Master in
Computer Vision
Barcelona

M5 – Visual Recognition

Project: Cross-modal Retrieval

Week 2: Introduction to Object Detection and Segmentation

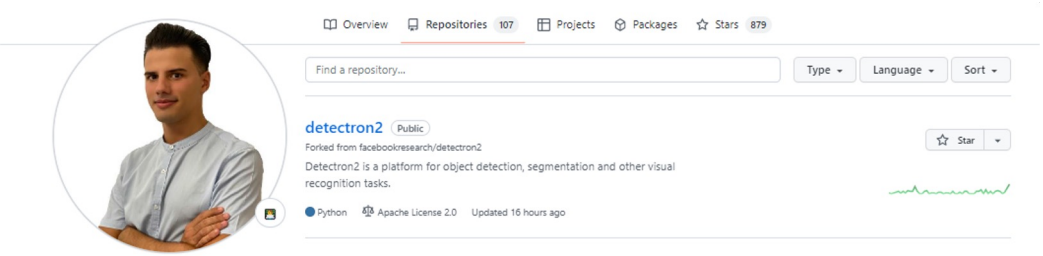
Group 3: Guillem Capellera, Abel García, Johnny Nuñez and Anna Oliveras

Tasks

Week 1: Introduction to Pytorch

A. Get familiar with Detectron2 framework.
B. Set up project.
C. Run inference with pre-trained Faster R CNN (detection) and Mask R CNN (detection and on KITTI MOTS dataset.
D. Evaluate pre-trained Faster R CNN (detection) and Mask R CNN (detection and segmentation) on KITTI MOTS dataset.
E. Fine tune Faster R CNN and Mask R CNN on KITTI MOTS
F. Start writing paper

Task A: Get familiar with Detectron2 framework



- CircleCI & GitHub Actions with generations of wheels automatically
- Compatibility with last GPU's as Ampere and Ada Lovelace (CUDA 11.8)
- Compatibility with last PyTorch including 2.0 and Python 3.11

```
import torch, detectron2
!nvcc --version
TORCH_VERSION = ".".join(torch.__version__.split(".")[:2])
CUDA_VERSION = torch.__version__.split("+")[-1]
print("torch: ", TORCH_VERSION, "; cuda: ", CUDA_VERSION)
print("detectron2:", detectron2.__version__)
```

✓ 1.8s

```
nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2023 NVIDIA Corporation
Built on Tue_Feb__7_19:32:13_PST_2023
Cuda compilation tools, release 12.1, V12.1.66
Build cuda_12.1.r12.1/compiler.32415258_0
torch:  2.0 ; cuda:  cu118
detectron2: 0.7
```

Task B: Setup project

Dataset → KITTI-MOTS [1]

- Training data
  - 12 sequences
  - 8073 pedestrian masks + 18831 car masks
- Validation data
  - 9 sequences
  - 3347 pedestrian masks + 8068 car masks

Backbone:

- Faster R-CNN
  - Faster_rcnn_X_101_32x8d_FPN_3x
- Mask R-CNN
  - Mask_rcnn_X_101_32x8d_FPN_3x

Read and learn to deal with the annotations
→ pycocotools [3] → mask_utils

[2]

```
time_frame id class_id img_height img_width rle
```

An example line from a txt file:

```
52 1005 1 375 1242 WSV:2d;1O10000O10000O1O100O100O1O100O10000000000000000O100O102N5K00O1O1N2O110002O001O1NTga3
```

