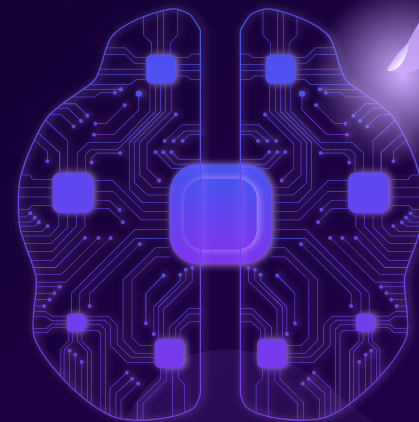


Object Detection 101

Gene, Jose, Ishrak,
and Rachel



Welcome!

Today, we will show you guys how to use one key AI tools:

- YOLOv8
- YOLOv8-obb

Computer Vision???

- Teaching computers how to see
- Images are not what our eyes see...
- Images are turned into numbers (RGB value)
- Computer reads the number and spots patterns in them.

Introduction to Image Data

[illegible]

△




```

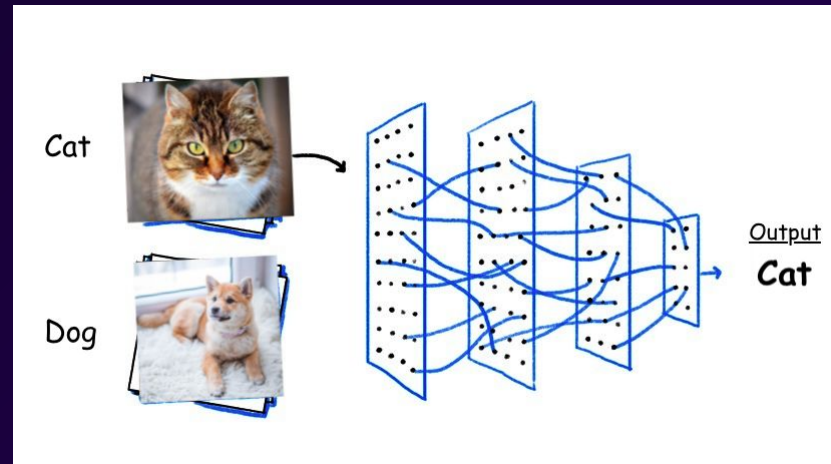
0 2 15 00 01 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 4 01357236526 255 177 95 61 32 0 0 0 29 0 0
0 0 0 0 16 110 28 238 254 244 243 245 250 255 225 101 0 0
0 14 70 155 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 136 255 228 254 251 264 214 111 116 122 125 238 235 235 49
13217 343 155 0 0 0 0 0 0 0 0 0 0 0 132 238 255 235 49
14 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 14 246 244 255 255 25 11 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 81 246 244 255 255 25 11 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 87 252 250 242 215 0 0 0 1 121 252 254 244 6 0 0 0
0 13 113 255 255 245 255 187 181 248 252 242 106 36 0 19
0 1 8 517 251 255 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 4 0 18 251 254 254 254 253 255 120 11 0 0 0
0 0 0 0 9 7 255 255 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 226 252 246 241 250 11 0 0 0 24 113 255 245 255 194 0
0 111 255 252 245 158 24 0 0 0 0 6 39 2525 2520 51 0
0 2124 255 177 7 11 0 0 0 0 2 6 2525 255 252 51 0
0 215 255 255 101 9 0 0 0 3 13 382 245 254 63 0
0 107 251 255 0 0 0 0 0 0 0 0 11 17 17 17 17 17 17 17
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 213 255 255 255 255 255 255 248 148 118 11 0 0
0 0 0 0 0 1 0 5213 329 255 252 147 3 0 0 0 0 1
0 0 5 5 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

```

The process

- Two things are given to a computer, an image and a label. The label tells the computer what is in the picture.
- The machine will convert the image to numbers and do some calculations. Based on the result of these calculations, it will guess what the image contains
- If the machine is correct, it is encouraged to keep on doing the same calculations
- If it's wrong, a slight correction is done.
- This is repeated a looooooot of times, until the machine is corrected and makes accurate guesses

Data	Label
	A cat
	Not a cat
	A cat



For example,

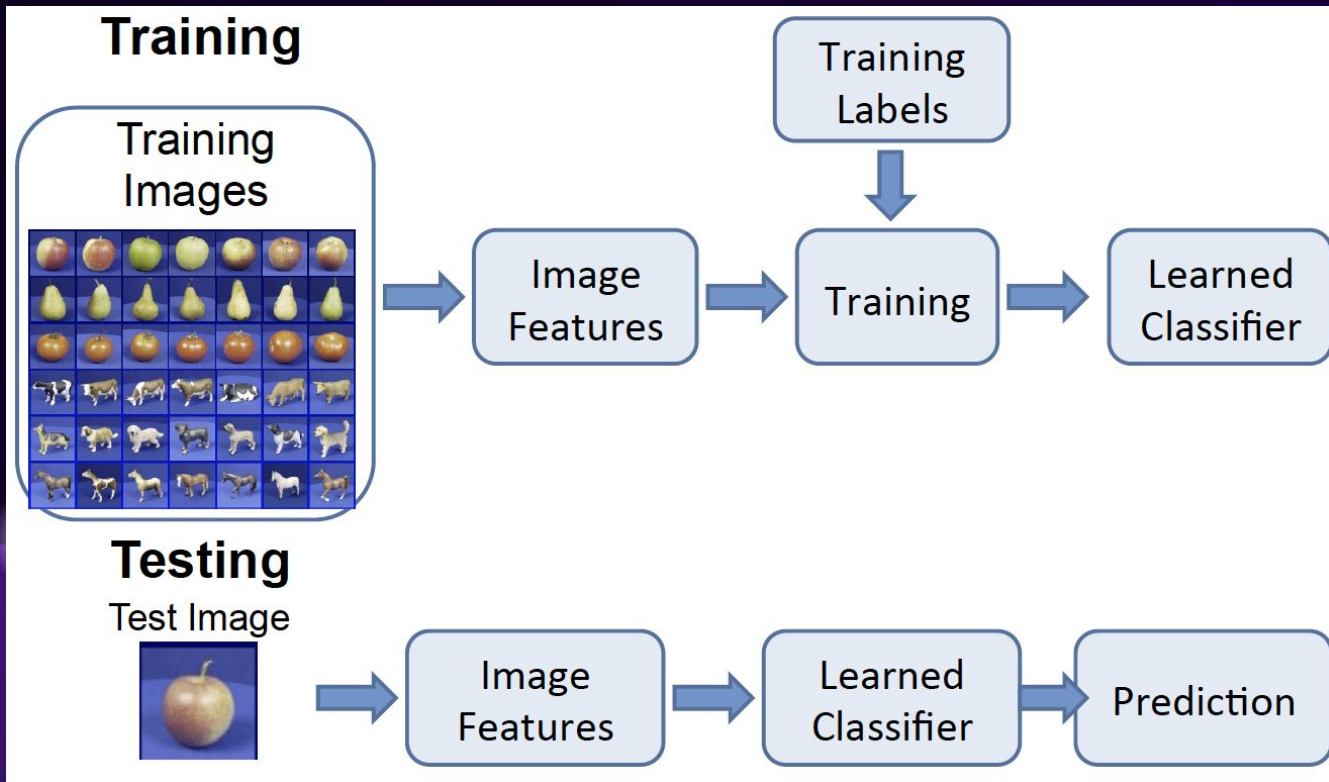


Image features:

converting the image to numbers that the computer understands.

