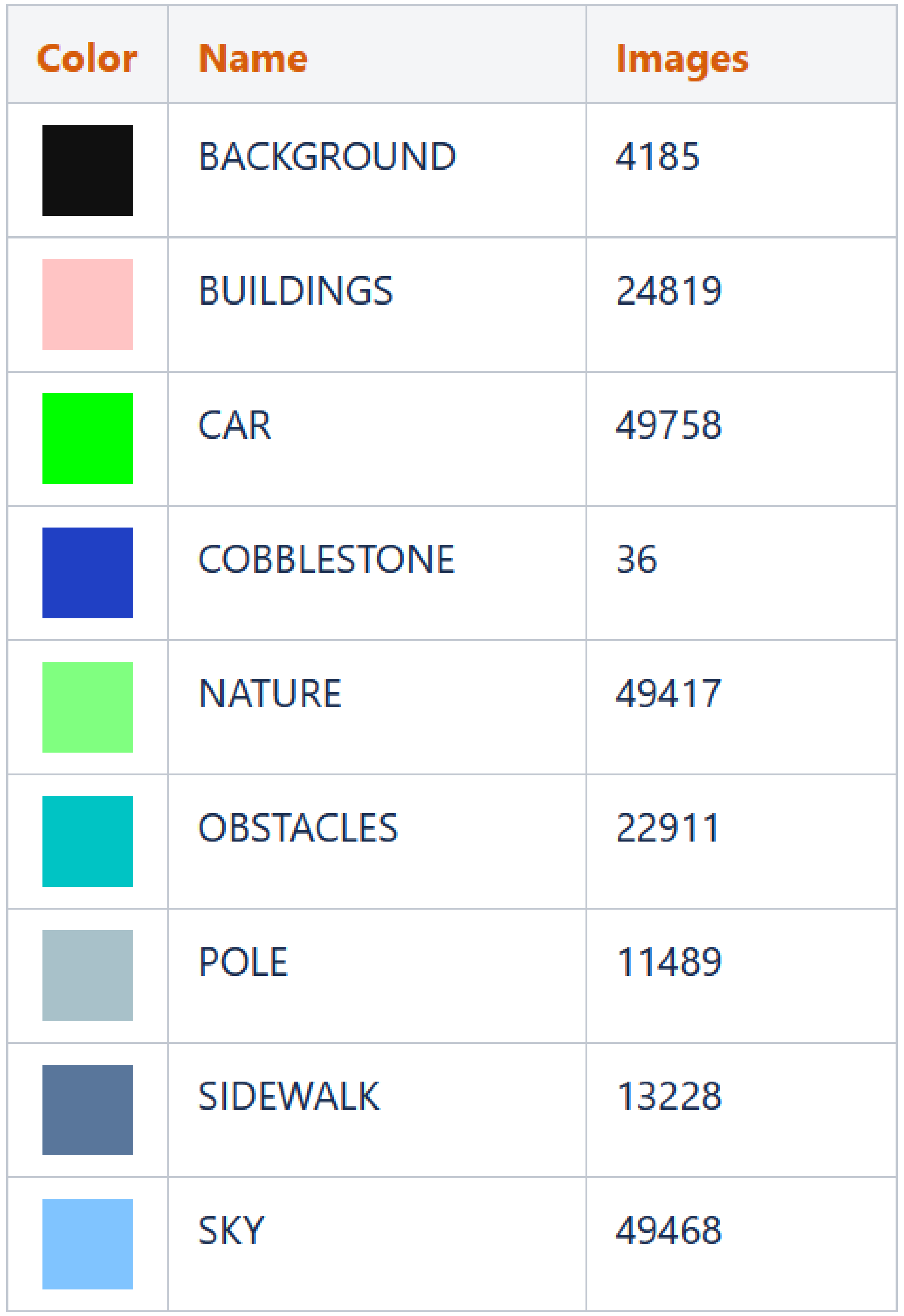
2 maps

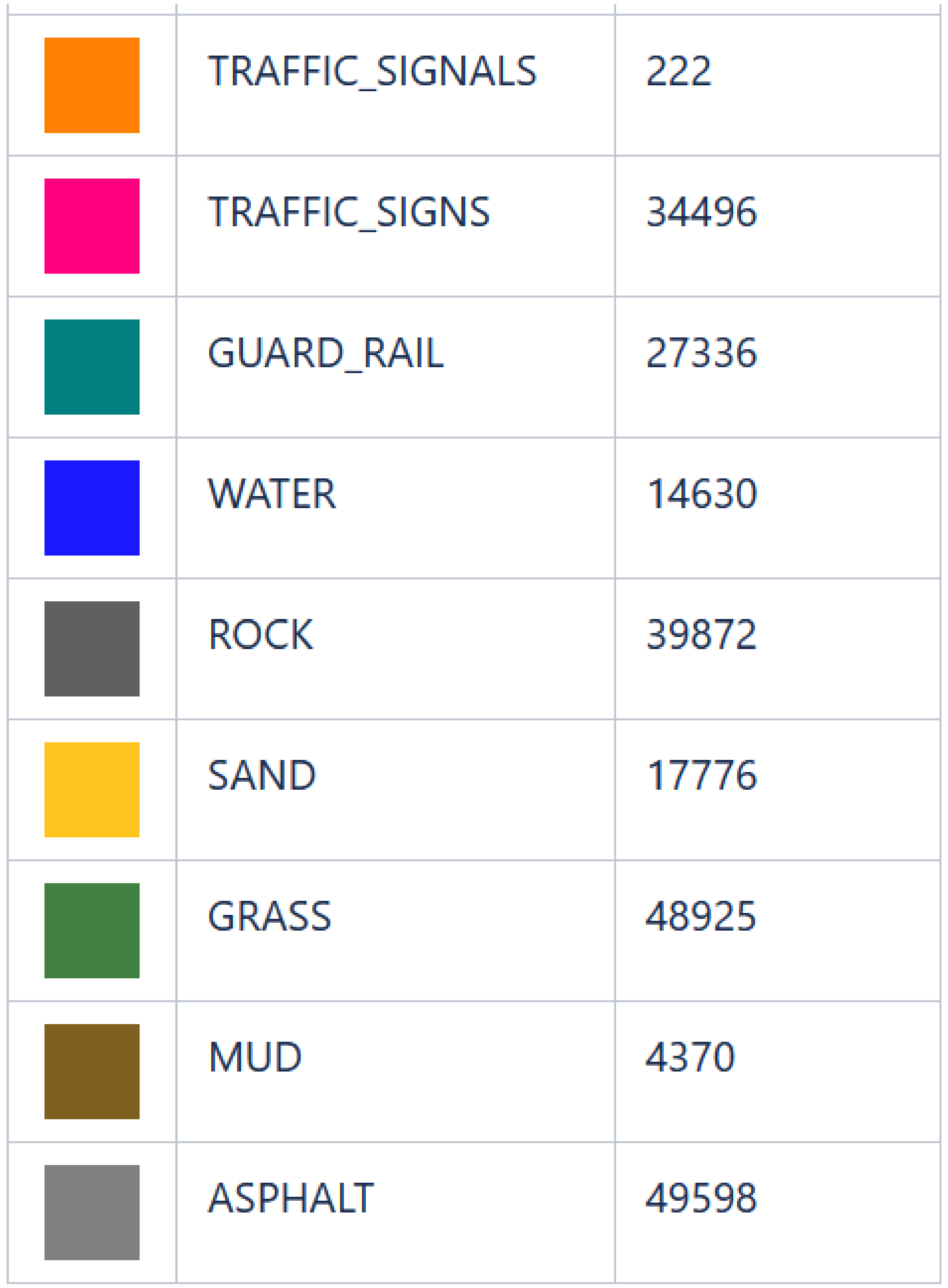
Italy and East Coast USA

>25 hours

of driving data

Classes





Color	Name	Images
⬛	BACKGROUND	4185
🟥	BUILDINGS	24819
🟩	CAR	49758
🟦	COBBLESTONE	36
🟩	NATURE	49417
