3099704: AI for Digital Health

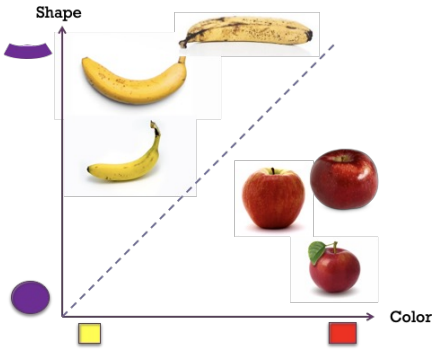
Data Labeling

Prof. Peerapon Vateekul, Ph.D.
Peerapon.v@chula.ac.th

Data is the most important factor.

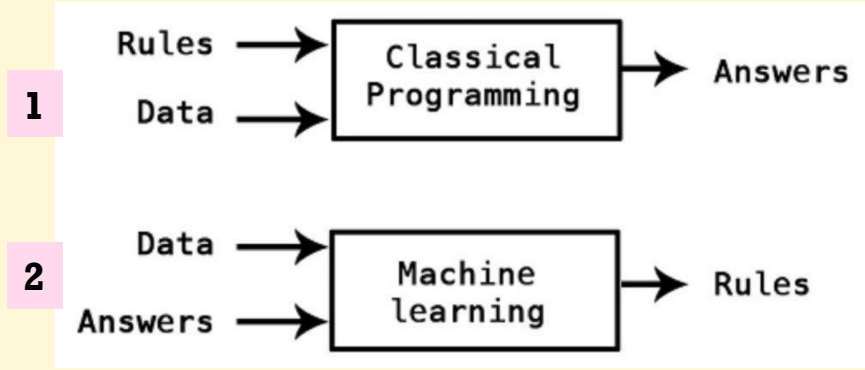
- We have a lot of data collected. Can we train AI model?
- Fact: We need to have labelled data.

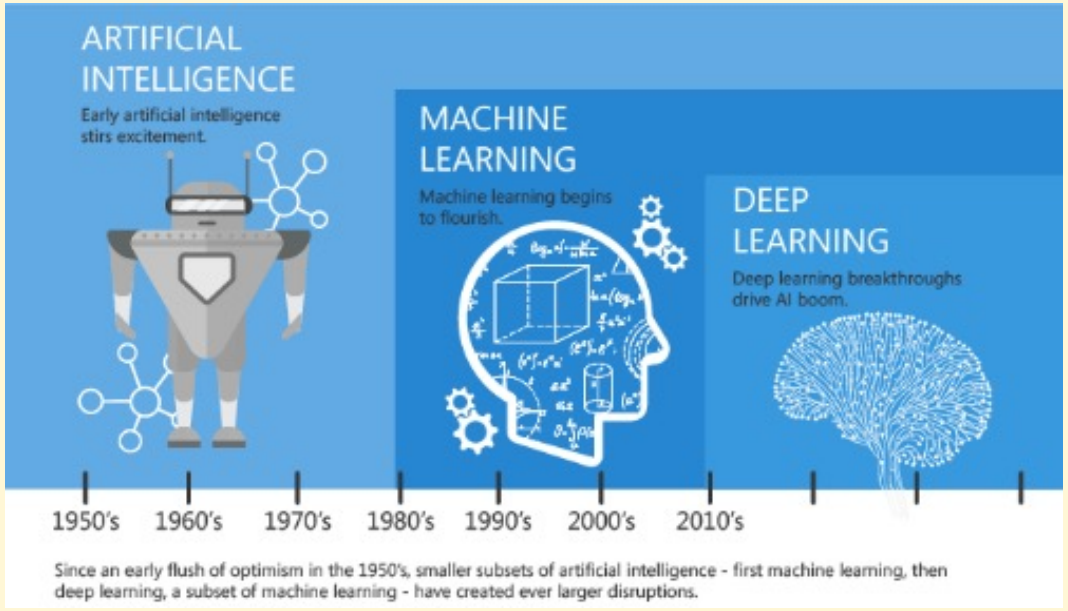
- What is the amount of data that we should have to train the model?
- **As many as possible!** Easy problem requires less data.
- Less than 100, it seems to be impossible.
- Recommend to have at least 1,000 cases.
- If not enough, transfer learning.

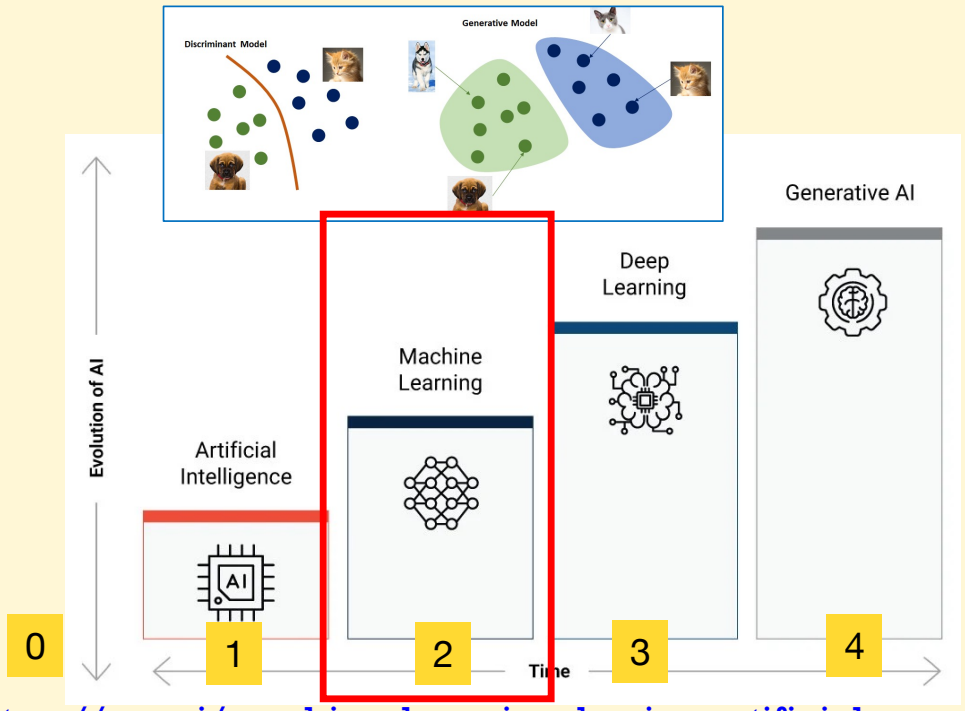


AI = Automation

- 0) Not AI Solution (not automatic)
- 1) Rule-based AI
- 2) Machine Learning (ML)

