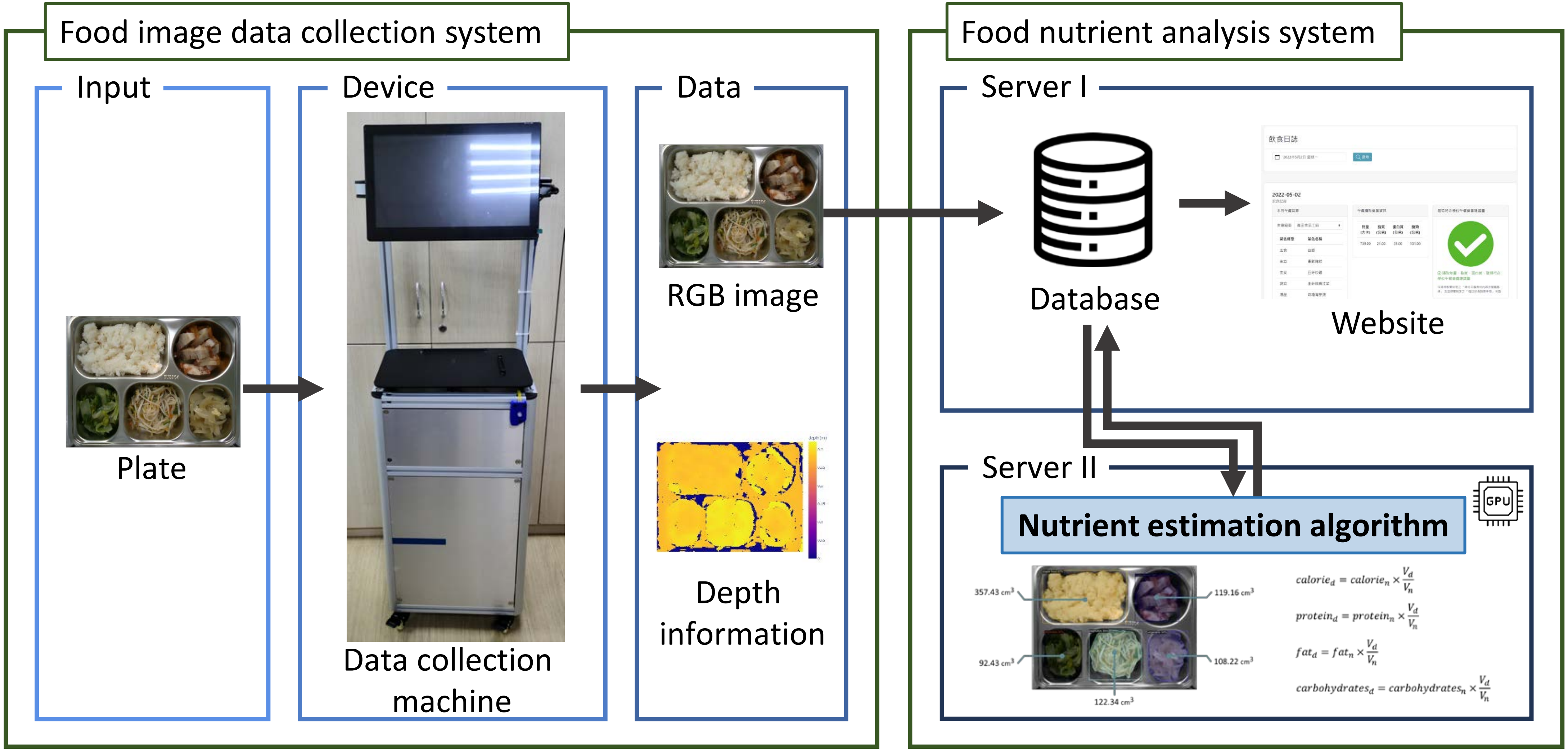
Food Instance Segmentation

Food Nutrient Analysis



$$calorie_d = calorie_n \times \frac{V_d}{V_n}$$

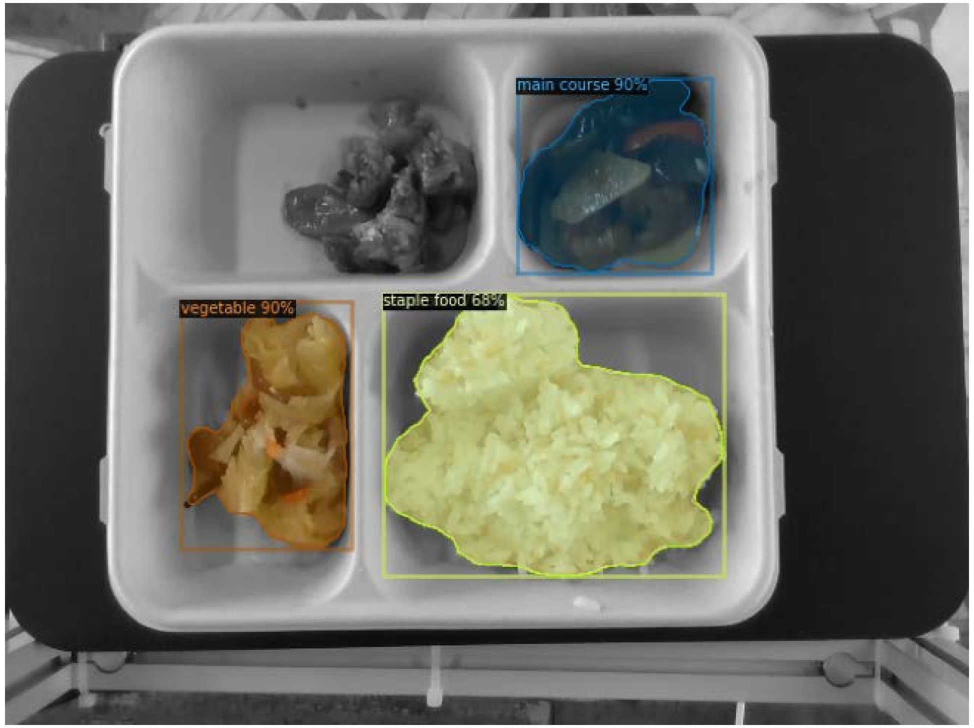
$$protein_d = protein_n \times \frac{V_d}{V_n}$$

$$fat_d = fat_n \times \frac{V_d}{V_n}$$

$$carbohydrates_d = carbohydrates_n \times \frac{V_d}{V_n}$$

Instance Segmentation

- Detecting instances of objects and demarcating their boundaries



Grading

		Points
Annotation		15
Code submission	code	15
	model	10
Report	model introduction	15
	how you train model	15
	training results	15
Test results		15

Data

- Training data
  - 455 images

Category	# instance
staple food	442
main course	434
side dish	394
vegetable	708
total	1978

- Validation data
  - 93 images

Category	# instance
staple food	89
main course	87
side dish	82
vegetable	146
total	404

Download link: https://drive.google.com/file/d/1p5EL-gYd6KKggaHMS4GOXR_DligvcjTl/view?usp=share_link