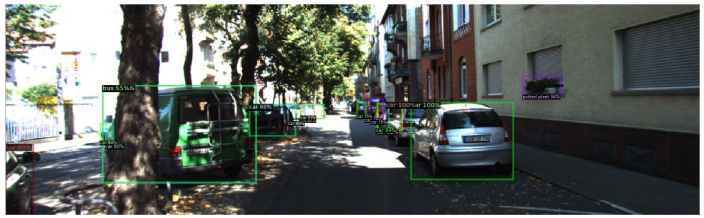
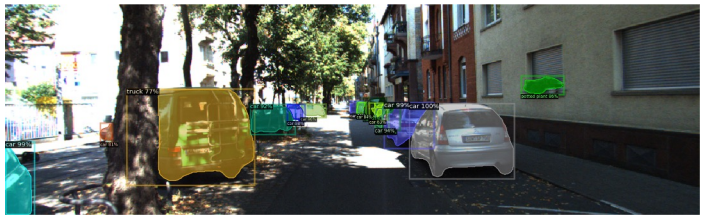
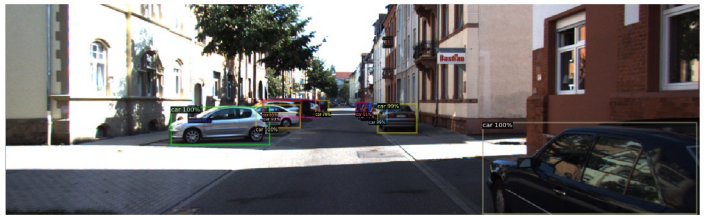
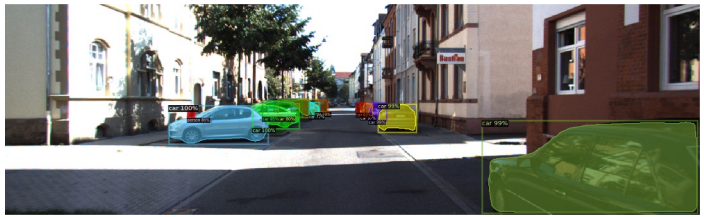
Which means

time frame 52
object id 1005 (meaning class id is 1, i.e. car and instance id is 5)
class id 1
image height 375
image width 1242
rle WSV:2d;1O10000O10000O1O100O100O1O100O10000000000000000O100O...1O1N

image height, image width, and rle can be used together to decode a mask using cocotools.

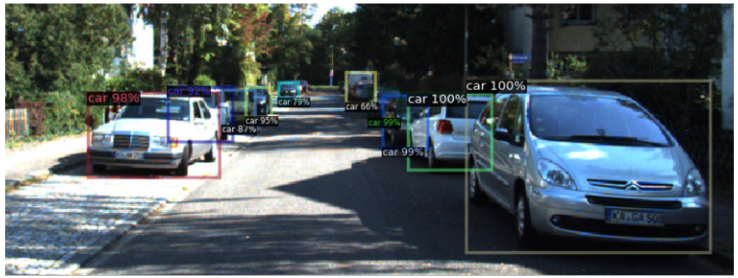
[1] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *Advances in Neural Information Processing Systems (NIPS)*, 91-99. Retrieved from https://arxiv.org/pdf/1506.01497.pdf
[2] https://www.vision.rwth-aachen.de/page/mots
[3] https://github.com/ppwwyyxx/cocoapi

Task C: Run inference with pre-trained Faster R CNN (detection) and Mask R CNN (detection and segmentation)

Image ID (seq. 0)	Faster-RCNN	Mask-RCNN
0		
100		
200		

Task C: Run inference with pre-trained Faster R CNN (detection) and Mask R CNN (detection and segmentation)

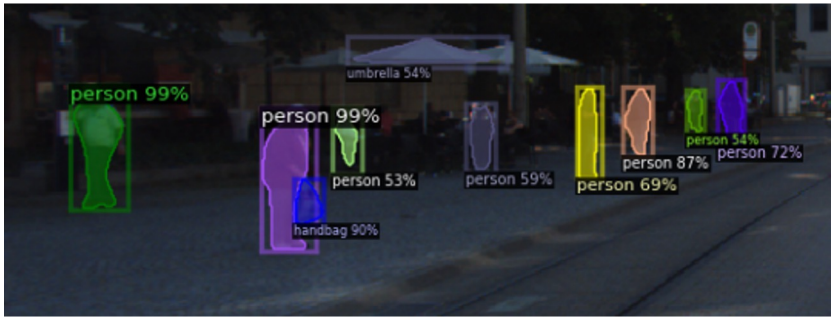
Good examples: Faster R CNN



- Faster R-CNN does a great job detecting cars and persons!
- It is able to detect the objects even when there are occlusions or the objects are far away from the camera