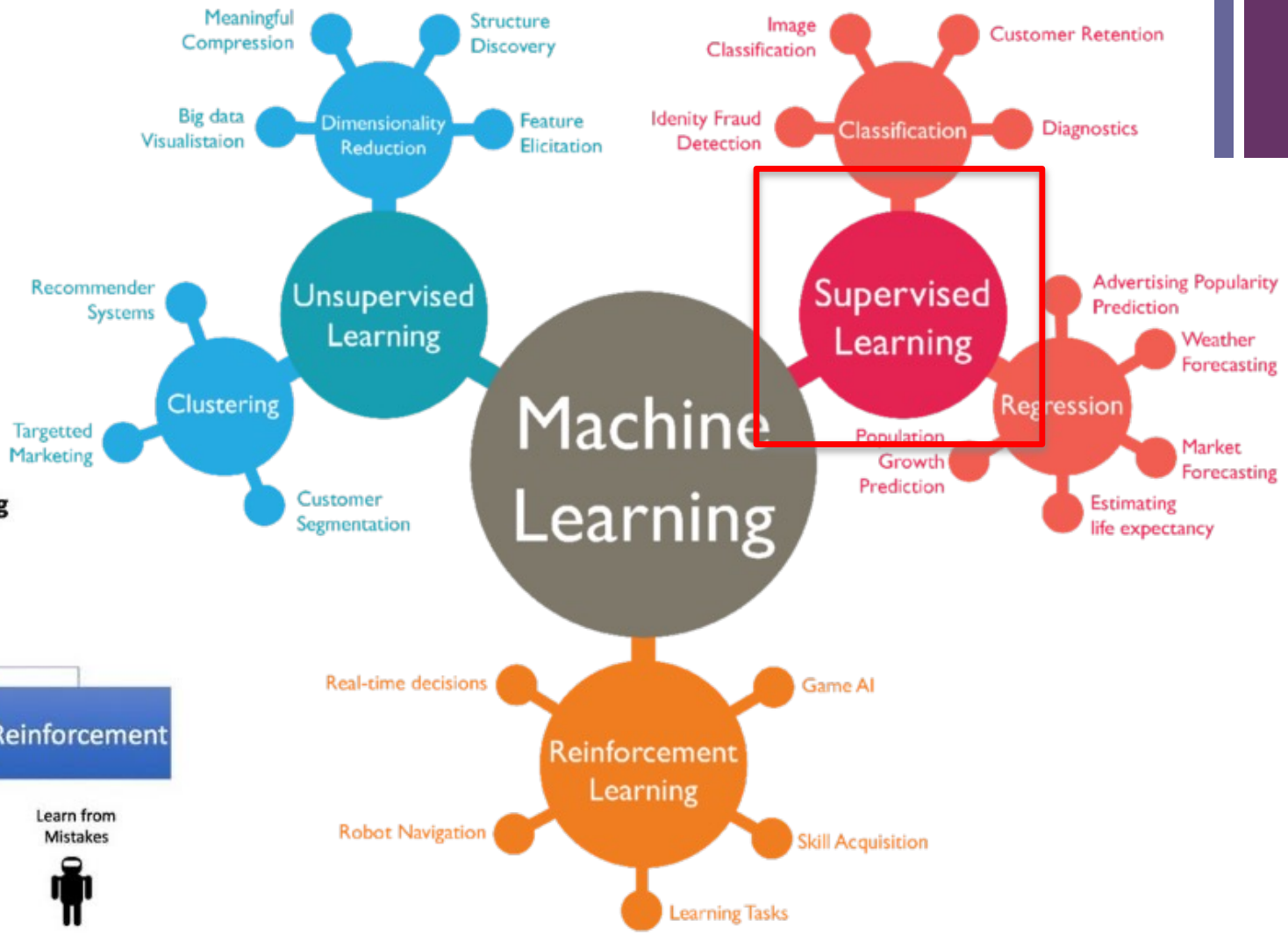




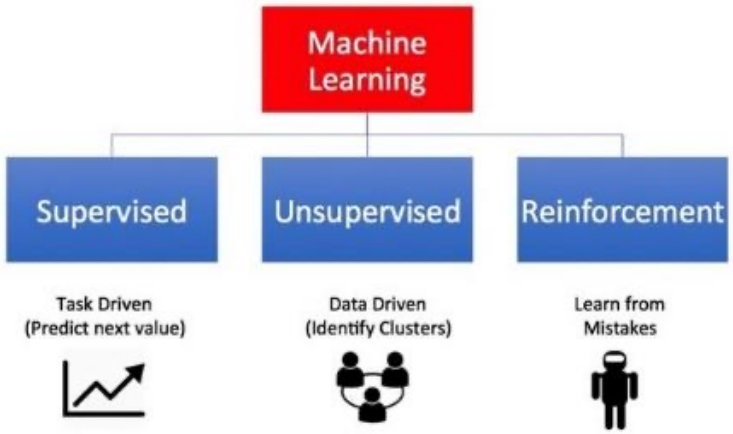


https://mc.ai/machine-learning-basics-artificial-intelligence-machine-learning-and-deep-learning/

Machine Learning



Types of Machine Learning

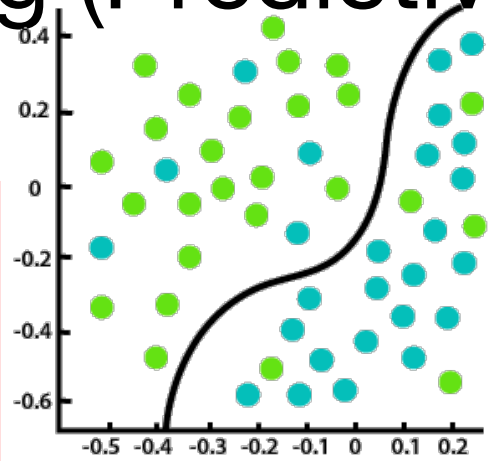


Supervised Learning (Predictive Task)

inputs | target

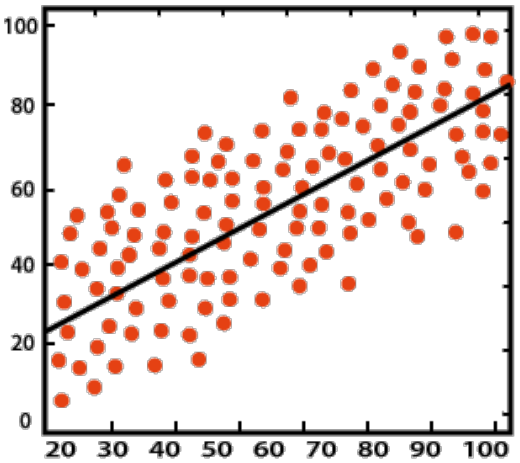
Age	Temp	Gender	Smell	Covid
25	39.0	Female	No	Yes
35	38.9	Female	No	Yes
32	36.5	Male	Yes	No

- Goal: To learn a prediction model mapping from inputs to output.
- Data without label (answer) is meaningless!
- Label should be provided by experts!



Classification

- Target is categorical variable.
- Example
- Covid diagnosis (yes/no)
- Disease diagnosis from gait information:
  - 1) Normal,
  - 2) Sick/Knee OA
  - 3) Sick/Parkinson



Regression

- Target is numeric variable.
- Example
- PD's state diagnosis from movement data.
- Glucose level prediction from breath particles.

There are two main processes: Train/Test

1) Training Phase: Model Construction

Training Data

inputs

target

Age	Income	Gender	Province	Purchase
25	25,000	Female	Bangkok	Yes
35	50,000	Female	Nontaburi	Yes
32	35,000	Male	Bangkok	No

2) Testing Phase: Model Evaluation, Model Assessment
Also called "prediction, inference, scoring"

Testing Data

Age	Income	Gender	Province	Purchase
25	25,000	Female	Bangkok	?

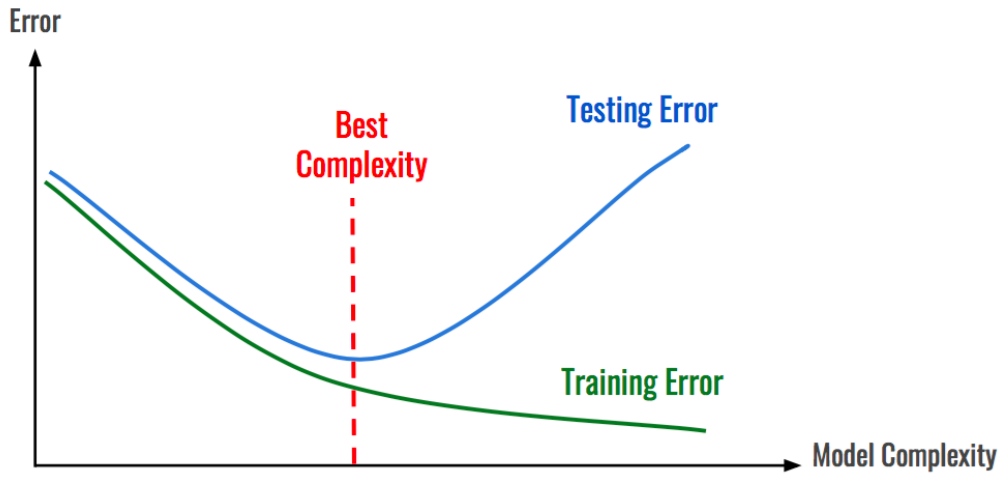
Evaluation (Train/Test Split)