Training labels: the indication of the objects in the image

Training: The machine compares prediction with real label and adjusts

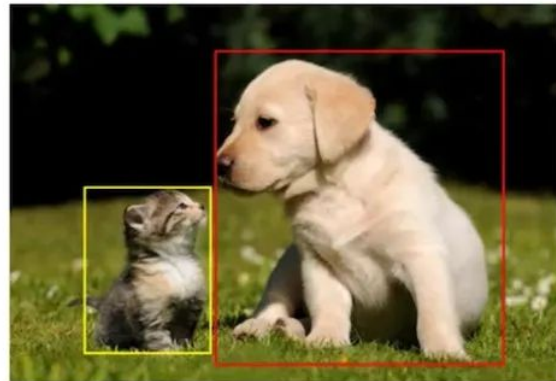
Learned classifier: a model that knows how to calculate!

Is this a dog?



Image Classification

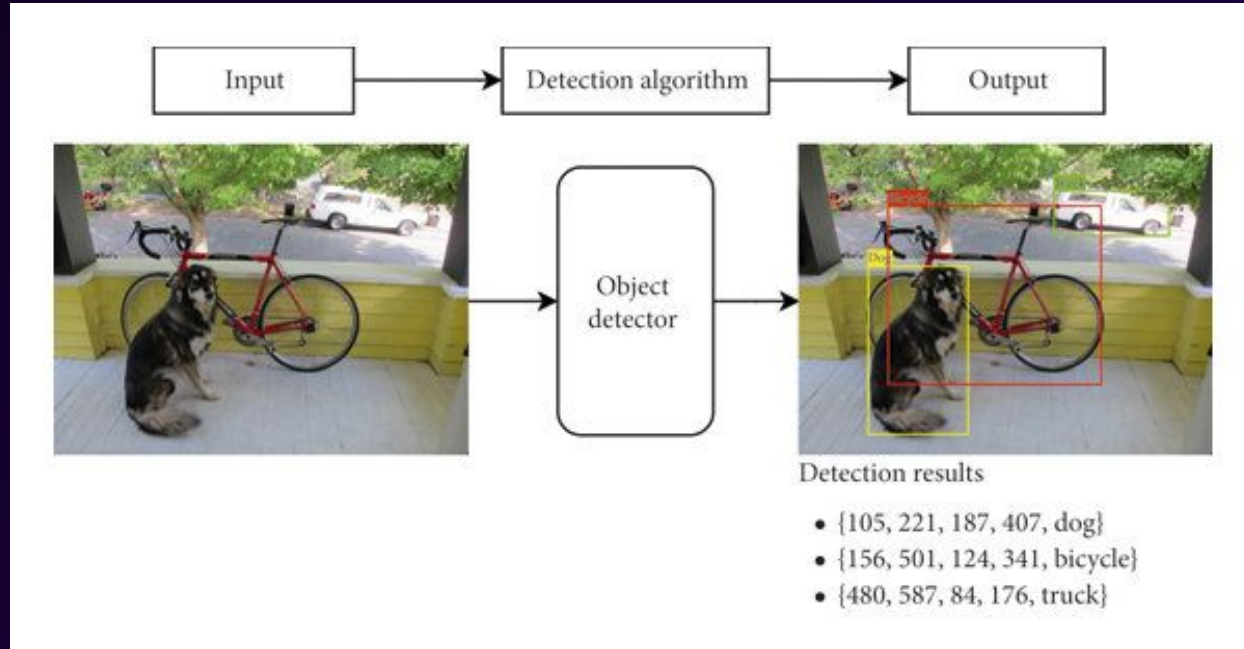
What is there in image
and where?



Object Detection

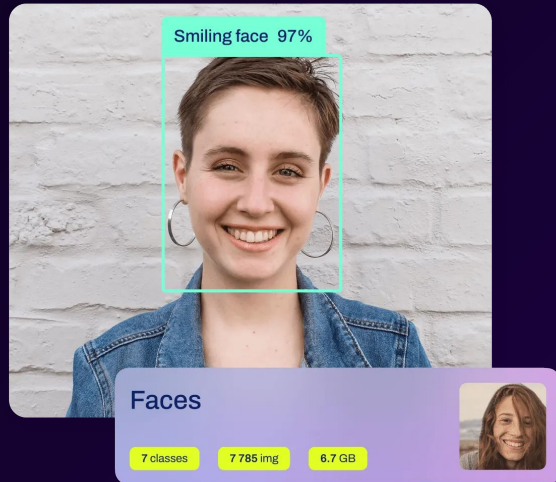
Object Detection...

The training works pretty much the same, but the algorithm has a few more things to calculate:



YOLOv8??

- VERY famous machine learning algorithm
- Intended to be used in REAL-TIME by doing the important calculations in a single step
- Supported by standard hardware (nothing too fancy needed)

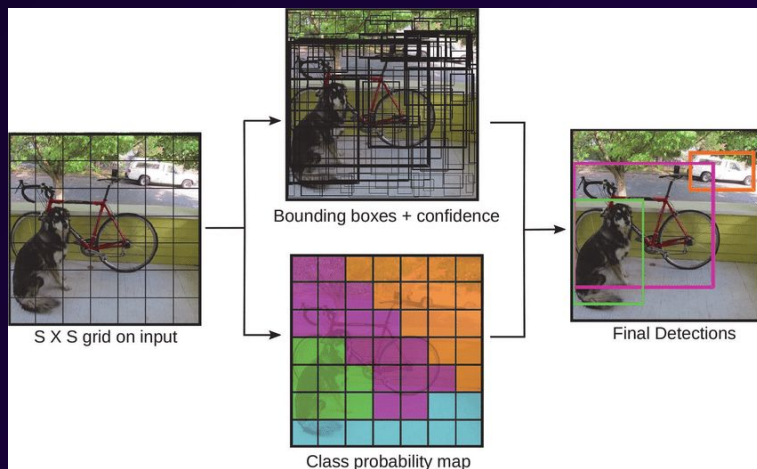


How does it work?

NOTE: this is an overly simplified explanation.

The picture is divided into cells. For each cell, YOLOv8 guesses where objects are present (bounding boxes) and also guesses what each object is (class probabilities)

Then, YOLOv8 deletes the overlapping bounding boxes and selects only the most likely one for each object in the picture.