Task C: Run inference with pre-trained Faster R CNN (detection) and Mask R CNN (detection and segmentation)

Good examples: Mask R CNN

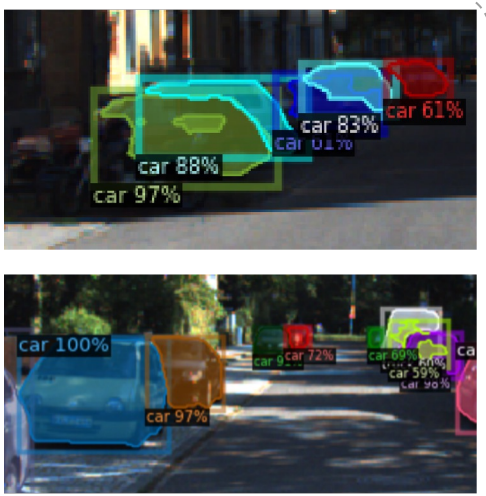


- Segmentation mask and bounding boxes are quite accurate in the majority of the cases.
- People can be even detected when they are inside the car!

Task C: Run inference with pre-trained Faster R CNN (detection) and Mask R CNN (detection and segmentation)

Bad examples

1



2

1. Cars parked too close are difficult to segment correctly by the Mask-RCNN due to the occlusions.
2. When there is more than one person close together and with occlusions, it is difficult to determine the different instances and masks. Mask R CNN seems to do better, although some masks could be better like the blue one.
3. Some examples may be difficult to segment due to light conditions or shadows → may lead to misclassifications / broken masks
4. Trucks are sometimes misclassified as cars

3

4

Task D: Metrics glossary

Average Precision (AP): This measures the average precision of the model across multiple levels of IoU overlap with ground truth boxes. In the context of object detection, AP is computed for each object class and averaged across all classes to obtain the mean average precision (mAP) for the model.

- **AP_{50}**: This measures the model's performance when it detects objects with at least 50% IoU overlap with ground truth boxes.
- **AP_{75}**: This measures the model's performance when it detects objects with at least 75% IoU overlap with ground truth boxes.
- **AP_{small}**: This measures the model's performance when detecting small objects (defined as having area less than 32^2 pixels).
- **AP_{medium}**: This measures the model's performance when detecting medium-sized objects (defined as having area between 32^2 and 96^2 pixels).
- **AP_{large}**: This measures the model's performance when detecting large objects (defined as having area greater than 96^2 pixels).

Average Recall (AR): This measures the fraction of ground truth objects that are successfully detected by the model at different levels of IoU overlap. In the context of object detection, AR is computed for each object class and averaged across all classes.

- **AR_{1}**: This measures the model's recall when only allowing one detection per image.
- **AR_{10}**: This measures the model's recall when allowing up to ten detections per image.
- **AR_{100}**: This measures the model's recall when allowing up to 100 detections per image.
- **AR_{small}**: This measures the model's recall when detecting small objects.
- **AR_{medium}**: This measures the model's recall when detecting medium-sized objects.
- **AR_{large}**: This measures the model's recall when detecting large objects.

