Mold Object Recognition Project (MORP)

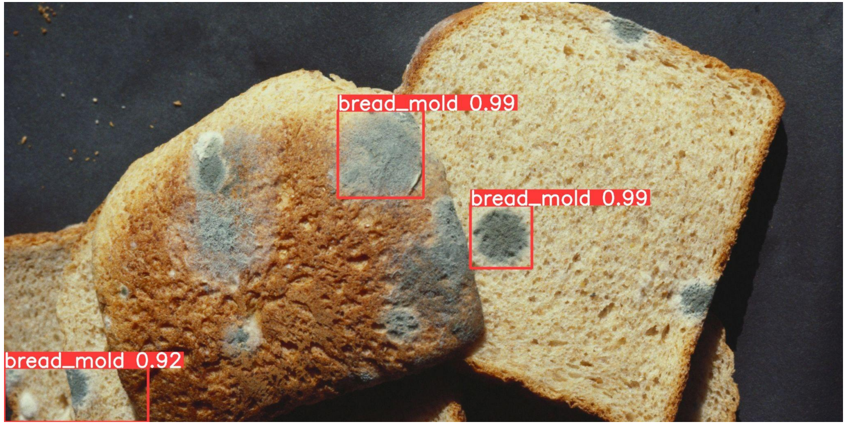
By Collin Bovenschen

Initial Problem Recap

- Mold is prevalent in production lines and distribution centers
- Can cause financial and health issues for consumers & producers

Solution

- Custom object detection model (can recognize mold)
- Transfer learning from YOLOv8 to custom model
- Desktop/Mobile version of detector

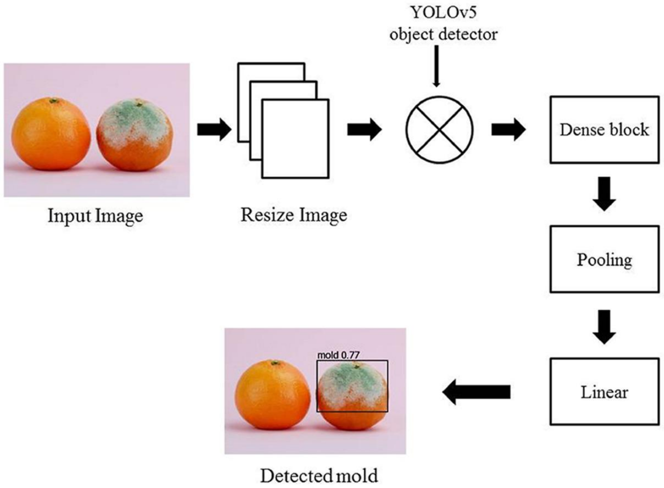


Process

1. Initial research
2. Choosing a dataset
3. Train model
4. Deploy final product

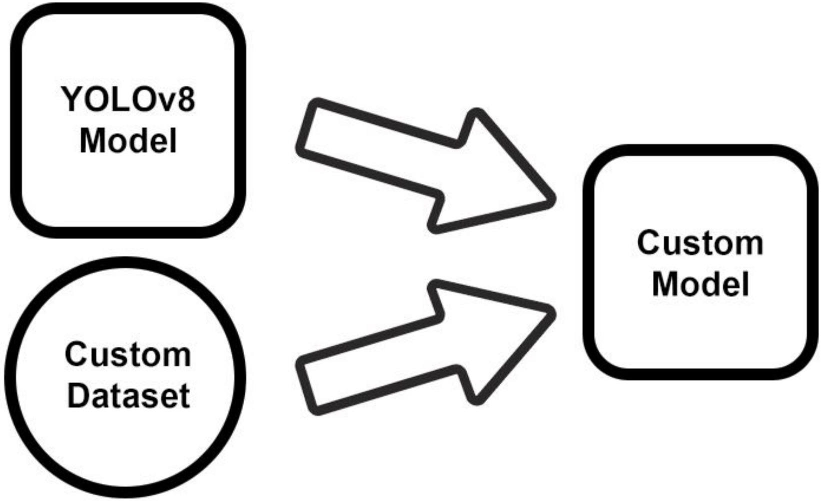
Initial Research

- Provided insight on a proper workflow
- Detailed tools for model training (YOLO and Ultralytics)