Let us get our hands dirty with
Yolov8! 🧐🧐

STEP-BY-STEP SET up Environment

1. Run:

```
git clone https://github.com/HKUGenAI/CV\_WorkShop
```

2. Setup Python environment:

```
conda create --name detectionworkshop python=3.12  
conda activate detectionworkshop
```

3. In requirements_yolo.txt:

```
pip install -r requirements_yolo
```

4. Our code only works on torch==2.2.1, so go to

<https://pytorch.org/get-started/previous-versions/>

and install v2.2.1 of torch, torchvision, torchaudio

What is this formula?

$$W = mg$$

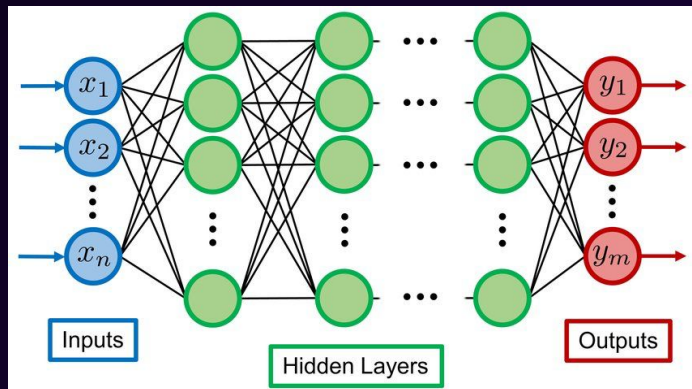
$$P_1 + \frac{1}{2}\rho v_1^2 + \rho gh_1 = P_2 + \frac{1}{2}\rho v_2^2 + \rho gh_2$$

In machine learning, we assume that every pattern in data can be approximated by mathematical functions

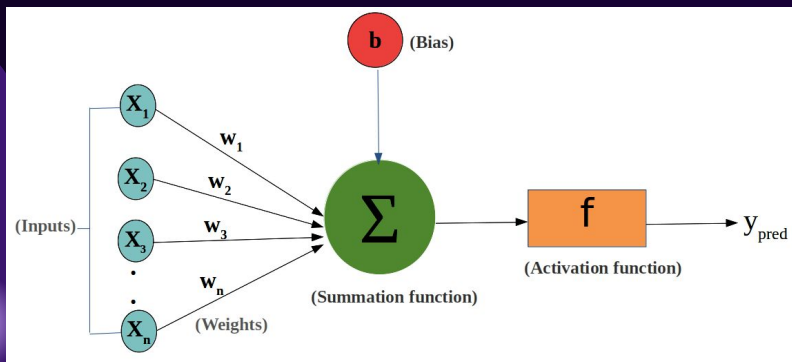
$$\hat{y} = f(w_1x_1 + w_2x_2 + w_3x_3 + \dots + w_{10,000}x_{10,000} + b)$$