Task D: Evaluate pre-trained **Faster R-CNN** (detection)

AP → Average Precision
AR → Average Recall

- An AP of 79.4% at IoU=0.5 indicates that the model had a relatively **high precision** when there was some overlap between the predicted and ground-truth bounding boxes.
- An AP of 64.8% at IoU=0.75 means that the model had a relatively **high precision** when the overlap was more strict.
- An AR of 68.4% means that, on average, the model was able to recall 68.4% of the objects in the test dataset. An AR of 92% for large objects indicates that the model was very good at detecting **large objects** like cars. However, the AR for small objects (53.6%) was relatively low, suggesting that the model struggled to detect small objects like pedestrians.
- The model achieved a higher AP for cars (69.6%) than for pedestrians (44.9%), indicating that it was better at detecting cars than pedestrians.

Detection metrics			
AP	57.3	AR	19.8
AP_{50}	**79.4**	AR_{50}	67.3
AP_{75}	64.8	AR_{75}	68.4
AP_{small}	40.3	AR_{small}	53.6
AP_{medium}	67.5	AR_{medium}	77.6
AP_{large}	68.0	AR_{large}	**92.0**

Category	bbox AP
Pedestrian	44.9
Car	69.6

Task D: Evaluate pre-trained **Mask R-CNN** (detection and segmentation)

AP → Average Precision
AR → Average Recall

Detection

- Mask R-CNN achieves an AP of 58.9%, while the Faster R-CNN achieves an AP of 57.3%. This indicates that pre-trained Mask R-CNN is better at detecting objects in the images.
- Similarly, in terms of AR values, Mask R-CNN outperforms Faster R-CNN. The AR values for Mask R-CNN are higher than those of Faster R-CNN in all categories, indicating that Mask R-CNN is better at detecting objects at different scales.

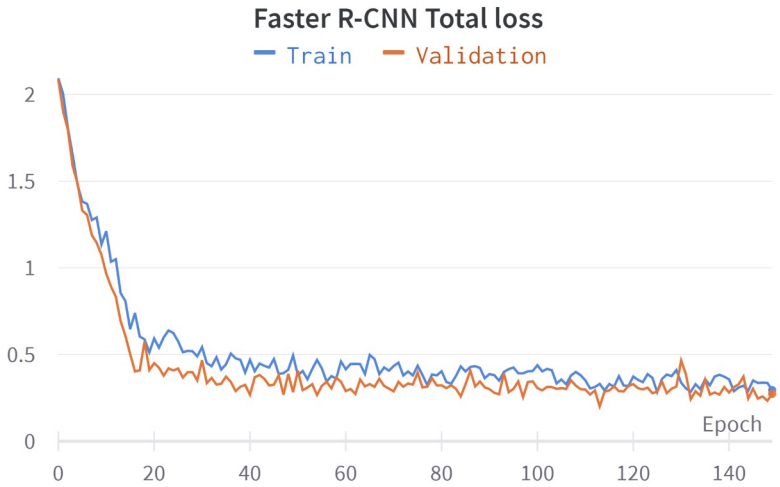
Segmentations

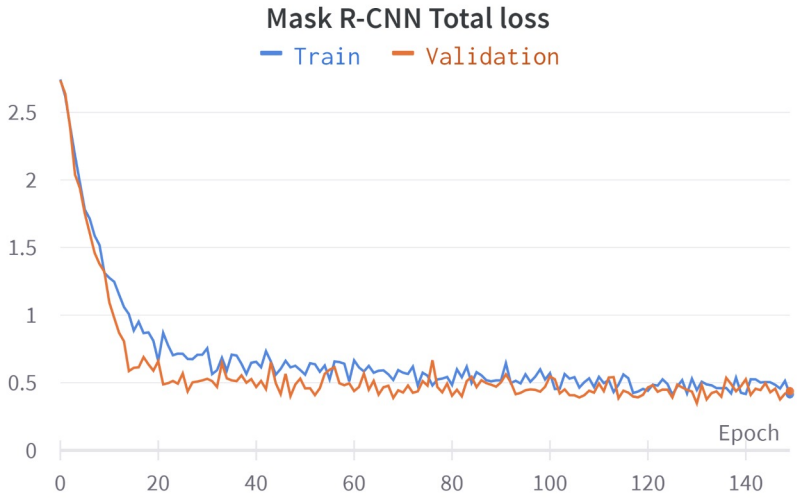
- The segmentation model has achieved an average precision of 54.4%.
- The average precision at IoU=0.50 is higher at 79.6%, indicating that the model is better at detecting objects that have a larger overlap with the ground truth segmentation.
- The model performs better on larger objects, with an AP of 75.2% for large objects compared to 33.4% for small objects. The per-category segm AP shows that the model has performed well in detecting and segmenting persons and cars, with APs of 38.1% and 70.6%, respectively.
- The results indicate that the segmentation model has performed reasonably well, but there is room for improvement, especially for small objects and some object categories.