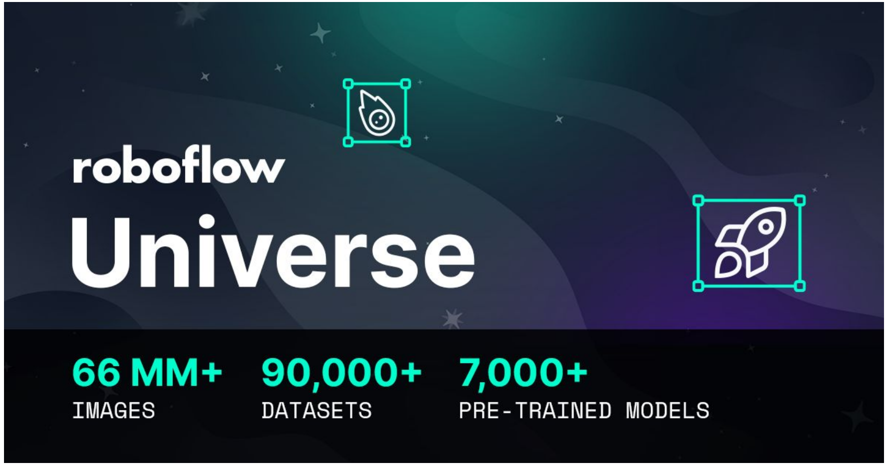
Transfer Learning

- “Transfer” information from pretrained model onto custom dataset
- Creates a smaller, more specific model



Choosing a Dataset

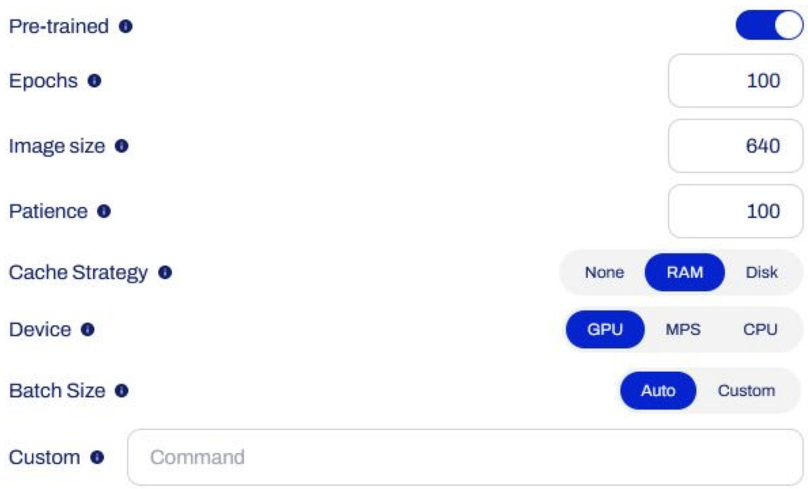
- Tested many from Roboflow Universe
- Used the dataset from the original research



Train Model

- Trained with Ultralytics HUB
- Streamlines training process
- Little coding required
- Many changeable parameters

Train Model (cont.)



Pre-trained ☑

Epochs 100

Image size 640

Patience 100

Cache Strategy None [✓ RAM] Disk

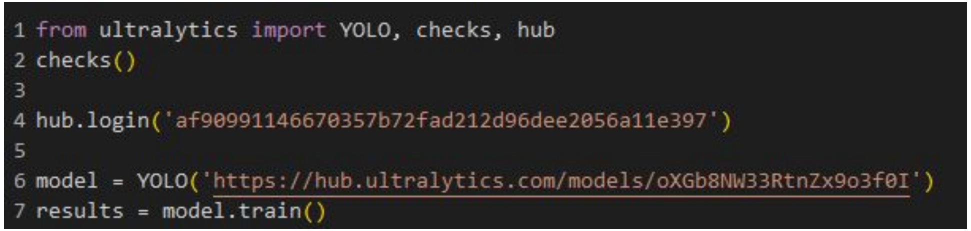
Device [✓ GPU] MPS CPU

Batch Size [✓ Auto] Custom

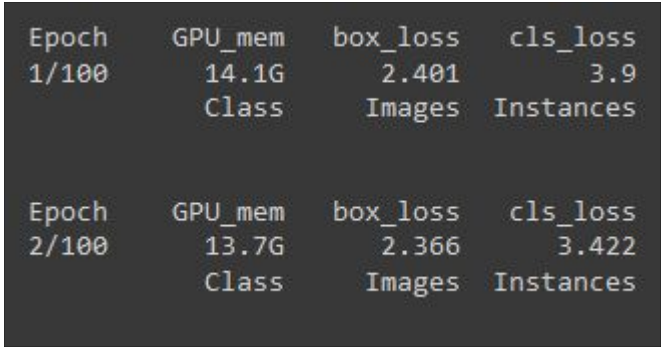
Custom Command

Detect

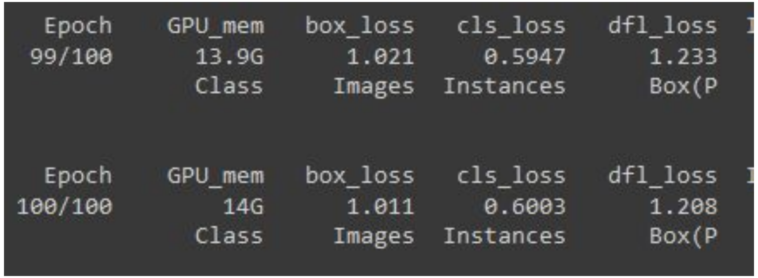
Train Model (cont.)



```
from ultralytics import YOLO, checks, hub
checks()

hub.login('af90991146670357b72fad212d96dee2056a11e397')

model = YOLO('https://hub.ultralytics.com/models/oXGb8NW33RtnZx9o3f0I')
results = model.train()
```