$$\begin{aligned} \mathcal{L}_{SM} = & -\frac{1}{2}\partial_\mu g_\nu^\mu \partial_\mu g_\nu^\mu - g_\nu f^{abc} \partial_\mu g_\nu^\mu g_\mu^b g_\mu^c - \frac{1}{2}g_\nu^2 f^{abc} f^{ade} g_\mu^b g_\mu^c g_\mu^d g_\mu^e - \partial_\mu W_\nu^\mu \partial_\mu W_\nu^\mu - \\ & M^2 W_\mu^\mu W_\mu^\mu - \frac{1}{2}\partial_\mu Z_\nu^\mu \partial_\mu Z_\nu^\mu - \frac{1}{2M^2} Z_\nu^\mu Z_\nu^\mu Z_\nu^\mu - \frac{1}{2}\partial_\mu A_\nu \partial_\mu A_\nu - ig_{cw} (\partial_\mu W_\nu^\mu) (W_\nu^\mu W_\nu^\mu - \\ & W_\nu^\mu W_\nu^\mu) - Z_\nu^\mu (W_\nu^\mu \partial_\mu W_\nu^\mu - W_\nu^\mu \partial_\mu W_\nu^\mu) + Z_\nu^\mu (W_\nu^\mu \partial_\mu W_\nu^\mu - W_\nu^\mu \partial_\mu W_\nu^\mu) - \\ & ig_{sw} (\partial_\mu A_\nu (W_\nu^\mu W_\nu^\mu - W_\nu^\mu W_\nu^\mu) - A_\nu (W_\nu^\mu \partial_\mu W_\nu^\mu - W_\nu^\mu \partial_\mu W_\nu^\mu) + A_\nu (W_\nu^\mu \partial_\mu W_\nu^\mu - \\ & W_\nu^\mu \partial_\mu W_\nu^\mu)) - \frac{1}{2}g^2 W_\nu^\mu W_\nu^\mu W_\nu^\mu + \frac{1}{2}g^2 W_\nu^\mu W_\nu^\mu W_\nu^\mu + g^2 c_w^2 (Z_\nu^\mu Z_\nu^\mu Z_\nu^\mu - \\ & Z_\nu^\mu Z_\nu^\mu W_\nu^\mu) + g^2 s_w^2 (A_\nu A_\nu W_\nu^\mu - A_\nu W_\nu^\mu W_\nu^\mu) + g^2 s_w c_w (A_\nu Z_\nu^\mu (W_\nu^\mu W_\nu^\mu - \\ & W_\nu^\mu W_\nu^\mu) - \frac{1}{2}A_\nu Z_\nu^\mu W_\nu^\mu) - \frac{1}{2}\partial_\mu H \partial_\mu H - 2M^2 \phi_\mu H - \partial_\mu \phi^\mu \partial_\mu \phi^\mu - \frac{1}{2}\partial_\mu \phi^\mu \partial_\mu \phi^\mu - \\ & \beta_\alpha \left(\frac{2\alpha_\alpha}{\Lambda^2} H + \frac{2\alpha_\alpha}{\Lambda^2} H + (H^2 + \phi^\mu \phi^\mu + 2\phi^\mu \phi^\mu) \right) + \frac{2\alpha_\alpha}{\Lambda^2} \alpha_\alpha - \\ & \frac{1}{2}g^2 \alpha_\alpha (H^4 + (\phi^\mu)^4 + 4(\phi^\mu)^2 \phi^\mu \phi^\mu + 4(H^2 \phi^\mu \phi^\mu + 2(\phi^\mu)^2 H^2) - \\ & g\alpha_\alpha M (H^3 + H\phi^\mu \phi^\mu + 2H\phi^\mu \phi^\mu) - \\ & gMW_\nu^\mu W_\nu^\mu H - \frac{1}{2}g\frac{\alpha_\alpha}{\Lambda^2} Z_\nu^\mu Z_\nu^\mu H - \\ & \frac{1}{2}ig (W_\nu^\mu (\phi^\mu \partial_\mu \phi^\mu - \phi^\mu \partial_\mu \phi^\mu) - W_\nu^\mu (\phi^\mu \partial_\mu \phi^\mu - \phi^\mu \partial_\mu \phi^\mu)) + \\ & \frac{1}{2}g (W_\nu^\mu (H\partial_\mu \phi^\mu - \phi^\mu \partial_\mu H) + W_\nu^\mu (H\partial_\mu \phi^\mu - \phi^\mu \partial_\mu H)) + \frac{1}{2}g\frac{1}{\Lambda^2} (Z_\nu^\mu (H\partial_\mu \phi^\mu - \phi^\mu \partial_\mu H) + \\ & M (\frac{1}{\Lambda^2} Z_\nu^\mu \partial_\mu \phi^\mu + W_\nu^\mu \partial_\mu \phi^\mu + W_\nu^\mu \partial_\mu \phi^\mu) - ig\frac{\alpha_\alpha}{\Lambda^2} M Z_\nu^\mu (W_\nu^\mu \phi^\mu - W_\nu^\mu \phi^\mu) + ig_{sw} M A_\nu (W_\nu^\mu \phi^\mu - \\ & W_\nu^\mu \phi^\mu) - ig\frac{1+2c_w^2}{\Lambda^2} Z_\nu^\mu (\phi^\mu \partial_\mu \phi^\mu - \phi^\mu \partial_\mu \phi^\mu) + ig_{sw} A_\nu (\phi^\mu \partial_\mu \phi^\mu - \phi^\mu \partial_\mu \phi^\mu) - \\ & \frac{1}{2}g^2 W_\nu^\mu W_\nu^\mu (H^2 + (\phi^\mu)^2 + 2\phi^\mu \phi^\mu) - \frac{1}{2}g^2 \frac{1}{\Lambda^2} Z_\nu^\mu Z_\nu^\mu (H^2 + (\phi^\mu)^2 + 2(2s_w^2 - 1)^2 \phi^\mu \phi^\mu) - \\ & \frac{1}{2}g^2 \frac{1}{\Lambda^2} Z_\nu^\mu Z_\nu^\mu \phi^\mu (W_\nu^\mu \phi^\mu + W_\nu^\mu \phi^\mu) - \frac{1}{2}ig^2 \frac{1}{\Lambda^2} Z_\nu^\mu Z_\nu^\mu H (W_\nu^\mu \phi^\mu - W_\nu^\mu \phi^\mu) + \frac{1}{2}g^2 s_w A_\nu \phi^\mu (W_\nu^\mu \phi^\mu + \\ & W_\nu^\mu \phi^\mu) + \frac{1}{2}ig^2 s_w A_\nu H (W_\nu^\mu \phi^\mu - W_\nu^\mu \phi^\mu) - g^2 \frac{1}{\Lambda^2} (2c_w^2 - 1) Z_\nu^\mu A_\nu \phi^\mu \phi^\mu - \\ & g^2 s_w^2 A_\nu A_\nu \phi^\mu \phi^\mu + \frac{1}{2}ig_\lambda \lambda_\nu^2 (\eta^\mu \gamma^\mu \eta^\mu) g_\nu^2 - e^2 (\gamma\partial + m_\nu^2) e^\mu - \nu^\mu (\gamma\partial + m_\nu^2) \nu^\mu - u_\nu^\mu (\gamma\partial + \\ & m_\nu^2) u_\nu^\mu - d_\nu^\mu (\gamma\partial + m_\nu^2) d_\nu^\mu + ig_{sw} A_\nu (-e^\mu \gamma^\mu e^\mu) + \frac{1}{2}(u_\nu^\mu \gamma^\mu u_\nu^\mu) - \frac{1}{2}(d_\nu^\mu \gamma^\mu d_\nu^\mu) + \\ & \frac{ig}{\Lambda^2} Z_\nu^\mu ((\nu^\mu \gamma^\mu (1 + \gamma^5) \nu^\mu) + (e^\mu \gamma^\mu (4s_w^2 - 1 - \gamma^5) e^\mu) + (d_\nu^\mu \gamma^\mu (\frac{1}{3}s_w^2 - 1 - \gamma^5) d_\nu^\mu) + \\ & (u_\nu^\mu \gamma^\mu (1 - \frac{2}{3}s_w^2 + \gamma^5) u_\nu^\mu)) + \frac{ig}{\Lambda^2} W_\nu^\mu ((\nu^\mu \gamma^\mu (1 + \gamma^5) U^{lq}_{\nu\mu} e^\mu) + (u_\nu^\mu \gamma^\mu (1 + \gamma^5) C_{\nu\mu} d_\nu^\mu)) + \\ & \frac{ig}{\Lambda^2} W_\nu^\mu ((e^\mu U^{lq}_{\nu\mu} \lambda_\nu \gamma^\mu (1 + \gamma^5) \nu^\mu) + (d_\nu^\mu C_{\nu\mu} \gamma^\mu (1 + \gamma^5) u_\nu^\mu)) + \\ & \frac{ig}{2M\sqrt{2}} \phi^\mu (-m_\nu^2 (\nu^\mu U^{lq}_{\nu\mu} \lambda_\nu (1 - \gamma^5) e^\mu) + m_\nu^2 (\nu^\mu U^{lq}_{\nu\mu} \lambda_\nu (1 + \gamma^5) e^\mu) + \\ & \frac{ig}{2M\sqrt{2}} \phi^\mu (-m_\nu^2 (e^\mu U^{lq}_{\nu\mu} \lambda_\nu (1 + \gamma^5) \nu^\mu) - m_\nu^2 (e^\mu U^{lq}_{\nu\mu} \lambda_\nu (1 - \gamma^5) \nu^\mu) - \frac{g}{2} \frac{m_\nu^2}{M} H (\nu^\mu \nu^\mu) - \\ & \frac{g}{2} \frac{m_\nu^2}{M} H (e^\mu e^\mu) + \frac{ig}{2} \frac{m_\nu^2}{M} \phi^\mu (\nu^\mu \gamma^\mu \nu^\mu) - \frac{ig}{2} \frac{m_\nu^2}{M} \phi^\mu (e^\mu \gamma^\mu e^\mu) - \frac{1}{2} \bar{\nu}_\lambda M_\lambda^\mu (1 - \gamma_5) \bar{\nu}_\lambda - \\ & \frac{1}{2} \bar{\nu}_\lambda M_\lambda^\mu (1 - \gamma_5) \bar{\nu}_\lambda + \frac{ig}{2M\sqrt{2}} \phi^\mu (-m_\nu^2 (\bar{u}_\nu^\mu C_{\nu\mu} (1 - \gamma^5) d_\nu^\mu) + m_\nu^2 (\bar{u}_\nu^\mu C_{\nu\mu} (1 + \gamma^5) d_\nu^\mu) + \\ & \frac{ig}{2M\sqrt{2}} \phi^\mu (-m_\nu^2 (\bar{d}_\nu^\mu C_{\nu\mu}^\dagger (1 + \gamma^5) u_\nu^\mu) - m_\nu^2 (\bar{d}_\nu^\mu C_{\nu\mu}^\dagger (1 - \gamma^5) u_\nu^\mu) - \frac{g}{2} \frac{m_\nu^2}{M} H (\bar{u}_\nu^\mu \bar{u}_\nu^\mu) - \\ & \frac{g}{2} \frac{m_\nu^2}{M} H (\bar{d}_\nu^\mu \bar{d}_\nu^\mu) + \frac{ig}{2} \frac{m_\nu^2}{M} \phi^\mu (\bar{u}_\nu^\mu \gamma^\mu \bar{u}_\nu^\mu) - \frac{ig}{2} \frac{m_\nu^2}{M} \phi^\mu (\bar{d}_\nu^\mu \gamma^\mu \bar{d}_\nu^\mu) + G^\mu \bar{\theta}^\mu G^\mu + g_\nu f^{abc} \partial_\mu G^\mu G^\mu g_\nu^\mu + \\ & \bar{X}^\mu (\bar{\theta}^\mu - M^2) X^\mu + \bar{X}^\mu (\bar{\theta}^\mu - M^2) X^\mu + \bar{X}^\mu (\bar{\theta}^\mu - \frac{M^2}{2}) X^\mu + \bar{Y} \bar{\theta}^\mu Y + ig_{cw} W_\nu^\mu (\partial_\mu \bar{X}^\mu X^\mu - \\ & \partial_\mu \bar{X}^\mu X^\mu) + ig_{sw} W_\nu^\mu (\partial_\mu \bar{Y} X^\mu - \partial_\mu \bar{Y} X^\mu) + ig_{cw} W_\nu^\mu (\partial_\mu \bar{X}^\mu X^\mu - \\ & \partial_\mu \bar{X}^\mu X^\mu) + ig_{sw} W_\nu^\mu (\partial_\mu \bar{Y} X^\mu - \partial_\mu \bar{Y} X^\mu) + ig_{cw} Z_\nu^\mu (\partial_\mu \bar{X}^\mu X^\mu - \\ & \partial_\mu \bar{X}^\mu X^\mu) + ig_{sw} A_\nu (\partial_\mu \bar{X}^\mu X^\mu - \\ & \partial_\mu \bar{X}^\mu X^\mu) - \frac{1}{2}igM (\bar{X}^\mu X^\mu + \bar{X}^\mu X^\mu - H + \frac{1}{\Lambda^2} \bar{X}^\mu X^\mu H) + \frac{1+2c_w^2}{2\Lambda^2} igM (\bar{X}^\mu X^\mu \phi^\mu - \bar{X}^\mu X^\mu \phi^\mu) + \\ & \frac{1}{2\Lambda^2} igM (\bar{X}^\mu X^\mu \phi^\mu - \bar{X}^\mu X^\mu \phi^\mu) + igM s_w (\bar{X}^\mu X^\mu \phi^\mu - \bar{X}^\mu X^\mu \phi^\mu) + \\ & \frac{1}{2}igM (\bar{X}^\mu X^\mu \phi^\mu - \bar{X}^\mu X^\mu \phi^\mu) . \end{aligned}$$