Detection metrics				Segmentation metrics			
AP	58.9	AR	20.3	AP	54.4	AR	18.6
AP_{50}	**81.1**	AR_{50}	68.5	AP_{50}	79.6	AR_{50}	63.8
AP_{75}	66.6	AR_{75}	69.8	AP_{75}	60.3	AR_{75}	64.8
AP_s	41.2	AR_s	55.4	AP_s	33.4	AR_s	49.7
AP_m	69.6	AR_m	78.7	AP_m	65.3	AR_m	74.2
AP_L	67.8	AR_L	**92.4**	AP_L	75.2	AR_L	**85.7**

Category	bbox AP	segmentation AP
Pedestrian	46.8	**38.1**
Car	71.0	70.6

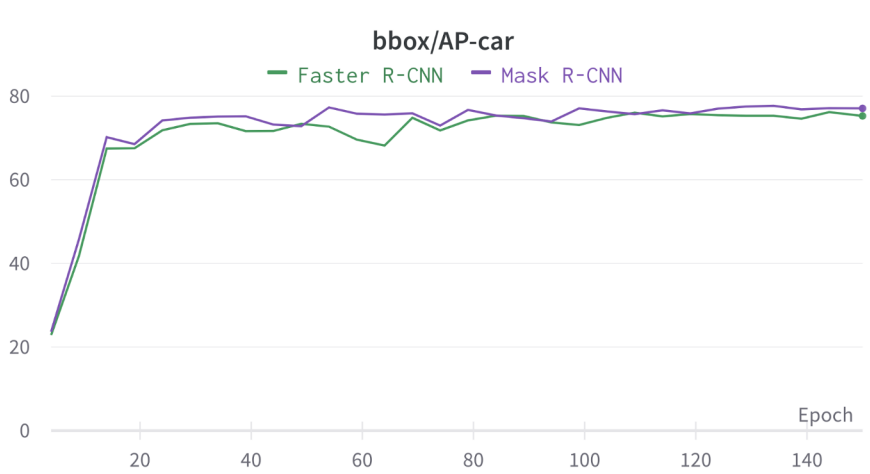
Task E: Fine tune Faster R CNN and Mask R CNN (Loss)

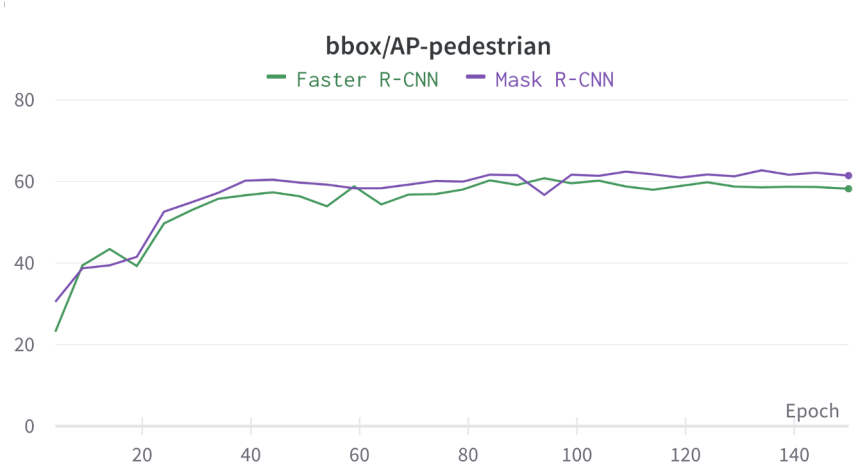




- Good behavior of the training and validation losses
- No overfitting
- Minor loss in Faster R-CNN than Mask R-CNN