🟦	OBSTACLES	22911
⬜	POLE	11489
🟦	SIDEWALK	13228
🟦	SKY	49468
🟧	TRAFFIC_SIGNALS	222
🟪	TRAFFIC_SIGNS	34496
🟦	GUARD_RAIL	27336
🟦	WATER	14630
⬛	ROCK	39872
🟨	SAND	17776
🟩	GRASS	48925
🟫	MUD	4370
⬜	ASPHALT	49598

Experiment

Try proving it’s not useless

✓ train on the generated dataset, test on the generated dataset – sanity check

✓ pretrain on the generated dataset, then on Cityscapes, test on Cityscapes test

✓ train on the generated dataset, test on Cityscapes

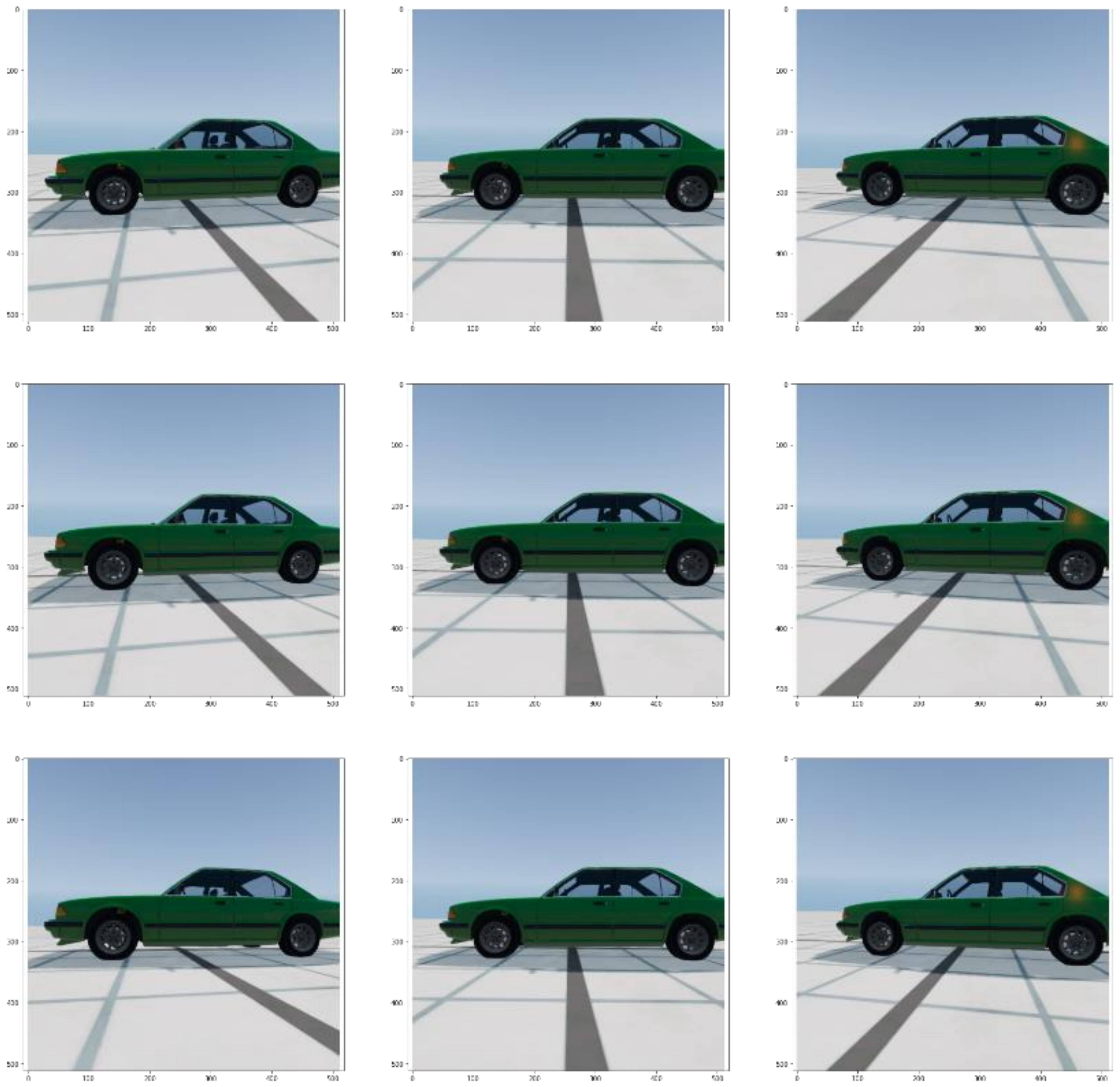
So far 1/3 done!