Process

ANNOTATION

MODEL CONSTRUCTION

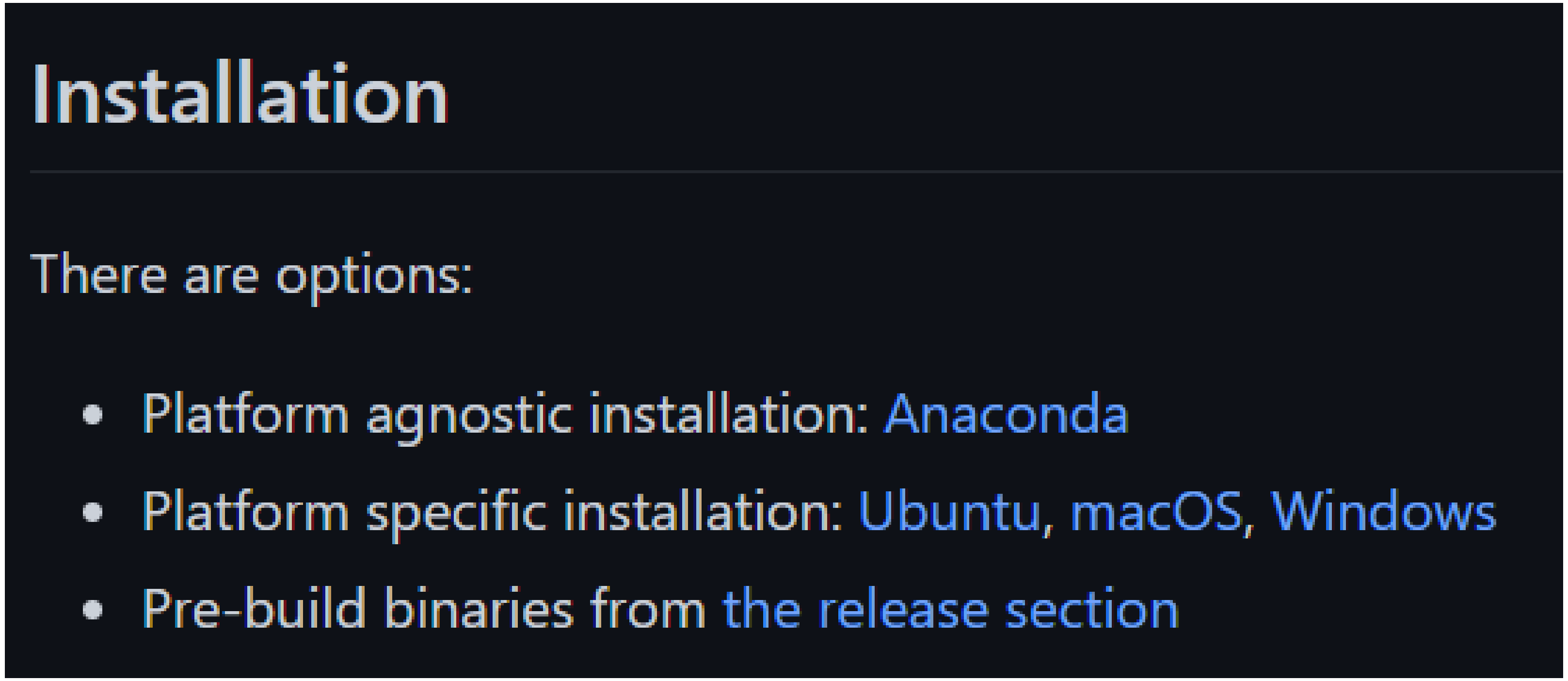
ANNOTATION FORMAT TRANSFORM

MODEL TRAINING

EVALUATION

Annotation

1. Install labelme



Installation

There are options:

- Platform agnostic installation: Anaconda
- Platform specific installation: Ubuntu, macOS, Windows
- Pre-build binaries from the release section