Training Data

Age	Income	Purchase
25	25,000	Yes
35	50,000	Yes
32	35,000	No

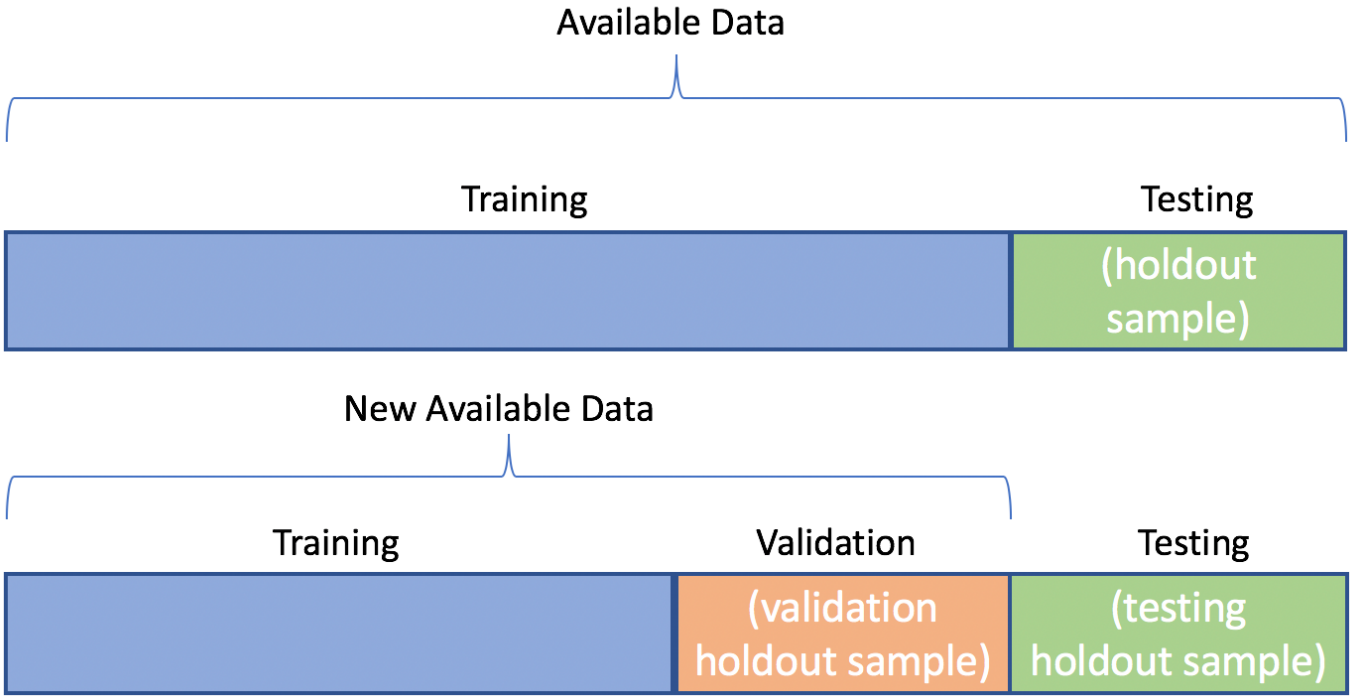
Testing Data

Age	Income	Purchase
27	35,000	Yes
23	20,000	No
45	34,000	No



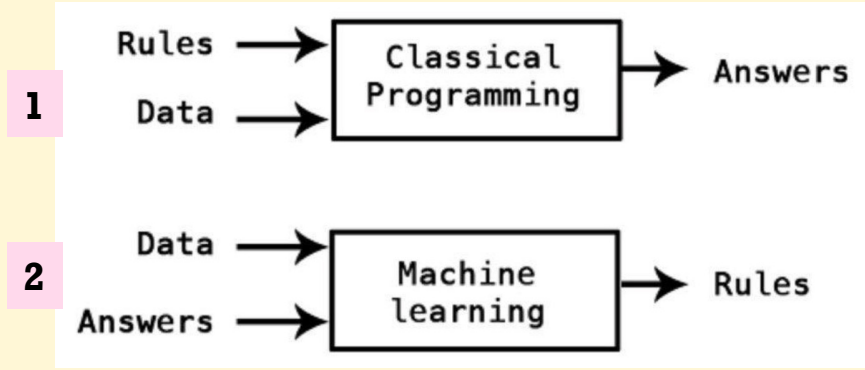
Train (Validation) & Test

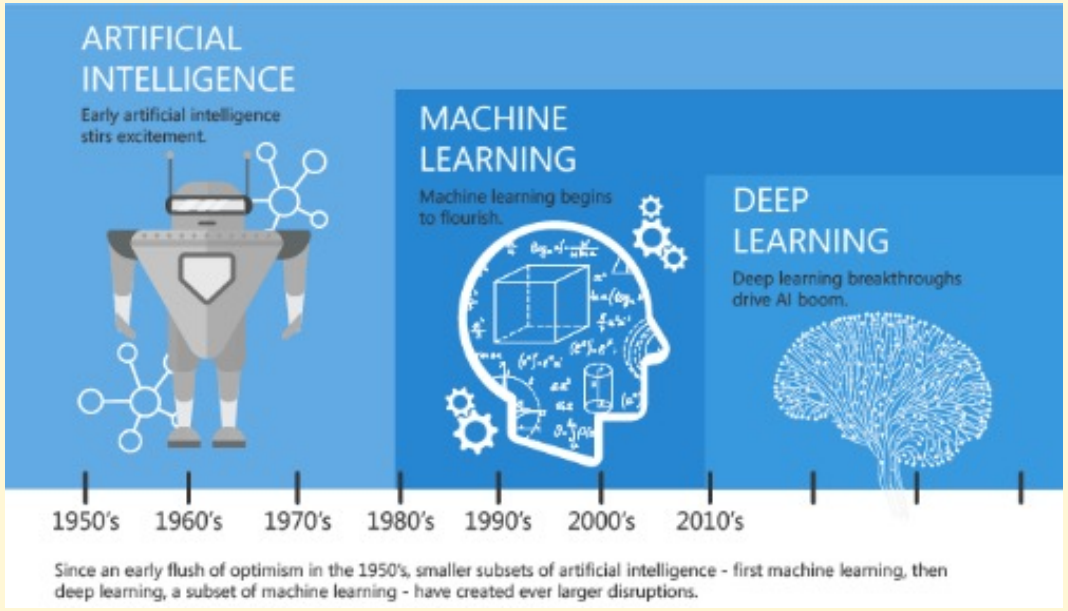
- Training data = Textbook
- Validation data = Exercise
- Testing data = Final exam

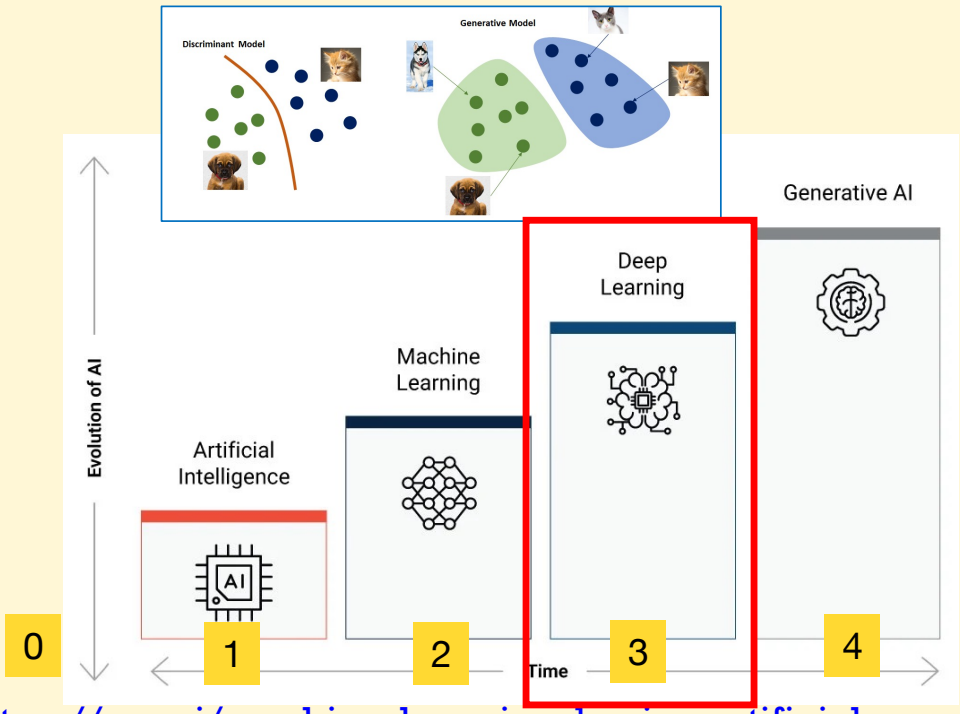


AI = Automation

- 0) Not AI Solution (not automatic)
- 1) Rule-based AI
- 2) Machine Learning (ML)