What's going on inside the machine?



- We train the machine with thousands of samples so that the machine can recognize in the data.
- During training, the machine adjusts weights and biases to improve accuracy.
- In YOLOv8, the image is passed as an input through layers of the network that do some calculation to help determine the image and come up with final prediction.

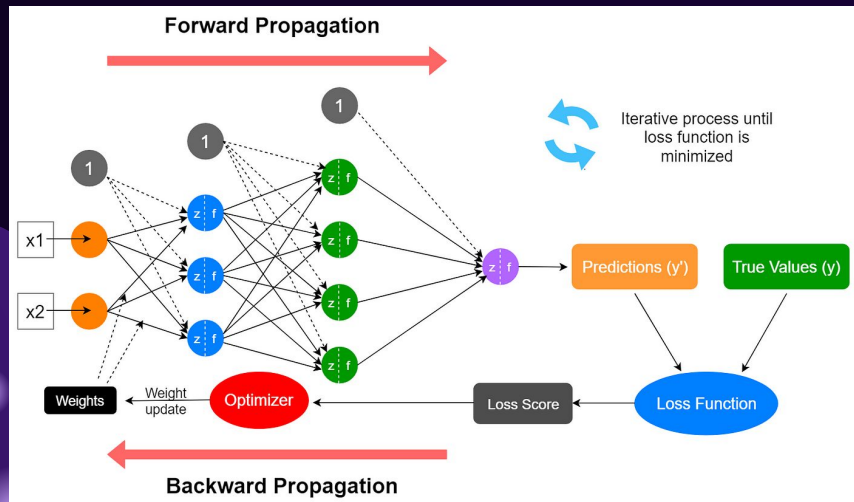


How does the network get better?

Prediction (Forward Pass) → **Compare to real answer** (Loss Function)
→ **Learn** (Backpropagation)



<https://xnought.github.io/backprop-explainer/>



- **Loss Function** tells how wrong the prediction (output) is.
- **Backpropagation** (Backward Pass) calculates how much each weight and bias contributes to final error.
- Then, those parameters are updated by optimizer.
- We train the machine until the loss reaching a desirable low value.

Evaluation metrics

- When the prediction and actual label do not match, it is FALSE, else TRUE
- When the class is predicted, it is POSITIVE, else NEGATIVE

		Predicted Label	
		Positive	Negative
Actual Label	Positive	TRUE POSITIVE TP	FALSE NEGATIVE FN
	Negative	FALSE POSITIVE FP	TRUE NEGATIVE TN

F1 Score

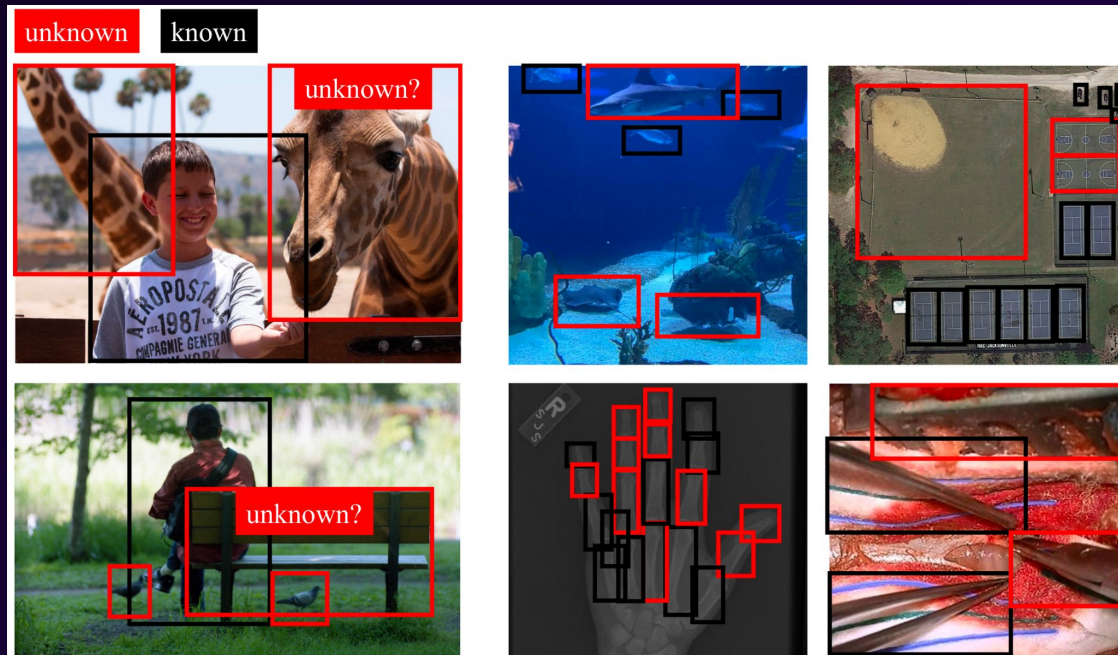
- Precision = $TP / (TP + FP)$
 - Focus on predicted classes being correct.
 - **Correctness - do not make wrong prediction.**
- Recall = $TP / (TP + FN)$
 - Focus on actual classes being predicted.
 - **Completeness - do not miss out.**
- F1 score = $(2 \times \text{Precision} \times \text{Recall}) / (\text{Precision} + \text{Recall})$
 - Balance between precision and recall