Task E: Fine tune Faster R CNN and Mask R CNN (AP in detection)





- Better performance in car detection as in pre-trained models
- No overfitting
- Minor AP in Faster R-CNN than Mask R-CNN

Task E: Fine tune Faster R CNN and Mask R CNN (HyperParameters)

Model

- NUM_CLASSES: 2
- BATCH_SIZE_PER_IMAGE: 128

Dataloader

- NUM WORKERS: 4
- EVAL_PERIOD = 100

Solver

- BASE_LR: 0.001
- MAX_ITER: 3000
- STEPS: (1000, 2000, 2500)
- GAMMA: 0.5
- IMS_PER_BATCH: 2

Task E: Fine tune Faster R CNN and Mask R CNN (Quantitative results)

Before fine tuning

	Faster RCNN	Mask RCNN	
Category	bbox AP	bbox AP	seg AP
Pedestrian	44.9	46.8	**38.1**
Car	69.6	71.0	70.6

Faster RCNN				Mask RCNN							
Detection metrics				Detection metrics				Segmentation metrics			
AP	57.3	AR	19.8	AP	58.9	AR	20.3	AP	54.4	AR	18.6
AP_{50}	**79.4**	AR_{50}	67.3	AP_{50}	81.1	AR_{50}	68.5	AP_{50}	79.6	AR_{50}	63.8
AP_{75}	64.8	AR_{75}	68.4	AP_{75}	66.6	AR_{75}	69.8	AP_{75}	60.3	AR_{75}	64.8
AP_s	40.3	AR_s	53.6	AP_s	41.2	AR_s	55.4	AP_s	33.4	AR_s	49.7
AP_m	67.5	AR_m	77.6	AP_m	69	AR_m	78	AP_m	65.3	AR_m	74.2

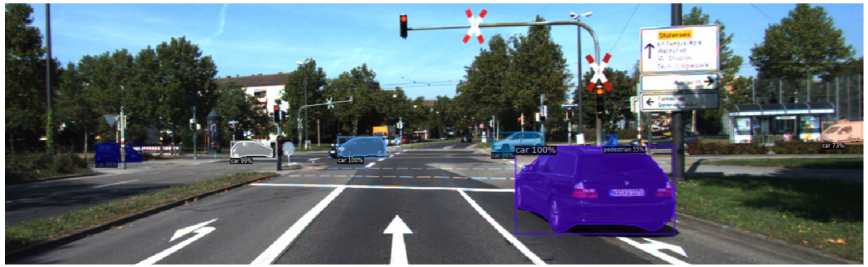
After fine tuning

	Faster RCNN	Mask RCNN	
Category	bbox AP	bbox AP	seg AP
Pedestrian	57.2	57.3	44.0
Car	72.4	74.0	72.3

Faster RCNN				Mask RCNN							
Detection metrics				Detection metrics				Segmentation metrics			
AP	64.8	AR	21.2	AP	65.6	AR	21.5	AP	58.2	AR	19.4
AP_{50}	**87.5**	AR_{50}	70.5	AP_{50}	**88.0**	AR_{50}	72.2	AP_{50}	84.6	AR_{50}	65.1
AP_{75}	75.0	AR_{75}	72.4	AP_{75}	75.8	AR_{75}	74.2	AP_{75}	65.7	AR_{75}	66.3
AP_s	48.3	AR_s	59.7	AP_s	49.1	AR_s	61.9	AP_s	38.4	AR_s	51.9
AP_m	74.2	AR_m	79.9	AP_m	75.0	AR_m	81.5	AP_m	70.2	AR_m	75.1