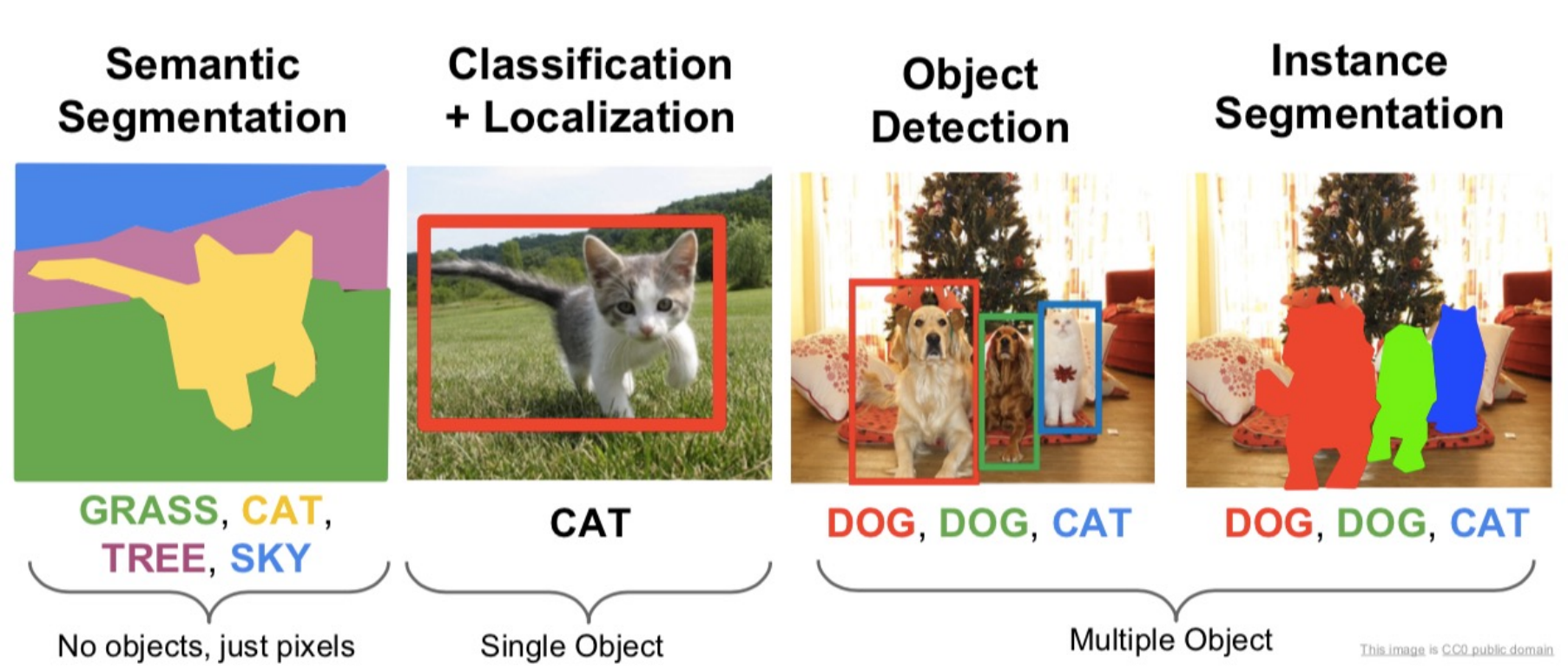






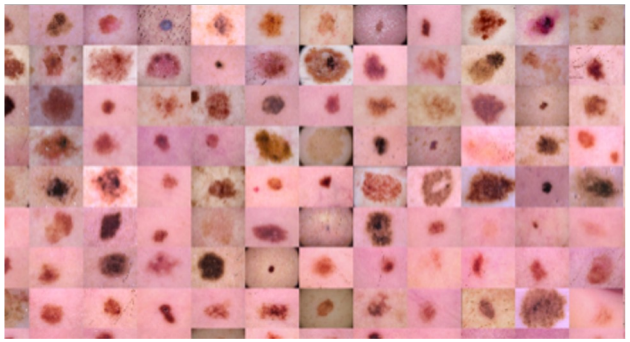
https://mc.ai/machine-learning-basics-artificial-intelligence-machine-learning-and-deep-learning/

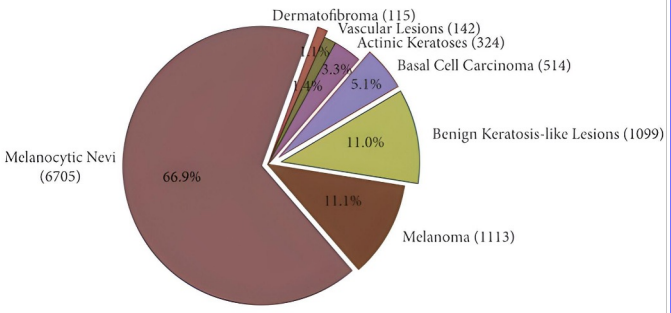
Type of image tasks: Image Classification



Dataset: Skin Cancer MNIST: HAM10000

- The dataset consists of 10015 images with 10013 labeled objects belonging to 7 skin cancer classes.
- The data contains image in JPG format and documents in JSON format
- In the experiment, we reduced the amount of data and formatted it to simplify the experiment.