Annotation

2. Open image directory

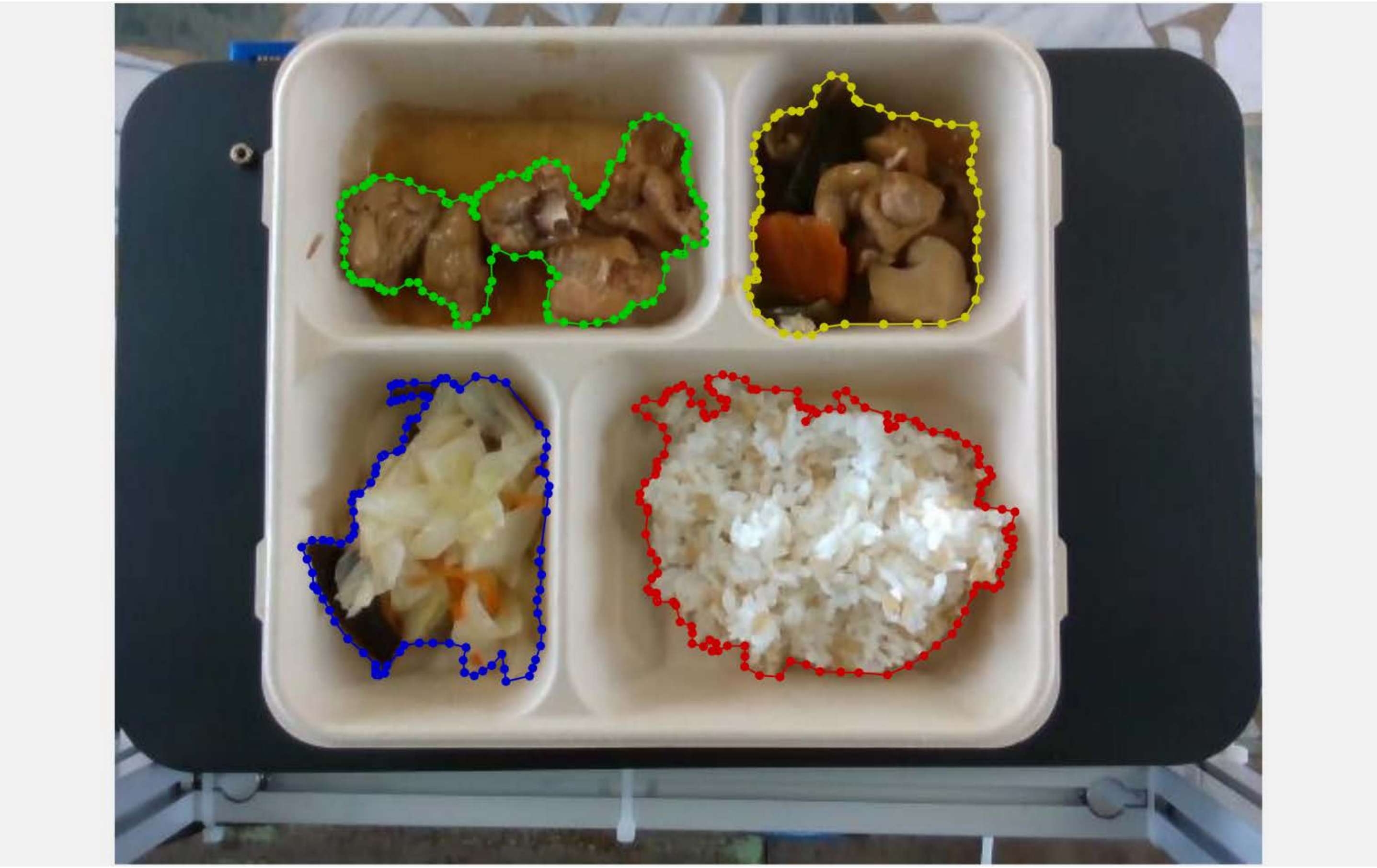
Annotation

3. Change output directory

Remember to click **“Save Automatically”**

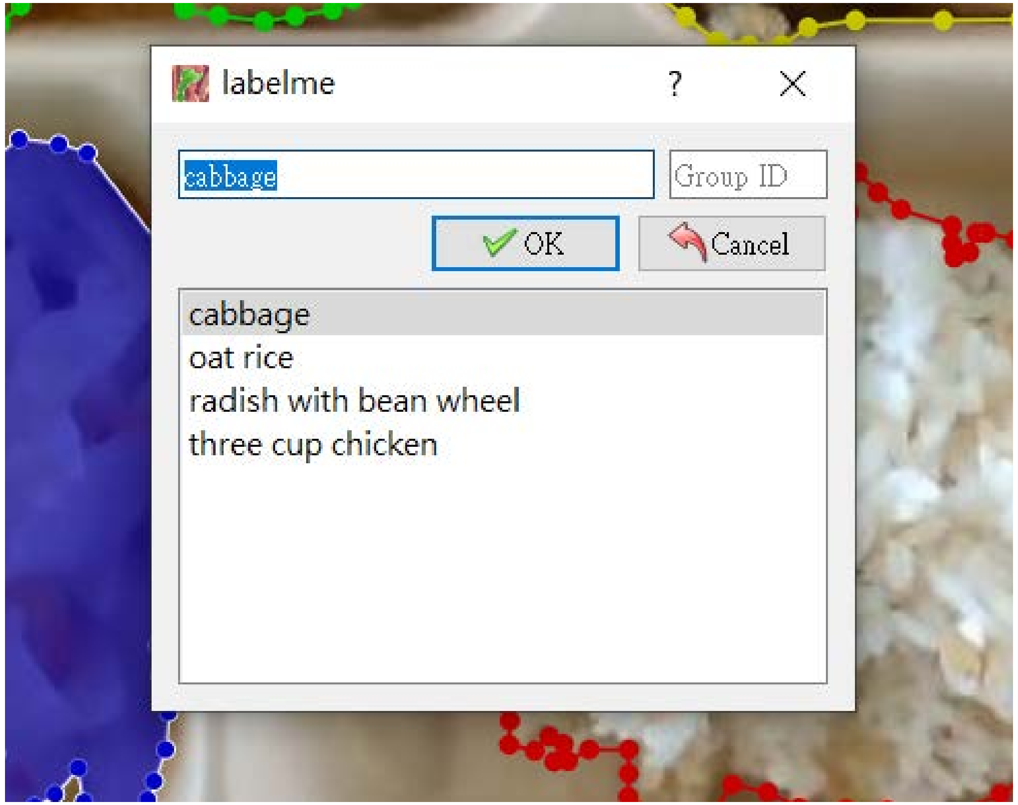
Annotation

4. Click “Create Polygons” and mark along the food contour



Annotation

5. After marking, set the label of the food



cabbage
oat rice
radish with bean wheel
three cup chicken

Annotation

6. If tray is empty, skip it

Annotation

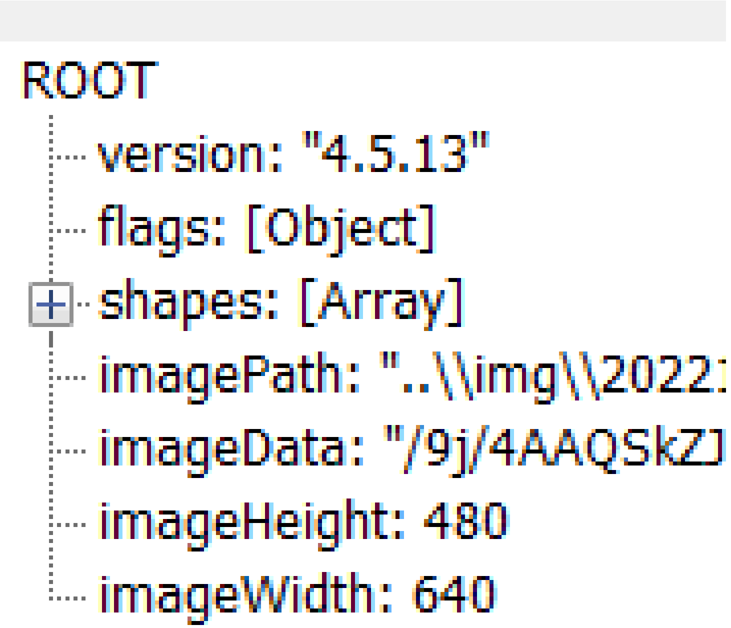
