

Cucumber Intelligence



An object counting tool for real-time
cucumber harvest monitoring

DSR Demo Day, 01/10/2024

Outline:

1) Introduction:

- a) How to pluck a pickle
- b) Computer vision recap

2) Training the model:

- a) YOLOv8
- b) Data and finetuning results

3) Application + Live demo

Cucumbers = Pickles = Gherkins



Harvesting close-up: Monitoring



CV recap (1): Classification



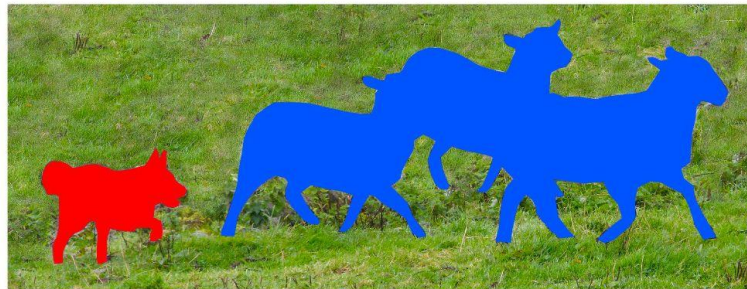
“This is an apple*.”

(* unless you're Magritte)

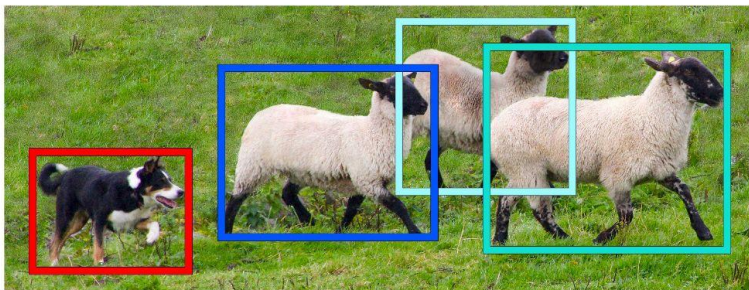
CV recap (2): Object detection, segmentation