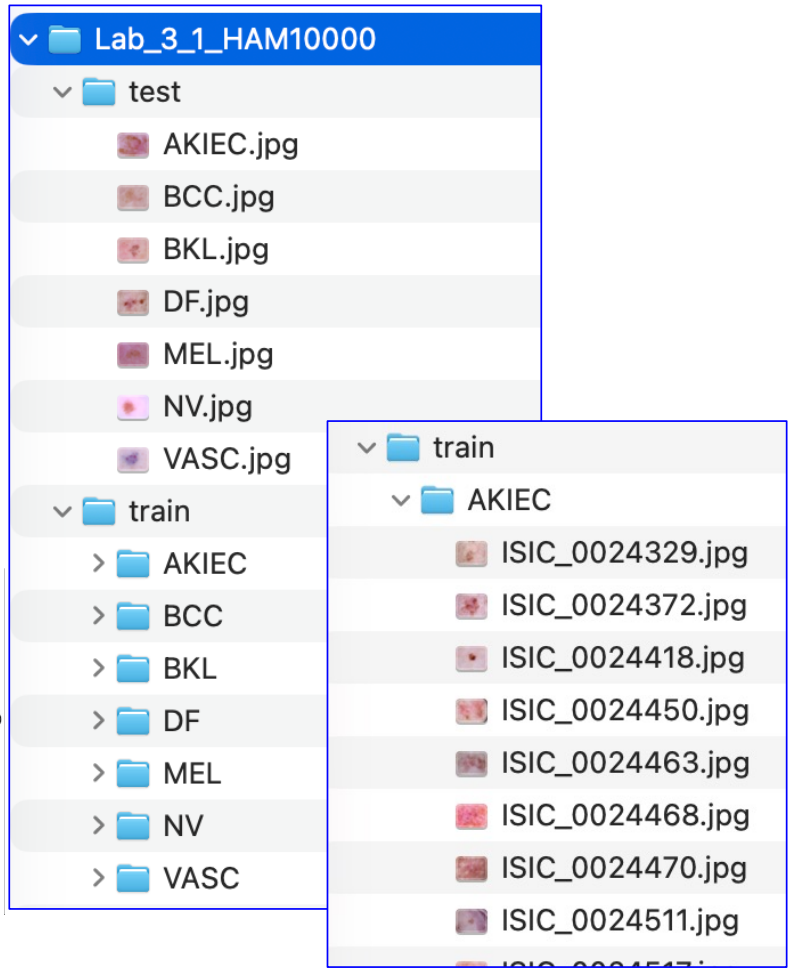
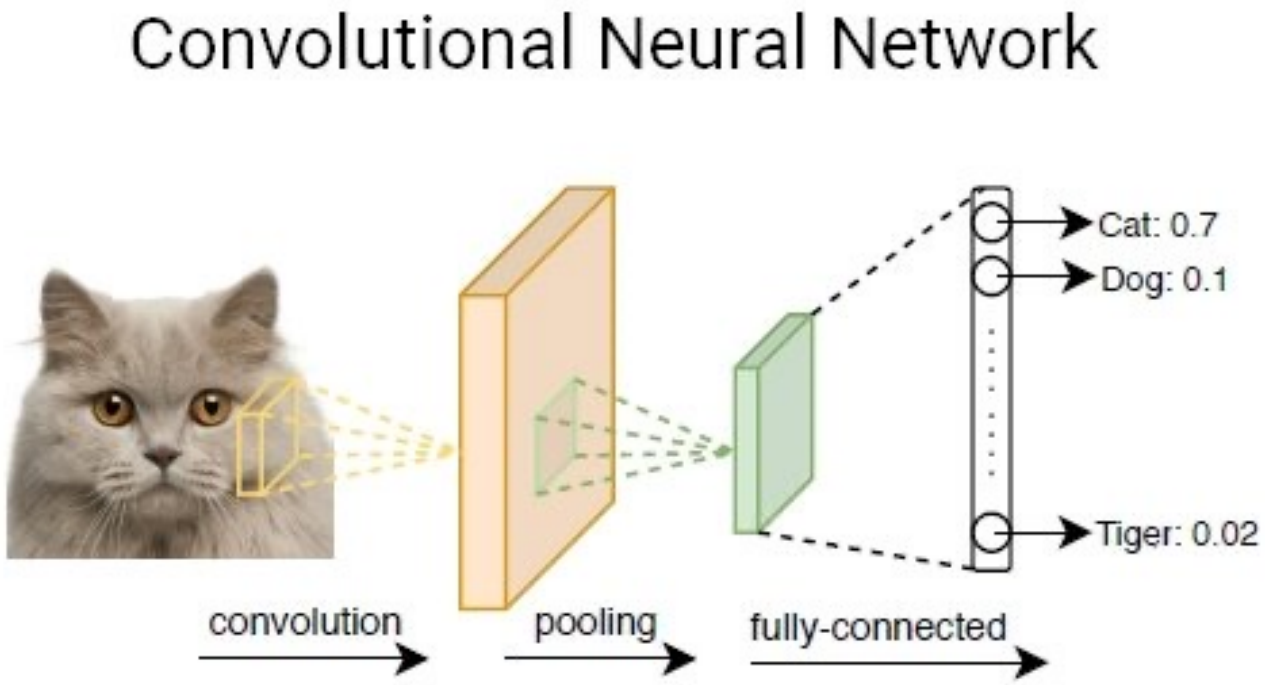
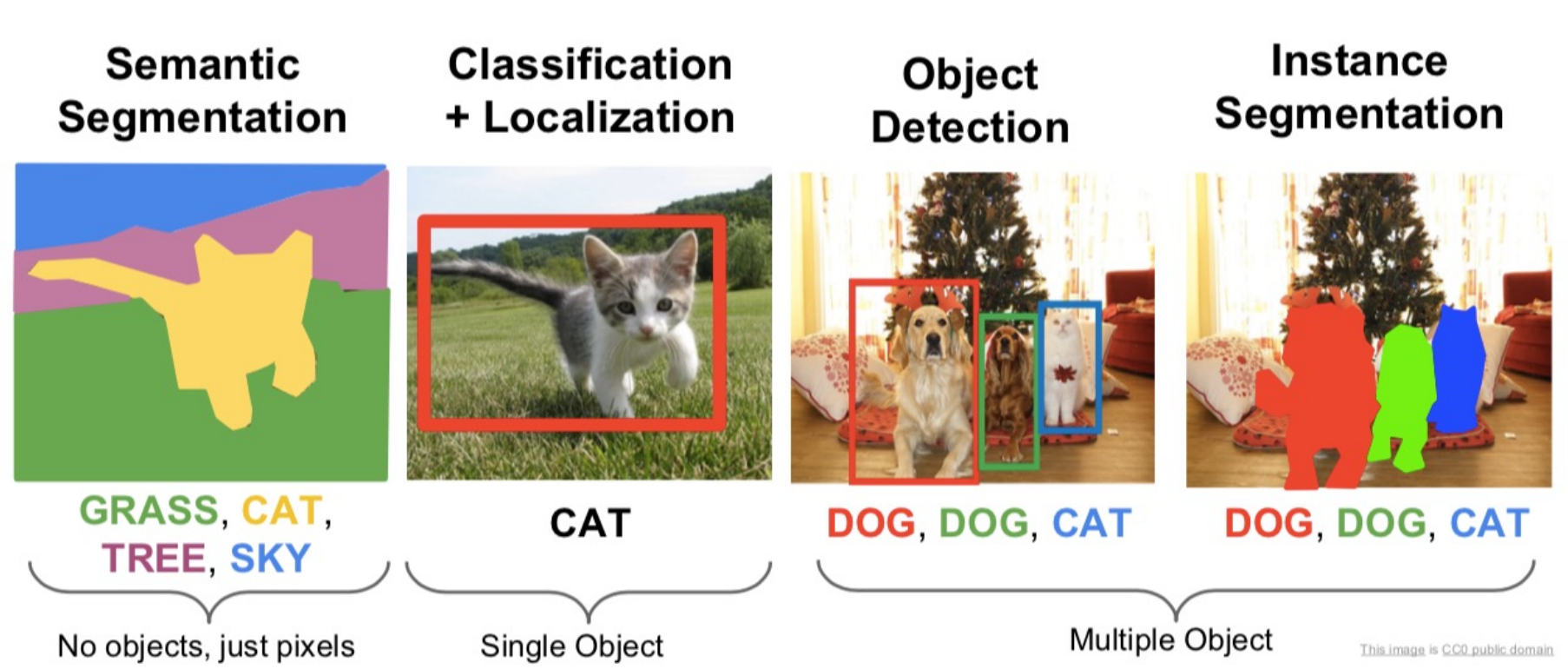


Image Classification



Type of image tasks: Object Detection



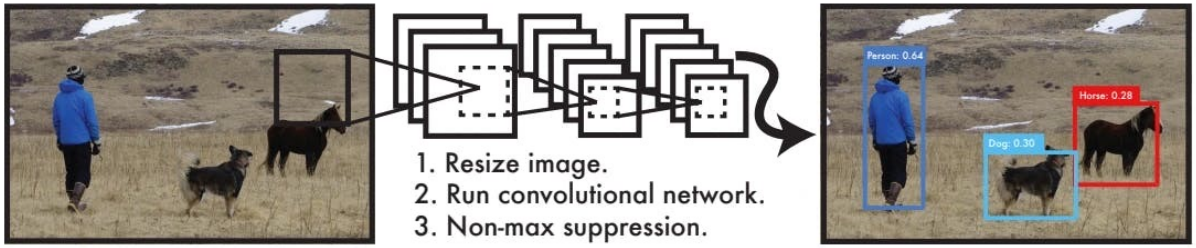
YOLO's Input

The label file corresponding to the above image contains 2 persons (class 0) and a tie (class 27):

zidane.txt

```
0  0.481719 0.634028 0.690625 0.713278
0  0.741094 0.524306 0.314750 0.933389
27 0.364844 0.795833 0.078125 0.400000
```

YOLO's output



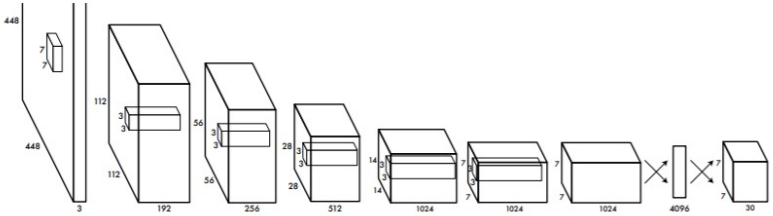
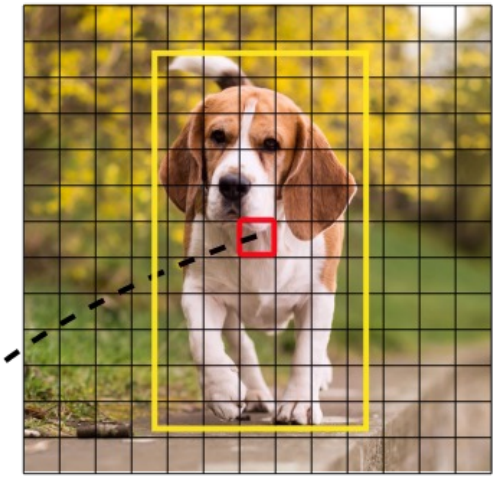
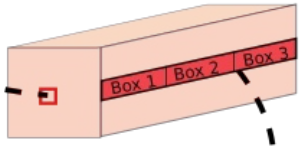


Figure 3: The Architecture. Our detection network has 24 convolutional layers followed by 2 fully connected layers. Alternating 1×1 convolutional layers reduce the features space from preceding layers. We pretrain the convolutional layers on the ImageNet classification task at half the resolution (224×224 input image) and then double the resolution for detection.