		Predicted Label	
		Positive	Negative
Actual Label	Positive	TRUE POSITIVE TP	FALSE NEGATIVE FN
	Negative	FALSE POSITIVE FP	TRUE NEGATIVE TN



The only things Yolo can detect:

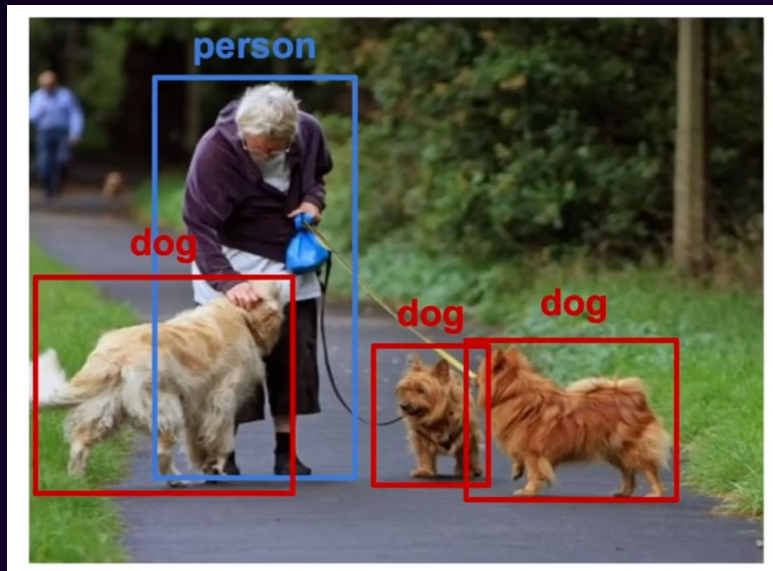
{0: 'person', 1: 'bicycle', 2: 'car', 3: 'motorcycle', 4: 'airplane', 5: 'bus', 6: 'train', 7: 'truck', 8: 'boat', 9: 'traffic light', 10: 'fire hydrant', 11: 'stop sign', 12: 'parking meter', 13: 'bench', 14: 'bird', 15: 'cat', 16: 'dog', 17: 'horse', 18: 'sheep', 19: 'cow', 20: 'elephant', 21: 'bear', 22: 'zebra', 23: 'giraffe', 24: 'backpack', 25: 'umbrella', 26: 'handbag', 27: 'tie', 28: 'suitcase', 29: 'frisbee', 30: 'skis', 31: 'snowboard', 32: 'sports ball', 33: 'kite', 34: 'baseball bat', 35: 'baseball glove', 36: 'skateboard', 37: 'surfboard', 38: 'tennis racket', 39: 'bottle', 40: 'wine glass', 41: 'cup', 42: 'fork', 43: 'knife', 44: 'spoon', 45: 'bowl', 46: 'banana', 47: 'apple', 48: 'sandwich', 49: 'orange', 50: 'broccoli', 51: 'carrot', 52: 'hot dog', 53: 'pizza', 54: 'donut', 55: 'cake', 56: 'chair', 57: 'couch', 58: 'potted plant', 59: 'bed', 60: 'dining table', 61: 'toilet', 62: 'tv', 63: 'laptop', 64: 'mouse', 65: 'remote', 66: 'keyboard', 67: 'cell phone', 68: 'microwave', 69: 'oven', 70: 'toaster', 71: 'sink', 72: 'refrigerator', 73: 'book', 74: 'clock', 75: 'vase', 76: 'scissors', 77: 'teddy bear', 78: 'hair drier', 79: 'toothbrush'}



If you want to detect something else, **YOU MUST TRAIN THE MODEL!!!**

This is called **“closed-set object detection”**...

Closed-Set Object Detection



Find me the objects that you
HAVE BEEN TRAINED

vs

Opened-Set Object Detection



Find me the rightmost dog
(just prompt keywords)

CLOSED-SET



etc.

OPEN-SET

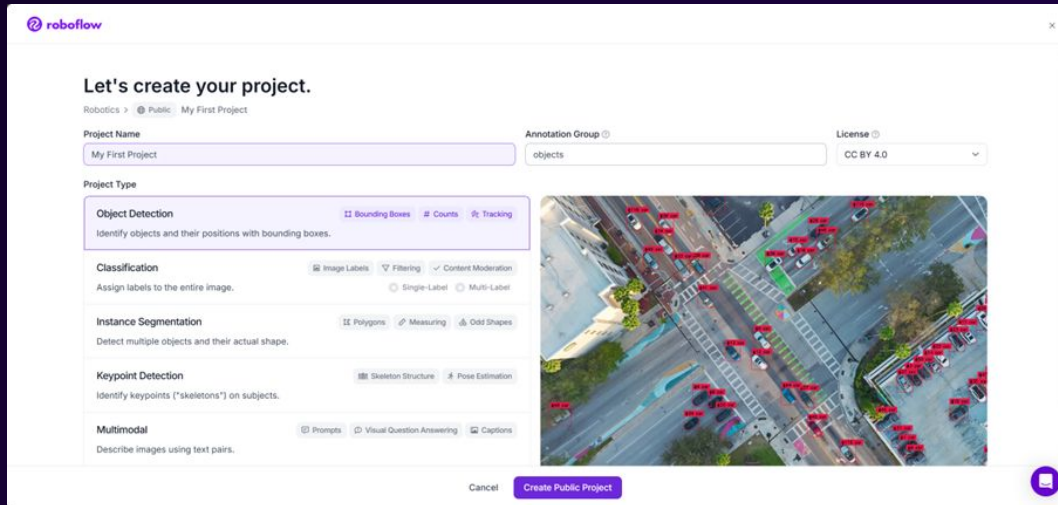


Custom Your Own Dataset

- Closed-Set for specific task!
- Create your Own Label for objects

Custom Your Own Dataset

- Platform – Roboflow
- Procedure – Startup



The image shows the Roboflow web interface for creating a new project. The header includes the Roboflow logo and a close button. The main heading is "Let's create your project." Below this, the breadcrumb "Robotics > Public > My First Project" is visible. The form is divided into several sections: "Project Name" with a text input field containing "My First Project"; "Annotation Group" with a dropdown menu set to "objects"; and "License" with a dropdown menu set to "CC BY 4.0". The "Project Type" section is expanded, showing options for "Object Detection" (with sub-options for Bounding Boxes, Counts, and Tracking), "Classification" (with sub-options for Image Labels, Filtering, and Content Moderation), "Instance Segmentation" (with sub-options for Polygons, Measuring, and Odd Shapes), "Keypoint Detection" (with sub-options for Skeleton Structure and Pose Estimation), and "Multimodal" (with sub-options for Prompts, Visual Question Answering, and Captions). A preview image on the right shows a street scene with various objects labeled with red bounding boxes. At the bottom, there are "Cancel" and "Create Public Project" buttons.