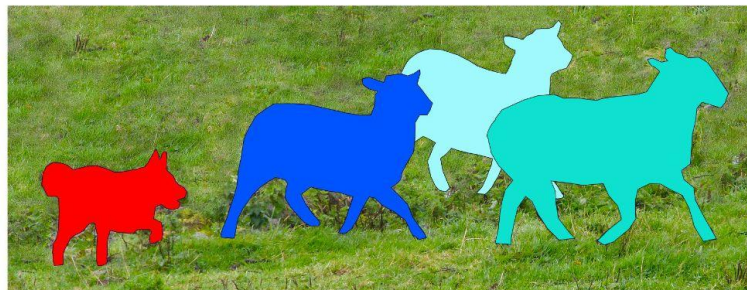
Image Recognition



Semantic Segmentation

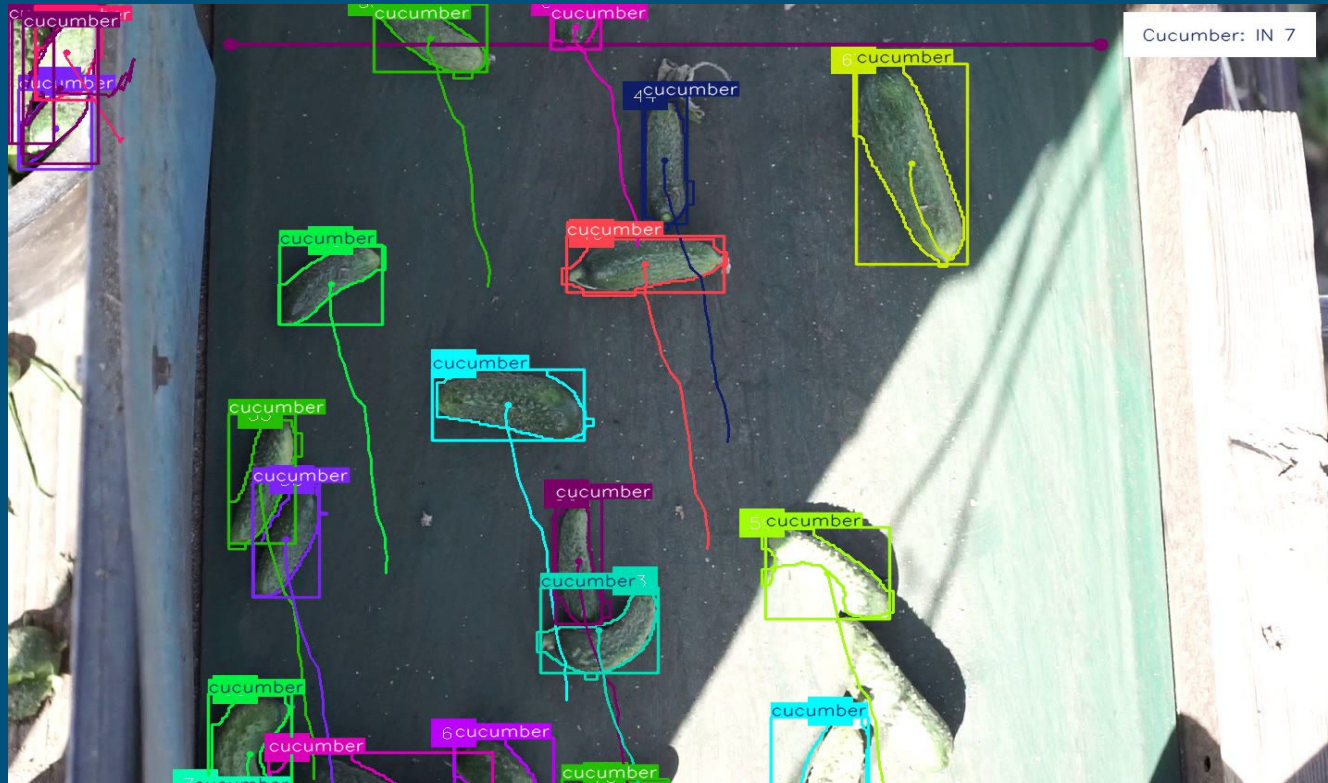


Object Detection

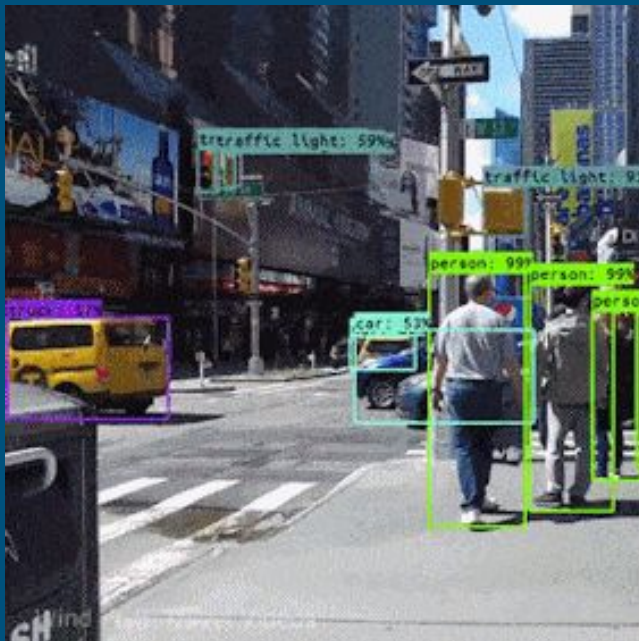


Instance Segmentation

CV recap (3): Instance tracking + counting



YOLO: “You only look once”