- For more label information, please refer to: https://github.com/huang0819/food-segmentation#label

Model Construction

- Select instance segmentation model
  - EVA, GitHub
  - Mask DINO, GitHub
  - Swin Transformer V2, GitHub
  - …

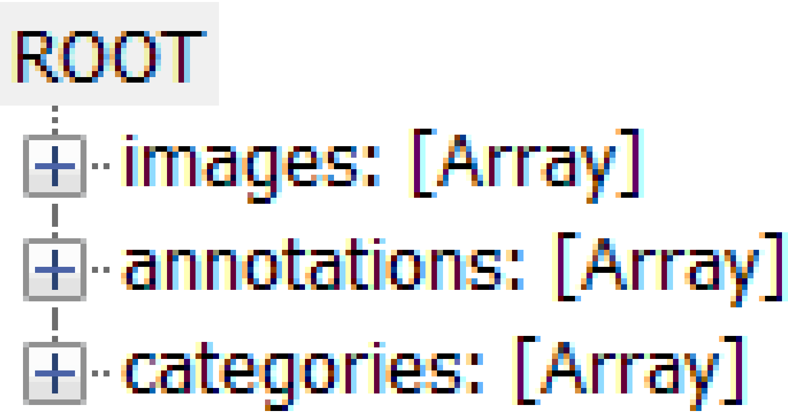
Annotation Format Transform

- Check the format specified by your model
- In this tutorial, we used COCO dataset format



```
ROOT
├ version: "4.5.13"
├ flags: [Object]
⊞ shapes: [Array]
├ imagePath: "..\\img\\2022
├ imageData: "/9j/4AAQSkZJ
├ imageHeight: 480
└ imageWidth: 640
```