Comparison, Limitations

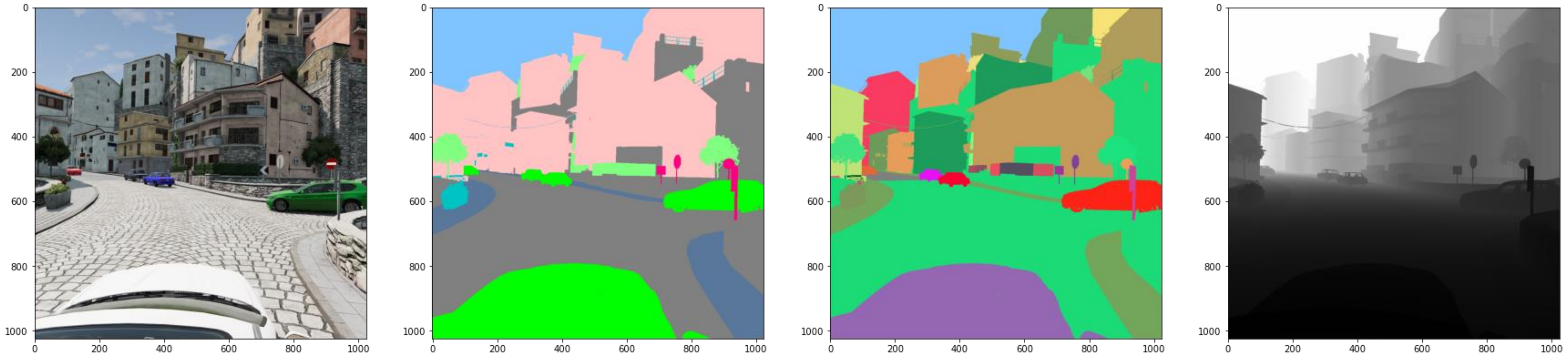
Human-annotated (Cityscapes) vs. Generated Dataset

Human-annotated Dataset	Generated Dataset
Limited size	Practically unlimited
“Random” human-factor labeling errors	Systematic labeling errors
Real life	Artificial environment
Costly and time-intensive to produce	Cheap, fast to produce (you can do it yourself)
High variability	Lower variability

Extensions



multiple cameras, different angles



instance/depth camera data

on-the-fly images generation

actually release the dataset

Resources

https://github.com/BeamNG/BeamNGpy
https://github.com/aivora-beamng/bngpy_generate_annotated_dataset
https://register.beamng.tech/ - free non-commercial academic license
https://www.youtube.com/c/beamng
https://youtu.be/YrHa402fQ0s - full 'Cool timelapse' video
this presentation – send me an e-mail

Get in Touch With Us

aivora@beamng.gmbh
tech@beamng.gmbh

https://beamng.tech/ - the dataset will be released here!
(but you can generate your own now anyways)
https://beamng.gmbh/career/

THANK YOU

Questions?

BeamNG