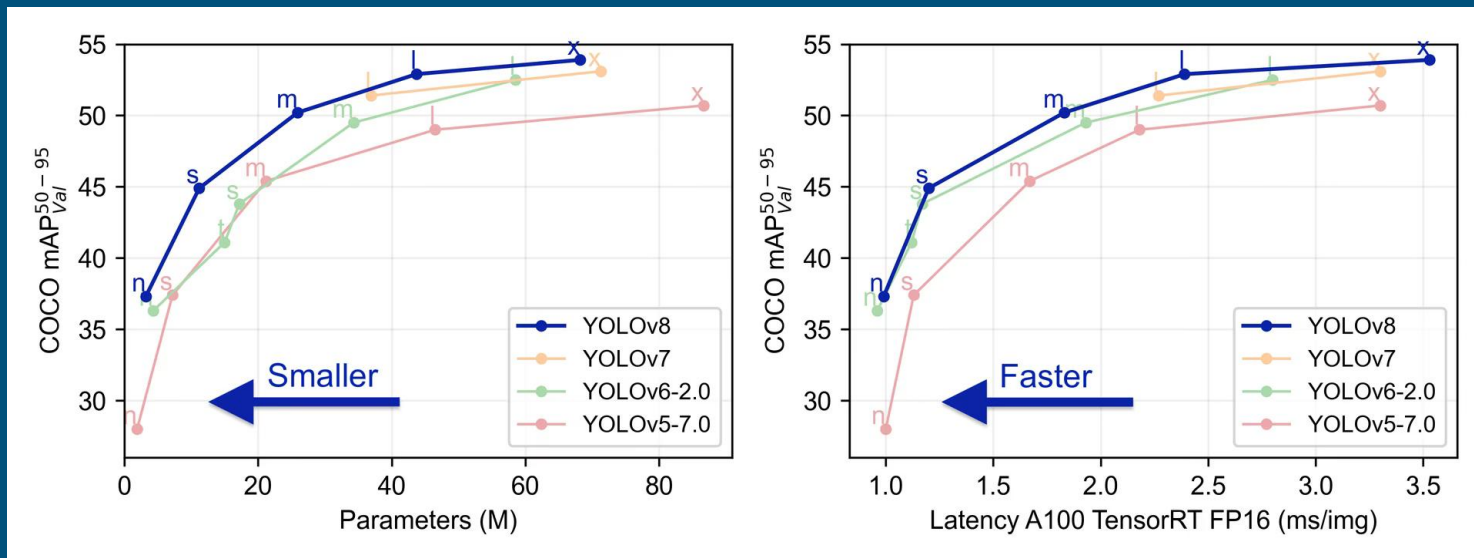


... works like a charm
(and out of the box)

... for ~80 pretrained
classes (COCO dataset).

YOLOv8 from ultralytics.com

Different pretrained models available for finetuning:



Finetuning the models



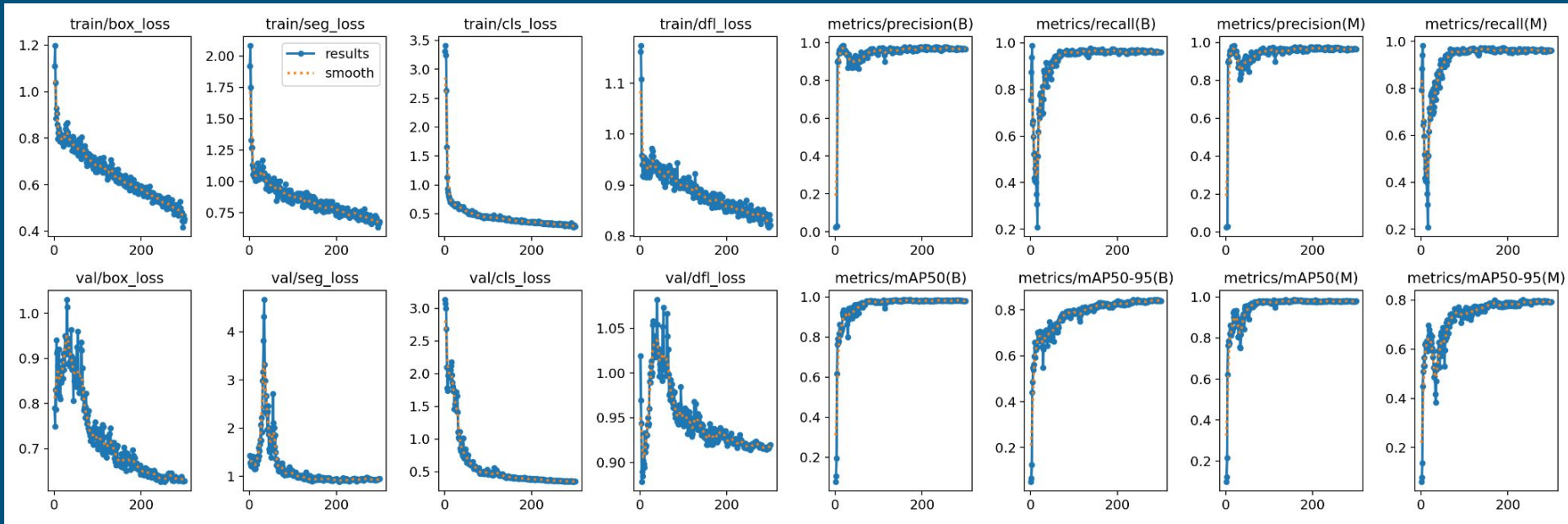
Data:

- ca. 350 photos
- hand-labelled segmentations

Two separate models:

- Detection
- Segmentation

Finetuning the model

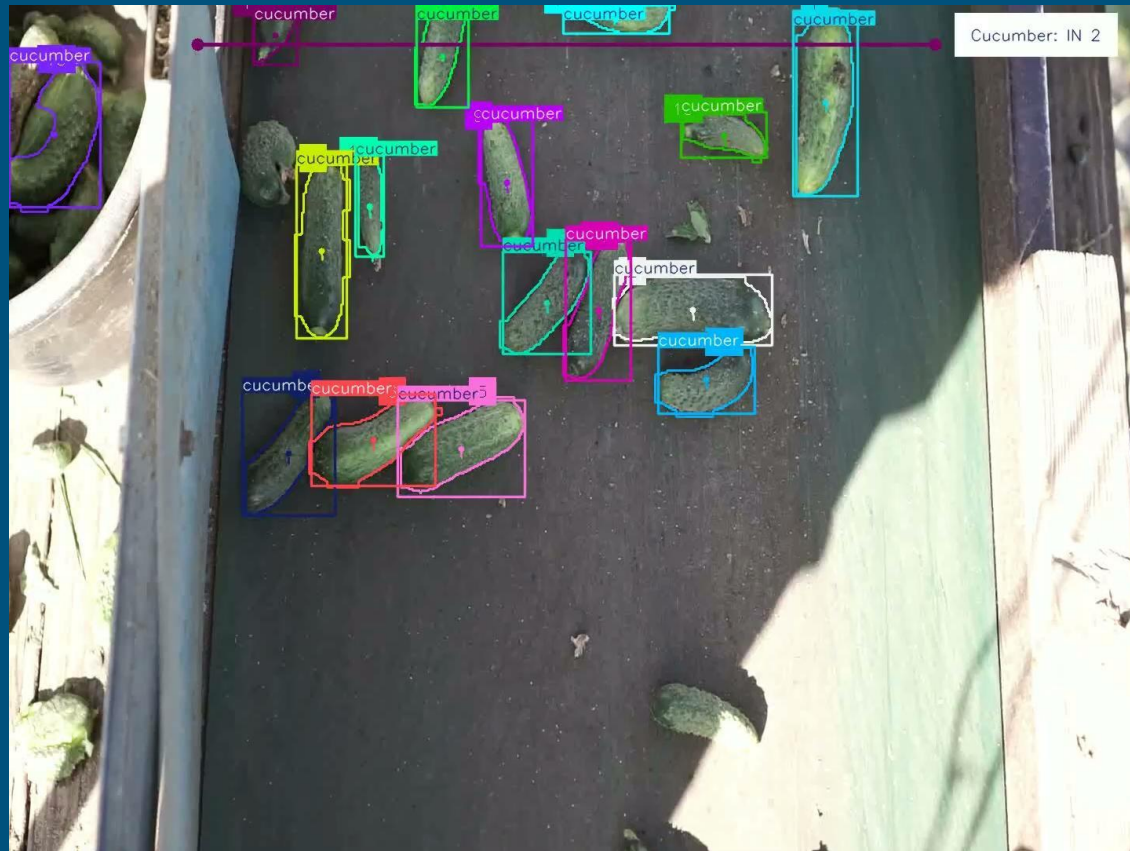


Methods summary:

- 1) Take pretrained YOLOv8 object detection/segmentation model from ultralytics
- 2) Train on hand-labelled cucumber photos
- 3) Use finetuned model for tracking and counting cucumbers in video data
- 4) Goal: Count the cucumbers and measure their sizes.