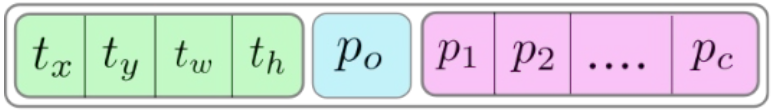
Image Grid. The Red Grid is responsible for detecting the dog



Prediction Feature Map



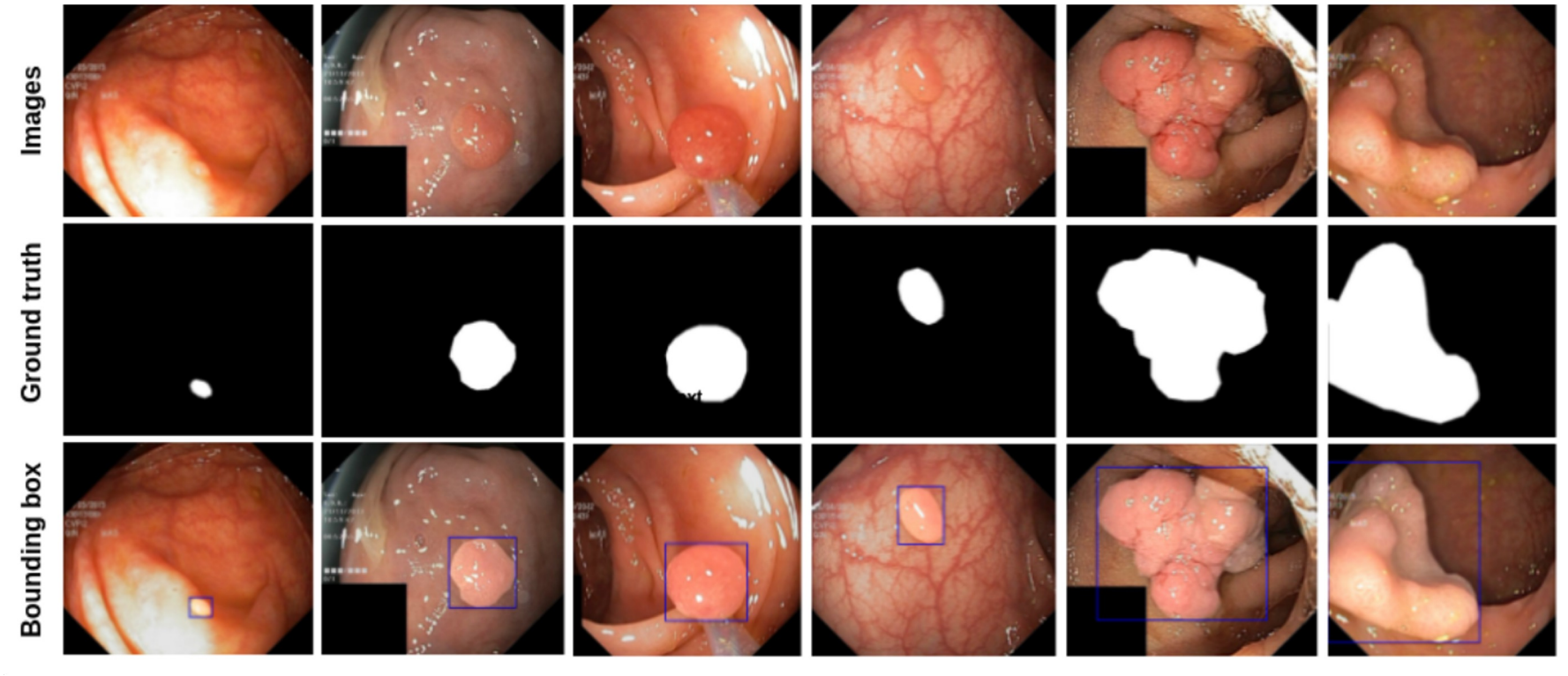
Attributes of a bounding box



https://blog.paperspace.com/how-to-implement-a-yolo-object-detector-in-pytorch/

Original Dataset

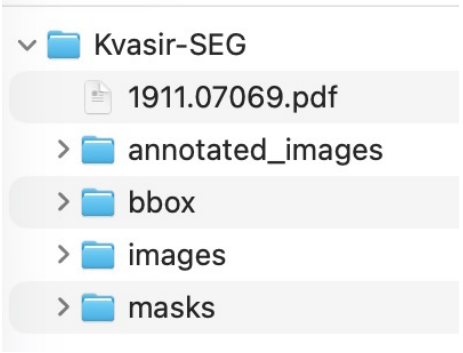
Dataset: kvasir dataset (2017)

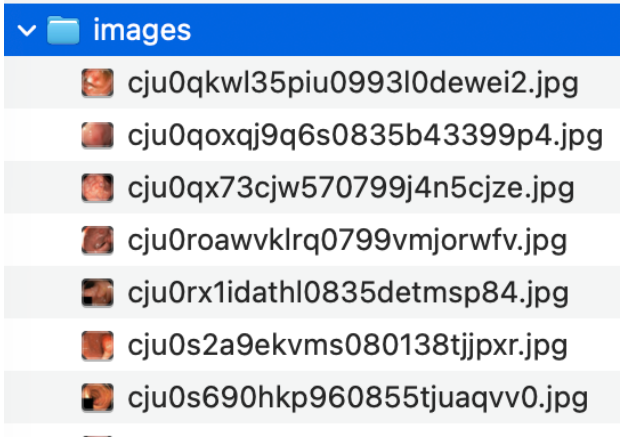


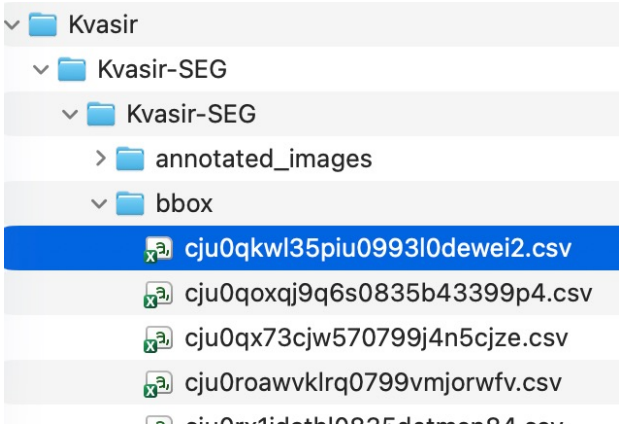
The figure shows the example images, bounding box, and mask from Kvasir-SEG. The white mask shows the area covered by the polyp region, and the background regions contain non-polyp tissue pixels.

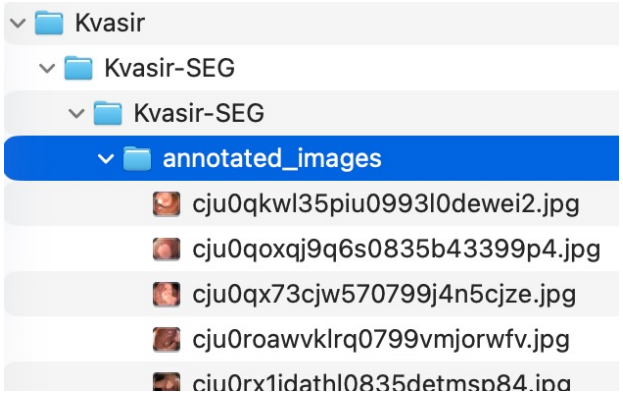
Original Dataset: Object Detection

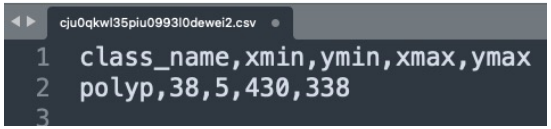
Dataset: kvasir dataset (cont.)