- **Improve** in all the results!
- A lot of improvement for the **pedestrian's** class

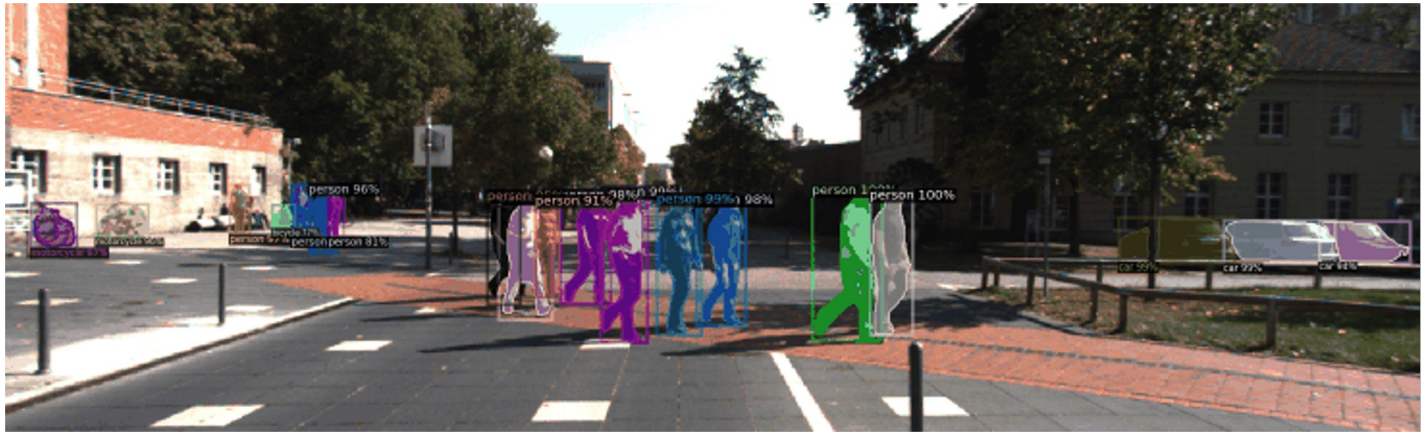
Task E: Fine tune Faster R CNN and Mask R CNN (Qualitative results)



- Much more **accurate** segmentation masks !
- Better in handling occlusions

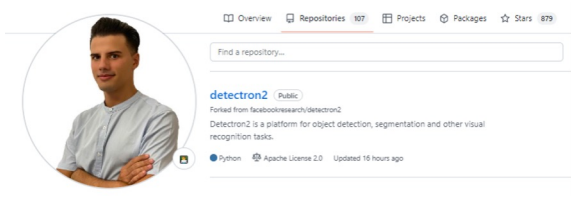
Task E: Fine tune Faster R CNN and Mask R CNN (Qualitative results)

Before fine tuning



After fine tuning

Conclusions



- **Detectron 2** → Understanding detectron2 source code creating Pull Request to update the repository to create wheels for new hardware and code.
- Mask R-CNN has better Average Precision and Recall than Faster R-CNN.
  → Better in detecting objects in the KITTI-MOTS dataset

- **Fine-tuning.**
  → Improves a lot the performance on pedestrian's class (+10 p)
  → Improves the overall results

- Understanding Mixed Precision and another techniques for training and inference.

		Faster R-CNN	Mask R-CNN	
	Category	**bbox AP**	**bbox AP**	**segmentation AP**
	Pedestrian	44.9	46.8	38.1
	Car	69.6	71.0	70.6
Fine-tuned	**Pedestrian**	57.2	57.3	**44.0**
Fine-tuned	**Car**	72.4	74.0	72.3