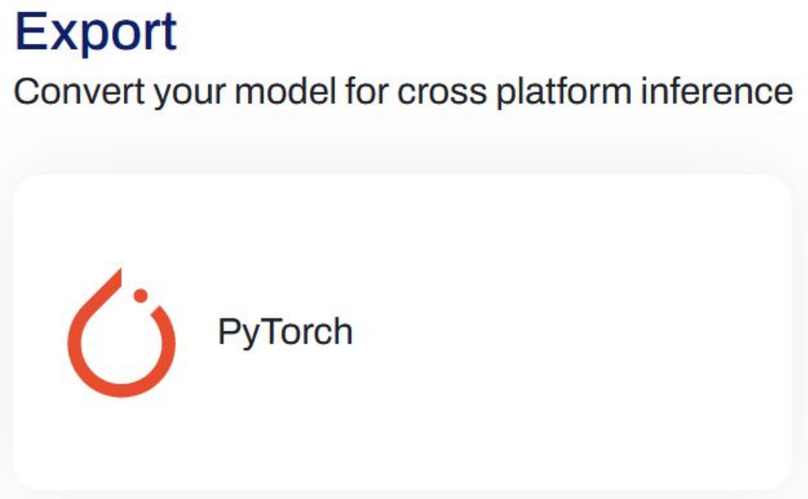
```
 Epoch    GPU_mem   box_loss   cls_loss
 1/100      14.1G      2.401        3.9
            Class     Images  Instances


 Epoch    GPU_mem   box_loss   cls_loss
 2/100      13.7G      2.366      3.422
            Class     Images  Instances
```



```
   Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  I
  99/100      13.9G      1.021     0.5947      1.233
              Class     Images  Instances      Box(P


   Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  I
 100/100        14G      1.011     0.6003      1.208
              Class     Images  Instances      Box(P
```

Train Model (cont.)

- Produces lightweight model made for real time image detection (49.6MB)
- Can be used in custom application