Let's create your project.

Robotics > Public > My First Project

Project Name: My First Project

Annotation Group: objects

License: CC BY 4.0

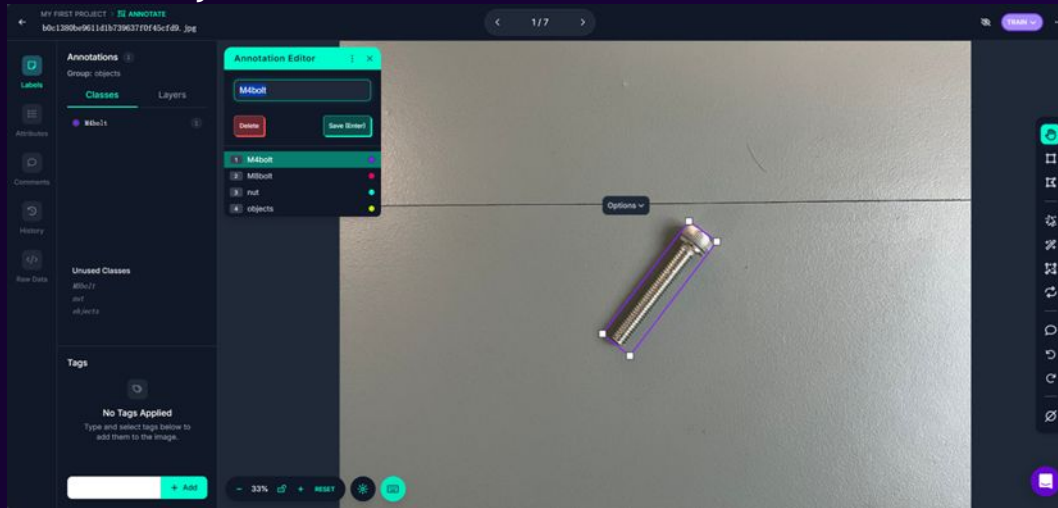
Project Type:

- Object Detection** (Identify objects and their positions with bounding boxes.)
 - Bounding Boxes
 - Counts
 - Tracking
- Classification** (Assign labels to the entire image.)
 - Image Labels
 - Filtering
 - Content Moderation
 - Single-Label
 - Multi-Label
- Instance Segmentation** (Detect multiple objects and their actual shape.)
 - Polygons
 - Measuring
 - Odd Shapes
- Keypoint Detection** (Identify keypoints ("skeletons") on subjects.)
 - Skeleton Structure
 - Pose Estimation
- Multimodal** (Describe images using text pairs.)
 - Prompts
 - Visual Question Answering
 - Captions

