

Some information about the status quo before actually starting the project:

Repo:

You should have access to my project repo now.

Data:

- At the moment, I have all the data on an external hard drive at my home.
- My colleague in the project had set up a rather complicated folder structure for the data - see [Documentation/setup_structure.pdf](#), unfortunately only in German (but there's a picture on page 1). This was all in vain, because we never worked on the data in the way he intended ;-), but it makes it complicated for me to search through ...
- Anyway, I think I can say that I have something like ~300 annotated photos, where the annotations are both bounding boxes and pixel masks for all the cucumbers in the photos.
- In addition, there's about 4000 unlabelled photos and only ~6 minutes of usable video. (About 3600 of these photos are also there in RAW format, that's why there's so much data [300 GB] in total, but I guess we won't need those.)
 - By the way: The photos are rather big (3000x4000 pixels) and I have no clear intuition if or how this is going to be problematic when putting them into a model.
- The annotations are provided in the COCO data format. Basically, long JSON files where the bboxes and masks are listed together with the corresponding image file names. (However, both the image files and the annotations are spread out over several different directories and I haven't yet collected this into a complete set.)

Software:

- The previously used software for creating and training the model and showing the results is in the folder [BerrysSoftware](#) (start from [generate_ai2.ipynb](#)).
- He used this pretrained model:
https://pytorch.org/vision/stable/models/generated/torchvision.models.detection.maskrcnn_resnet50_fpn.html
 - [https://pytorch.org/vision/stable/models/faster_rcnn.html?](https://pytorch.org/vision/stable/models/faster_rcnn.html?highlight=torchvision+models+detection+faster_rcnn)
`highlight=torchvision+models+detection+faster_rcnn` is also used
 - if you want to have a look at the model architecture, you can use <https://colab.research.google.com/drive/1davaeVToCI4G4PM1AMVcm9gzp8kPdS5b?usp=sharing>
 - if we want to continue using this model,
<https://github.com/NVIDIA/DeepLearningExamples/blob/master/PyTorch/Segmentation/MaskRCNN/README.md> may also be helpful.
- In [berry_utils.py](#), a [CustomCucumberDataset](#) is defined for putting together images and labels. Maybe I can use that. However, there have been somewhat intricate problems with the indexing of images and labels, so maybe we can find an easier way.

- Unlike what I remembered, YOLO has not been used for these cucumbers so far.
 - So do I want to switch? I don't know yet.
 - I vaguely remember that I had found pretrained YOLO models for object detection (bounding boxes) and segmentation (masks), but only the former ones were trained on the ImageNet data that have a "cucumber" class.
 - I would have to look up in my notes tomorrow.

Plans:

1. I have a handwritten "mind map" here which I guess I could turn into something on Miro.
;-) But not today anymore.
 - maybe it's actually a better idea if I do this *after* our next call.
2. I definitely have to do some sorting of the data, in order to allow some preliminary model training software to be programmed ... just to make sure that our data and labels are usable. I would like to start with a Colab or Kaggle notebook to do that. Also, I'll have to figure out a way to make the data available online. (At the moment, however, I don't really know yet *how many* GB there actually are that are necessary/helpful.)
3. I'll also definitely have to recap a little the computer vision stuff we've done in DSR. The Kaggle you sent me uses tensorflow, I guess I'd prefer pytorch ... I'll see.