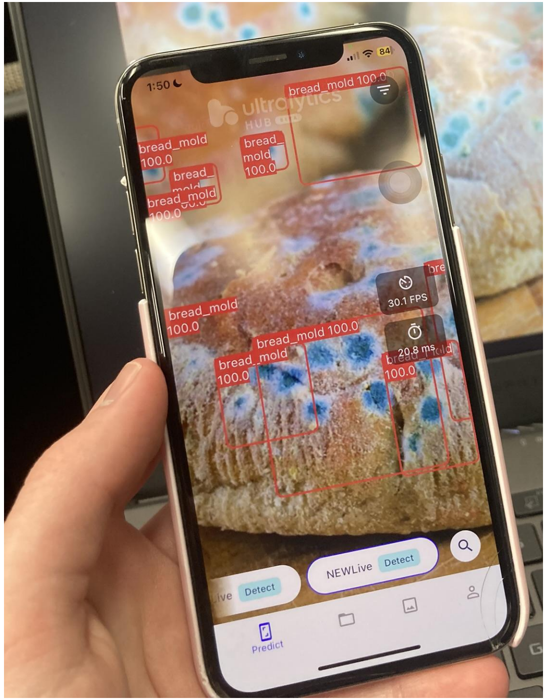


Export

Convert your model for cross platform inference

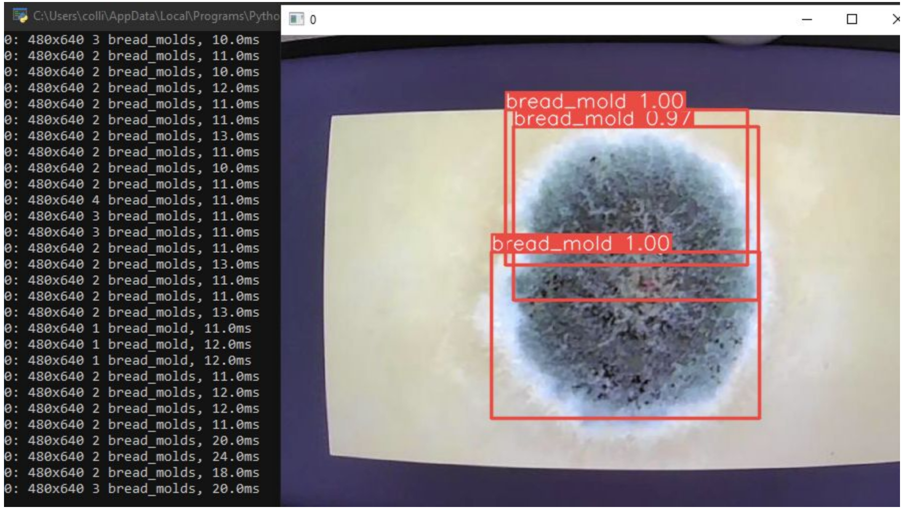
Deploy Final Product

- Model can be loaded through Ultralytics HUB mobile app
- Can run real time using the phone's camera as image data



Deploy Final Product (cont.)

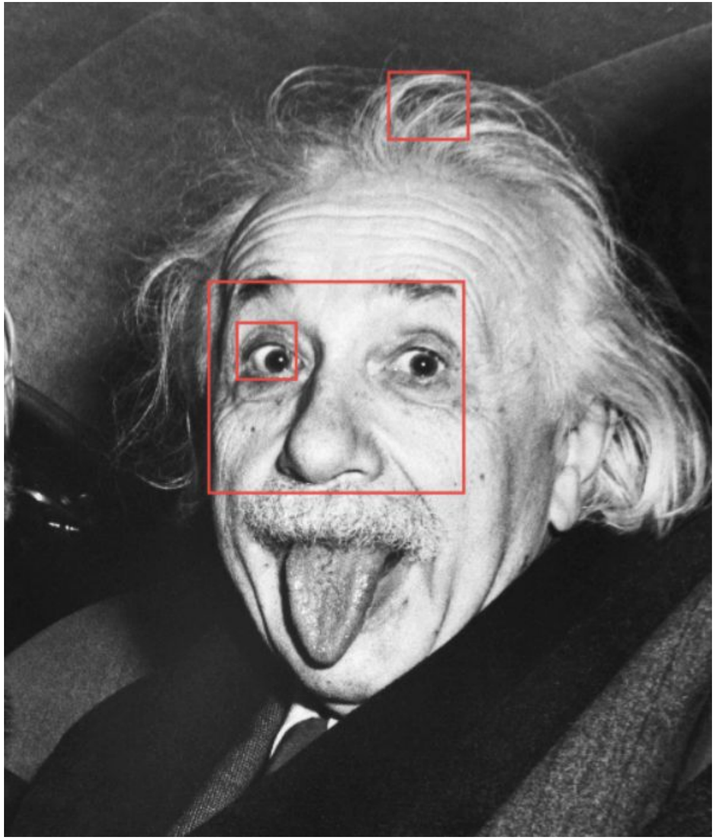
- Custom Python Application for desktop use
- Outputs are saved to a folder
- Can process both images and video



```
0: 480x640 3 bread_molds, 10.0ms
0: 480x640 2 bread_molds, 11.0ms
0: 480x640 2 bread_molds, 10.0ms
0: 480x640 2 bread_molds, 12.0ms
0: 480x640 2 bread_molds, 11.0ms
0: 480x640 2 bread_molds, 11.0ms
0: 480x640 2 bread_molds, 13.0ms
0: 480x640 2 bread_molds, 11.0ms
0: 480x640 2 bread_molds, 10.0ms
0: 480x640 2 bread_molds, 11.0ms
0: 480x640 4 bread_molds, 11.0ms
0: 480x640 3 bread_molds, 11.0ms
0: 480x640 3 bread_molds, 11.0ms
0: 480x640 2 bread_molds, 11.0ms
0: 480x640 2 bread_molds, 13.0ms
0: 480x640 2 bread_molds, 11.0ms
0: 480x640 2 bread_molds, 11.0ms
0: 480x640 2 bread_molds, 13.0ms
0: 480x640 1 bread_mold, 11.0ms
0: 480x640 1 bread_mold, 12.0ms
0: 480x640 1 bread_mold, 12.0ms
0: 480x640 2 bread_molds, 11.0ms
0: 480x640 2 bread_molds, 12.0ms
0: 480x640 2 bread_molds, 12.0ms
0: 480x640 2 bread_molds, 11.0ms
0: 480x640 2 bread_molds, 20.0ms
0: 480x640 2 bread_molds, 24.0ms
0: 480x640 2 bread_molds, 18.0ms
0: 480x640 3 bread_molds, 20.0ms
```

Issues

- False positives are common
- Noisy backgrounds confuse the model



Final Thoughts

- Easy to implement across a wide range of devices
- Could be trained on mold data from a specific product
- Could load in another model to the program as needed