Cancel Create Public Project

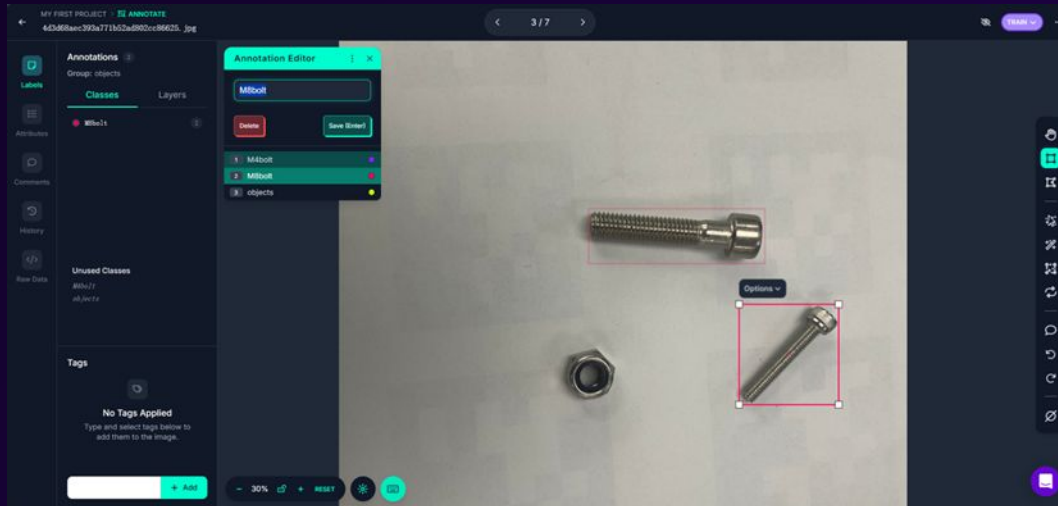
Custom Your Own Dataset

- Procedure – Label
- Single / Multi objects



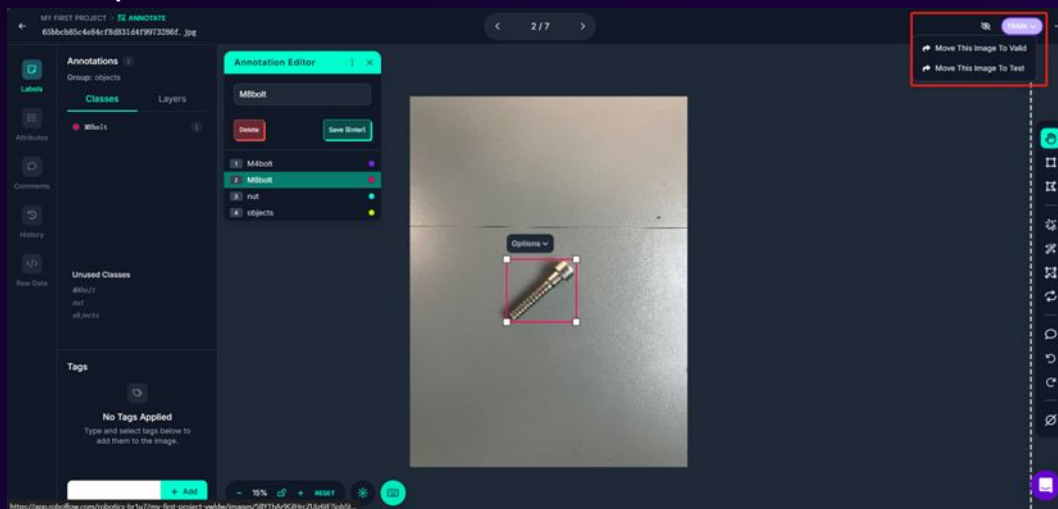
Custom Your Own Dataset

- Procedure – Label
- Bounding Box Shape



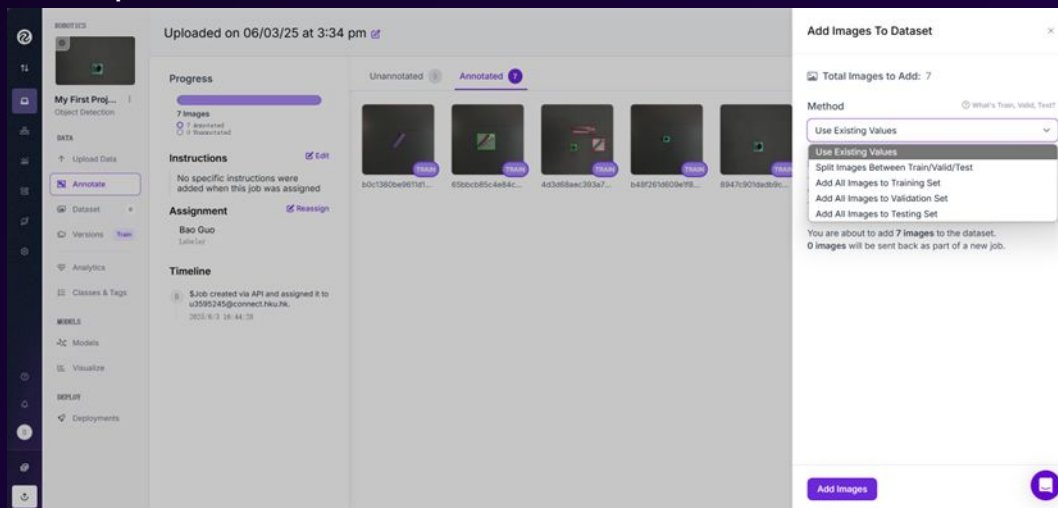
Custom Your Own Dataset

- Procedure – Label
- Data for Purpose



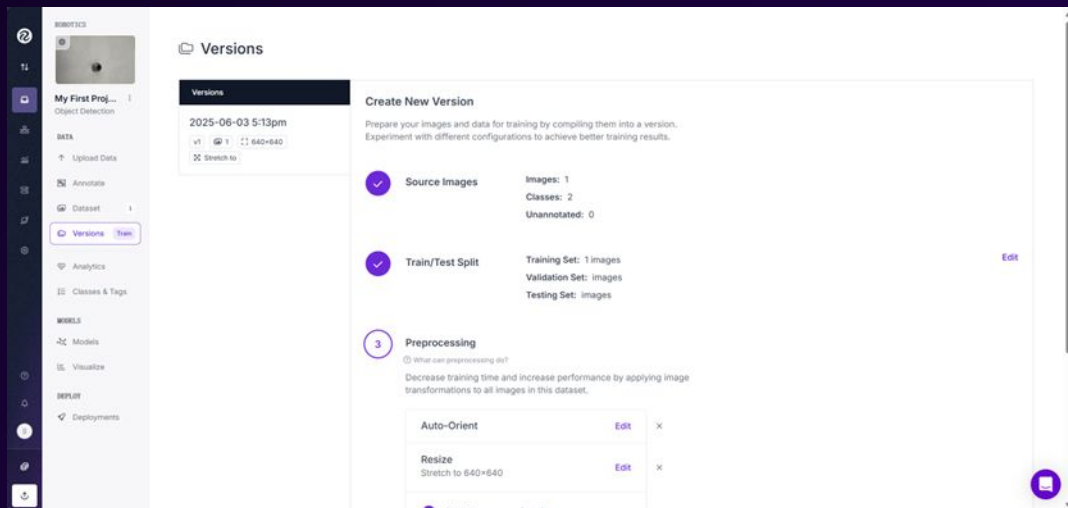
Custom Your Own Dataset

- Procedure – Add to Dataset
- Data for Purpose



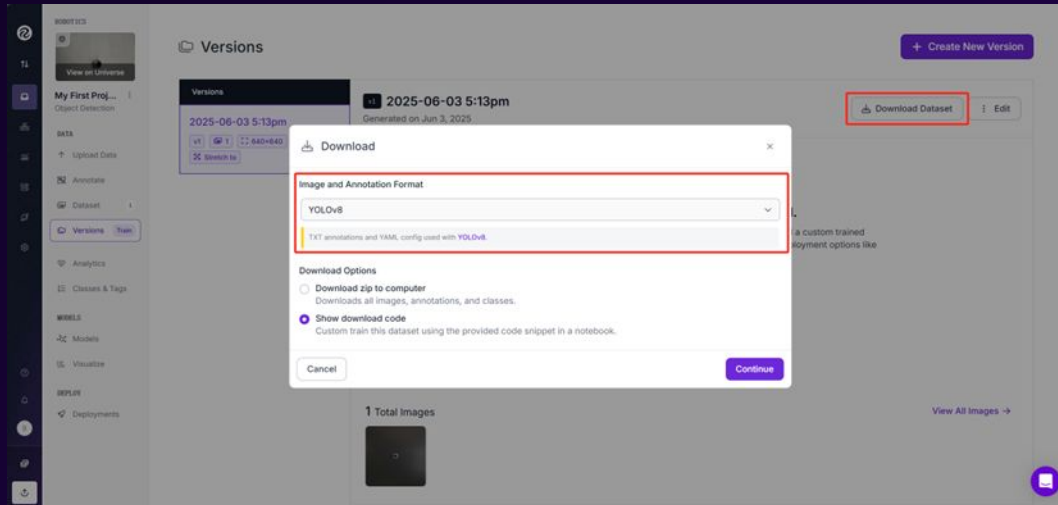
Custom Your Own Dataset

- Procedure – Processing
- Preprocess & Augmentation



Custom Your Own Dataset

- Almost Done!
- Select and download your dataset



Custom Your Own Dataset

- Almost Done!
- Folder Structure

```
May not be exactly the same!

root\
|_
|_ data.yaml
|_ README.dataset.txt
|_ README.roboflow.txt
|_ train\
|_ |_ images\
|_ |_ |_ xxx.jpeg (.png, .jpg)
|_ |_ |_ ...
|_ |_ |_ xxx.jpeg
|_ |_ labels\
|_ |_ |_ xxx.jpeg (.png, .jpg)
|_ |_ |_ ...
|_ |_ |_ xxx.jpeg
|_ val\
|_ |_ ...
|_ test\
|_ |_ ...
```

Custom Your Own Dataset

- Reference: [Notion Page](#)

Thanks!!

Any questions?