Results (1)

- Inference: Bounding boxes and segmentations
- “nano” model, quantized for speed
- ~0.3 sec per predicted frame (on my CPU)
- 43 -> 68!
- “68 cucumbers counted in 396 frames, i.e. 4.29 cucumbers per second.”



Results (2)

Live* demo!

*(640x480 px, 4 fps)

Next steps:

- ~~1) Improve object counting algorithm~~
- 2) Cucumber area/volume integration:
 - a) NB: Segmentations *change* over time
 - b) Idea: Sum over pixels when the Bbox is counted
 - c) Calibration
- 3) Think about occlusion, i.e. cucumbers lying on top of each other
- 0) Have a pickle!

Thank you for your attention!



Questions?

bjoern.opitz@posteo.de

<https://github.com/hrbjoern/CucumberIntelligence/>

(will be made public soon)

Thanks to:

- Krzysztof for being my mentor
- My former colleague Berry for having the idea and writing the first software
- Secret Santa who gave me the data
- ... yes, also the interns who did the scrappy annotations
- Ultralytics
- Claude and ChatGPT (Liar!)
- **Everybody at DSR and in Batch 39!**

Backup

Live demo emergency
replacement:



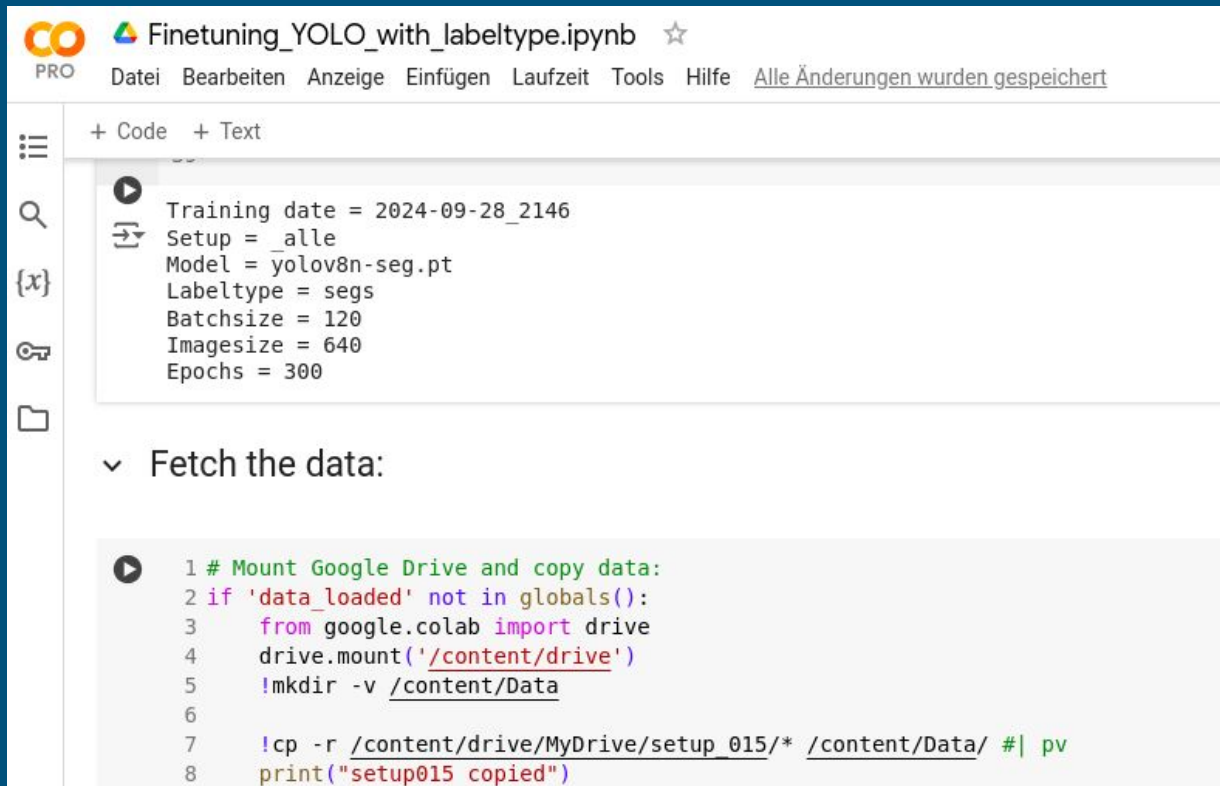
Backup: Object counter

```
# Count objects using line
elif len(self.reg_pts) == 2:
    if prev_position is not None and track_id not in self.count_ids:
        # Check if the object's movement segment intersects the counting line
        if LineString([(prev_position[0], prev_position[1]), (box[0], box[1])]).intersects(self.counting_line_segment):
            self.count_ids.append(track_id)

        # Determine the direction of movement (IN or OUT)
        dx = (box[0] - prev_position[0]) * (self.counting_region.centroid.x - prev_position[0])
        dy = (box[1] - prev_position[1]) * (self.counting_region.centroid.y - prev_position[1])
        if dx > 0 and dy > 0:
            self.in_counts += 1
            self.class_wise_count[self.names[cls]]["IN"] += 1
        else:
            self.out_counts += 1
            self.class_wise_count[self.names[cls]]["OUT"] += 1
```