20221123174818_357_0.json
20221123174858_265_0.json
⋮



```
ROOT
⊞ images: [Array]
⊞ annotations: [Array]
⊞ categories: [Array]
```

dataset.json

Model Training

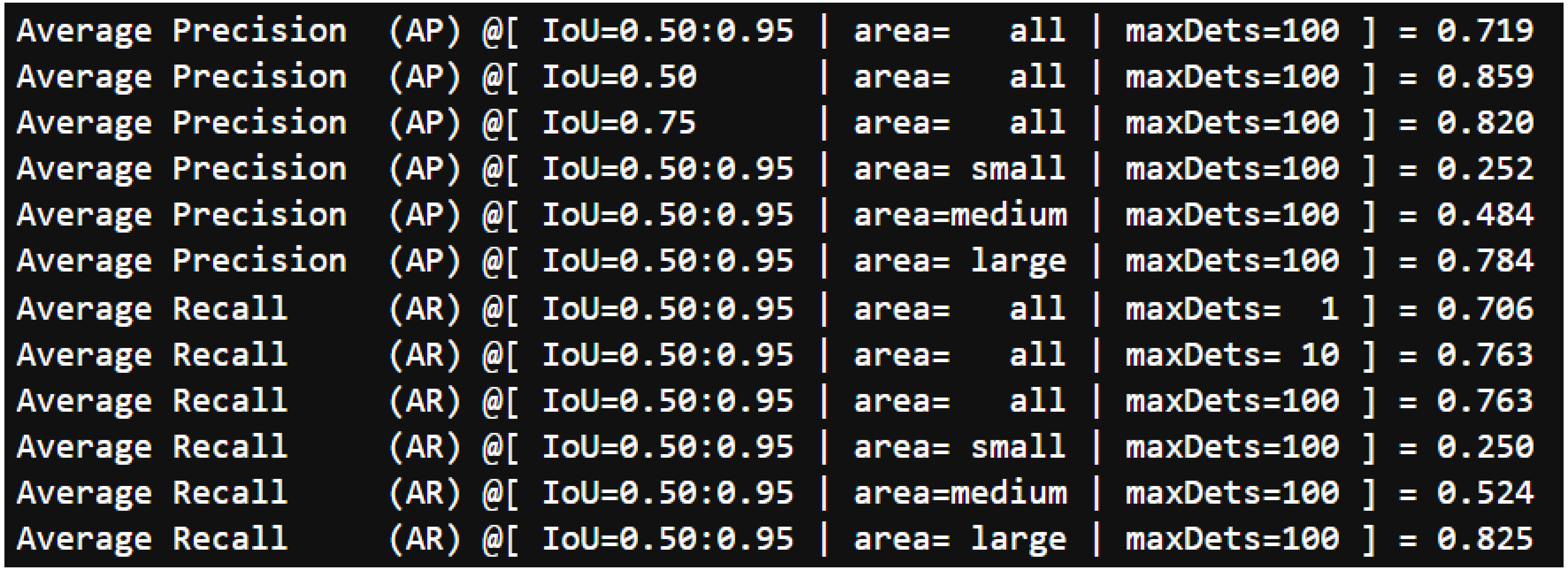
- Sample code: Mask R-CNN

Detectron2

- More model structures: https://github.com/facebookresearch/detectron2/blob/main/MODEL_ZOO.md
- Do not use test data during model training

Evaluation

- Use the trained model to predict test data and generate reports
- Evaluation sample code: https://gist.github.com/huang0819/3f6bf5b8d0f0ad5c8954c9d0ce559f02



```
Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.719
Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.859
Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.820
Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.252
Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.484
Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.784
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.706
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.763
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.763
Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.250
Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.524
Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.825
```

Any questions ?