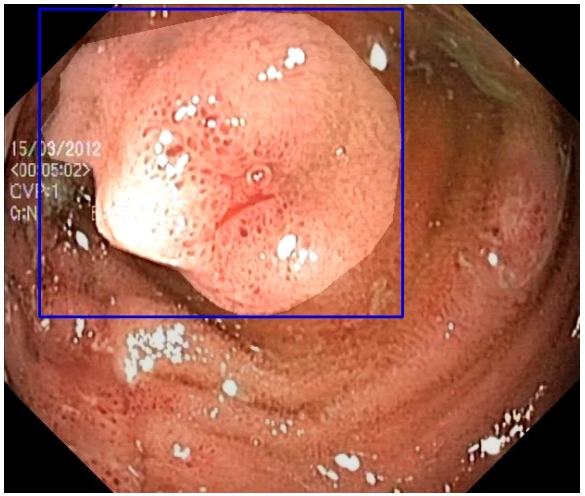






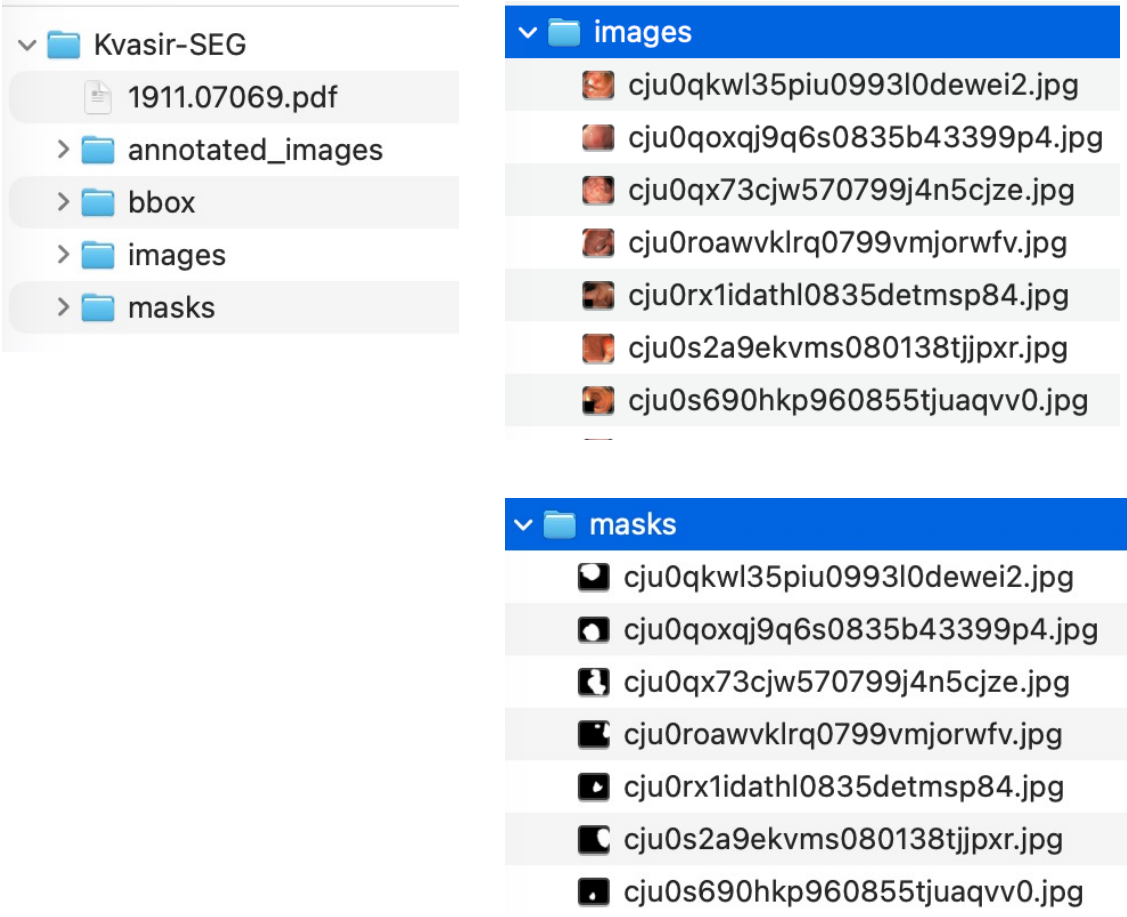


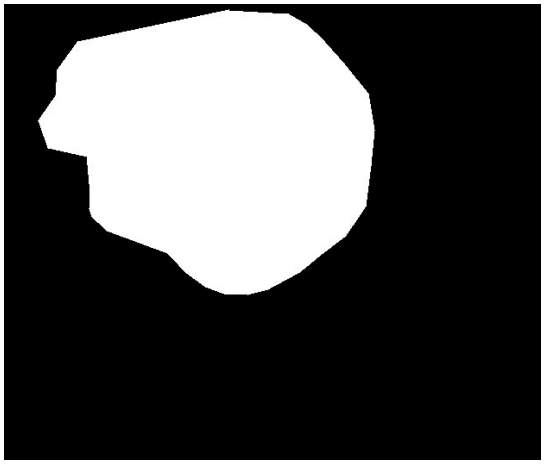




Original Dataset: Segmentation

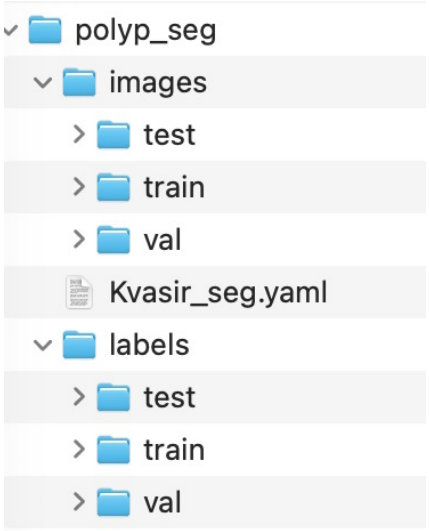
Dataset: kvasir dataset (cont.)

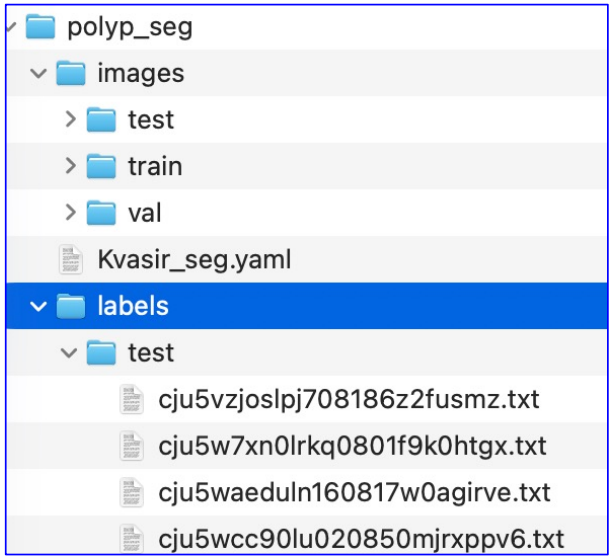


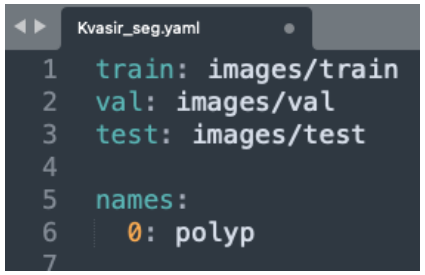


polyp_detect.zip (YOLO format)

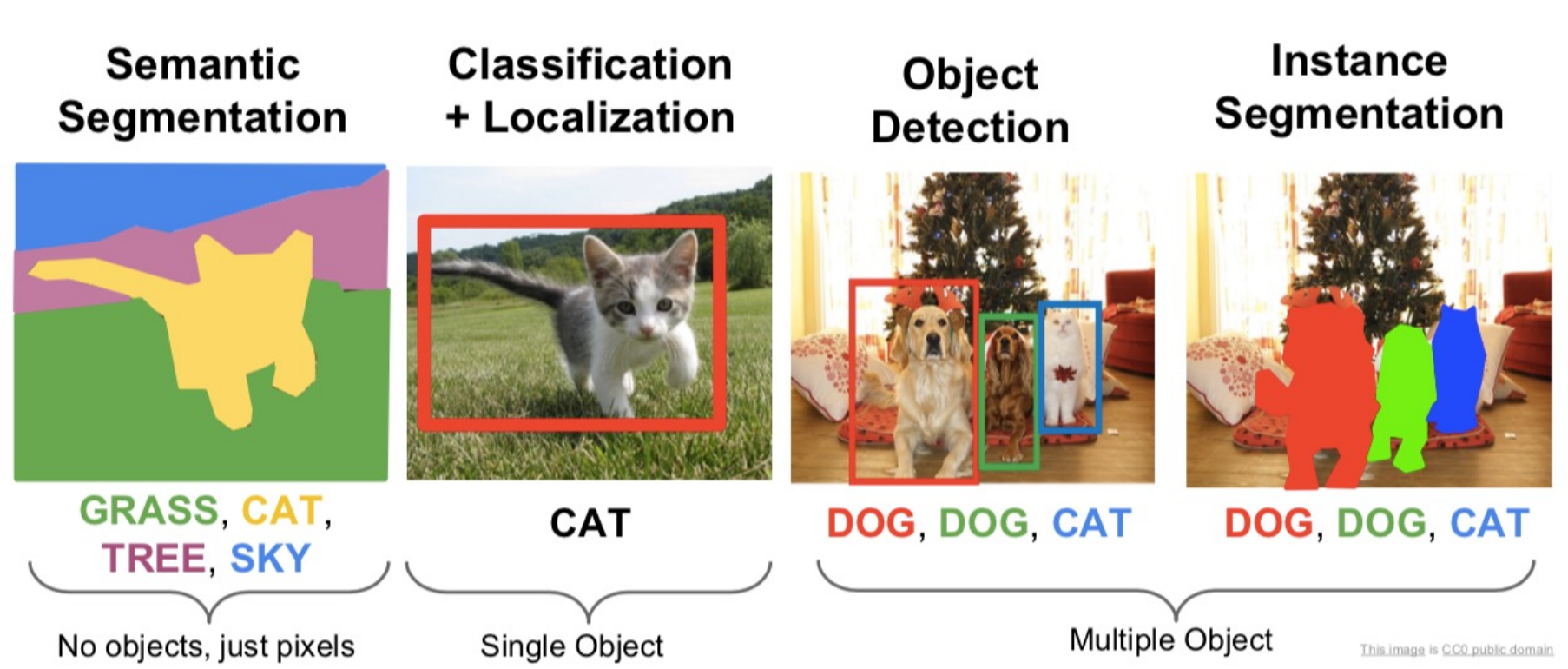
Dataset: kvasir dataset (cont.)