Backup:

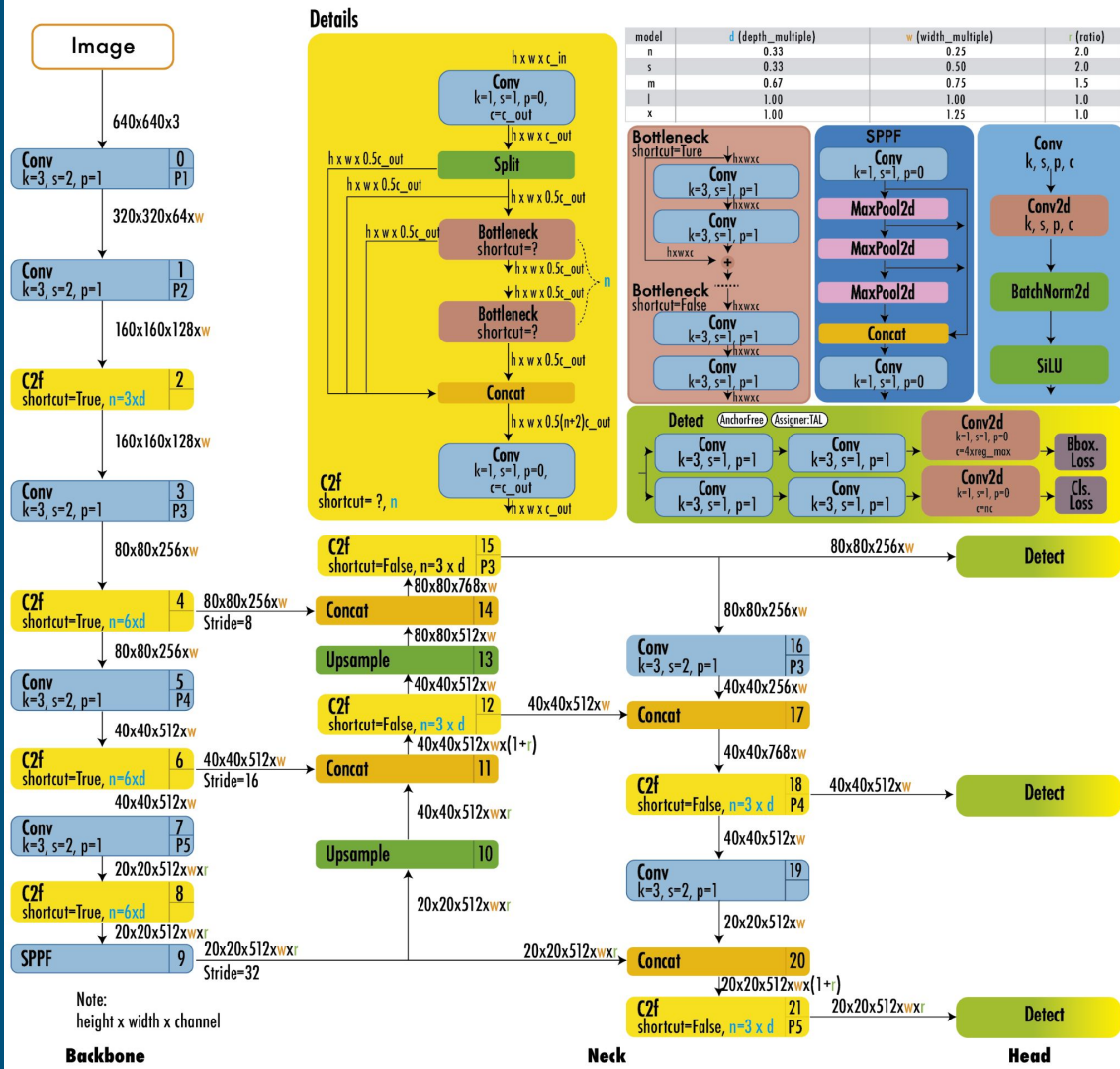
Training on Colab:
roughly 1-2 h per run
on an L4



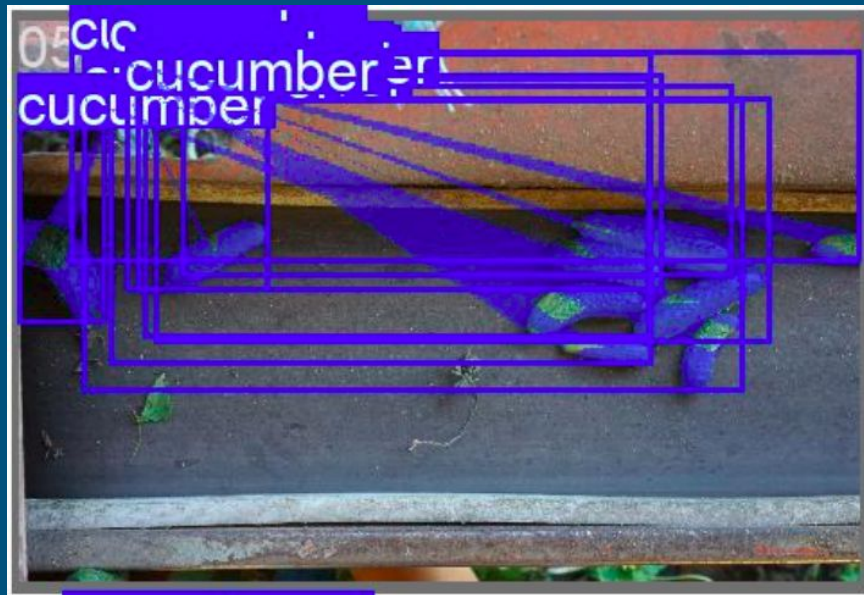
The screenshot shows a Google Colab notebook interface. At the top, the title bar reads 'Finetuning_YOLO_with_labeltype.ipynb' with a star icon. Below the title bar, a menu bar contains 'Datei', 'Bearbeiten', 'Anzeige', 'Einfügen', 'Laufzeit', 'Tools', 'Hilfe', and a link 'Alle Änderungen wurden gespeichert'. The notebook has two tabs: '+ Code' (selected) and '+ Text'. On the left sidebar, there are icons for a menu, search, variables, keys, and files. The main code area contains two code cells. The first cell is a text cell with a play icon and contains the following text: 'Training date = 2024-09-28_2146', 'Setup = _alle', 'Model = yolov8n-seg.pt', 'Labeltype = segs', 'Batchsize = 120', 'Imagesize = 640', and 'Epochs = 300'. The second cell is a code cell with a play icon and contains a Python script to mount Google Drive and copy data. The script is as follows:

```
1 # Mount Google Drive and copy data:
2 if 'data_loaded' not in globals():
3     from google.colab import drive
4     drive.mount('/content/drive')
5     !mkdir -v /content/Data
6
7     !cp -r /content/drive/MyDrive/setup_015/* /content/Data/ #| pv
8     print("setup015 copied")
```


YOLOv8 architecture



Damn you, ChatGPT



YOLO format:

<category_id> <x_center> <y_center> <width> <height> <x1> <y1> ... <xn> <yn>

... sounds convincing, isn't true.

Where the cucumbers grow

Spreewald gherkins

Article Talk

✎



Spreewald gherkins (German: **Spreewälder Gurken** or **Spreewaldgurken**) are a specialty [pickled cucumber](#) from [Brandenburg](#), which are protected by the [EU](#) as a [Protected Geographical Indication](#) (PGI).

Spreewald gherkins