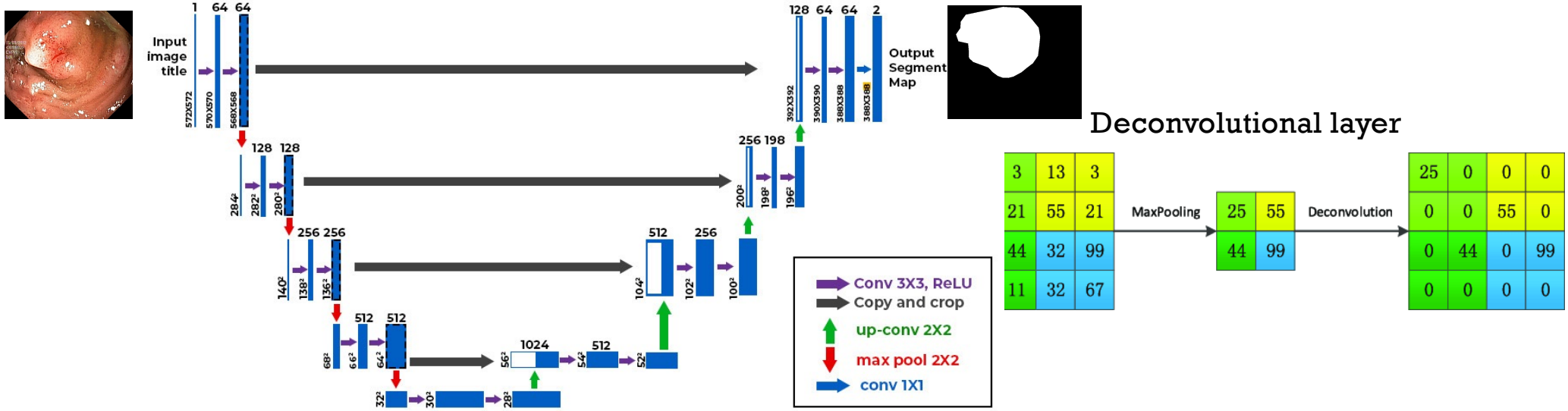


Type of image tasks: Semantic Segmentation



UNet: Encoder-Decoder Network: Encoder, Decoder, Skip Connections

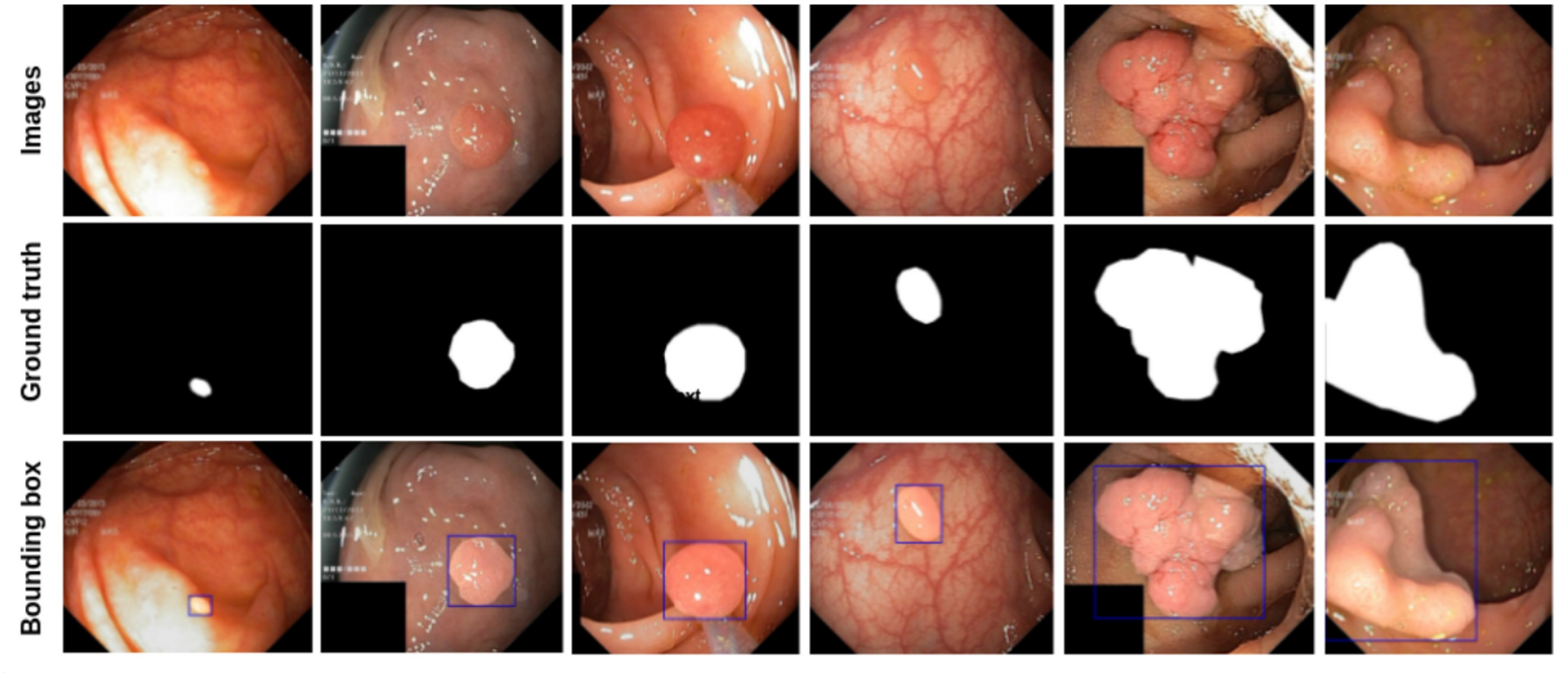
- **U-Net**



$\mathcal{L} = \lambda_1 \cdot \mathrm{CE} + \lambda_2 \cdot \mathrm{Dice}$; CE (pixel loss) & Dice (region loss; semantic)

Architecture of the U-net for a given input image. The blue boxes correspond to feature maps blocks with their denoted shapes. The white boxes correspond to the copied and cropped feature maps.
Source: O. Ronneberger et al. (2015)

Dataset: kvasir dataset (2017)



The figure shows the example images, bounding box, and mask from Kvasir-SEG. The white mask shows the area covered by the polyp region, and the background regions contain non-polyp tissue pixels.

polyp_seg.zip (YOLO format)

Dataset: kvasir dataset (cont.)

```
polyp_seg
  images
    test
    train
    val
  Kvasir_seg.yaml
  labels
    test
    train
    val
```

```
images
  test
    cju5vzjoslpj708186z2fusmz.jpg
    cju5w7xn0lrkq0801f9k0htgx.jpg
    cju5waeduln160817w0agirve.jpg
    cju5wcc90lu020850mjrxppv6.jpg
    cju5wi6bqlxy90755bu227nvb.jpg
```

```
polyp_seg
  images
    test
    train
    val
  Kvasir_seg.yaml
  labels
    test
      cju5vzjoslpj708186z2fusmz.txt
      cju5w7xn0lrkq0801f9k0htgx.txt
      cju5waeduln160817w0agirve.txt
      cju5wcc90lu020850mjrxppv6.txt
```

Kvasir_seg.yaml

```yaml
train: images/train
val: images/val
test: images/test

names:
  0: polyp
```

cju0qkwl35piu0993l0dewei2.csv | cju5vzjoslpj708186z2fusmz.txt

```
0 0.758842 0.342742 0.757235 0.344758 0.755627 0.344758 0.754019 0.346774
0.750804 0.346774 0.749196 0.348790 0.745981 0.348790 0.744373 0.350806
0.741158 0.350806 0.739550 0.352823 0.736334 0.352823 0.734727 0.354839
0.731511 0.354839 0.729904 0.356855 0.728296 0.356855 0.726688 0.358871
0.723473 0.358871 0.721865 0.360887 0.718650 0.360887 0.717042 0.362903
0.713826 0.362903 0.712218 0.364919 0.709003 0.364919 0.709003 0.366935
```

code

```
<class_id> x1 y1 x2 y2 x3 y3 ... xn yn
```

code

```
0 0.758842 0.342742 0.757235 0.344758 0.755627 0.344758 ...
```

code

```
x_pixel = x * image_width
y_pixel = y * image_height
```

Thank you
& any questions