

工业人工智能技术

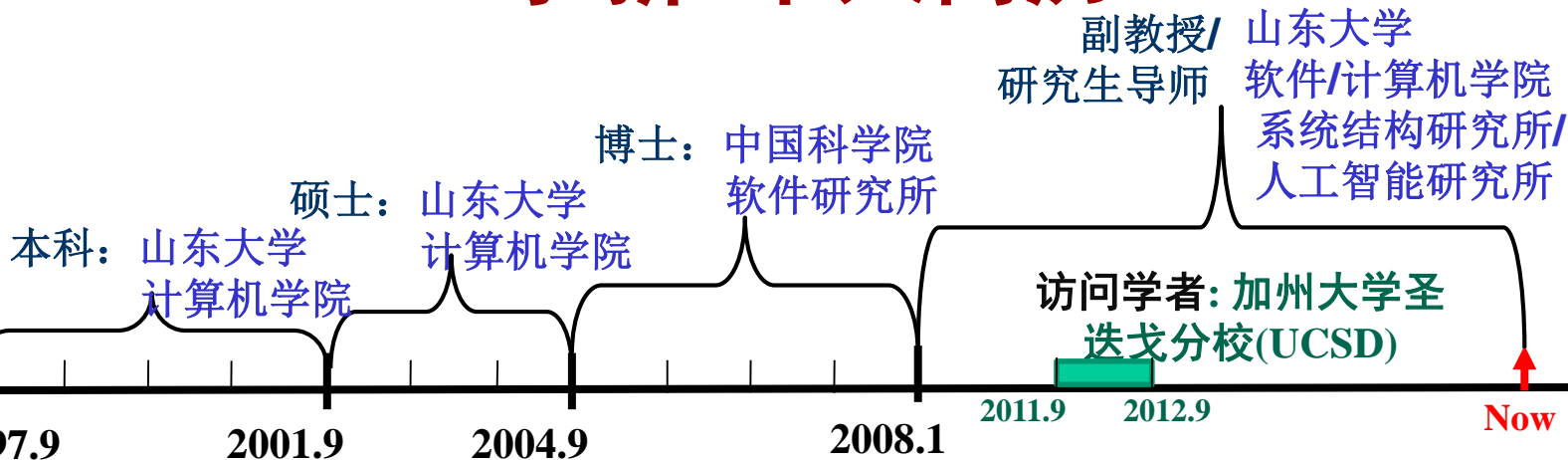
基于深度学习的目标检测

李新

山东大学 软件学院



李新 个人简历



➤ 研究方向

- 目标检测与跟踪
- 无人机智能巡检
- 大数据处理

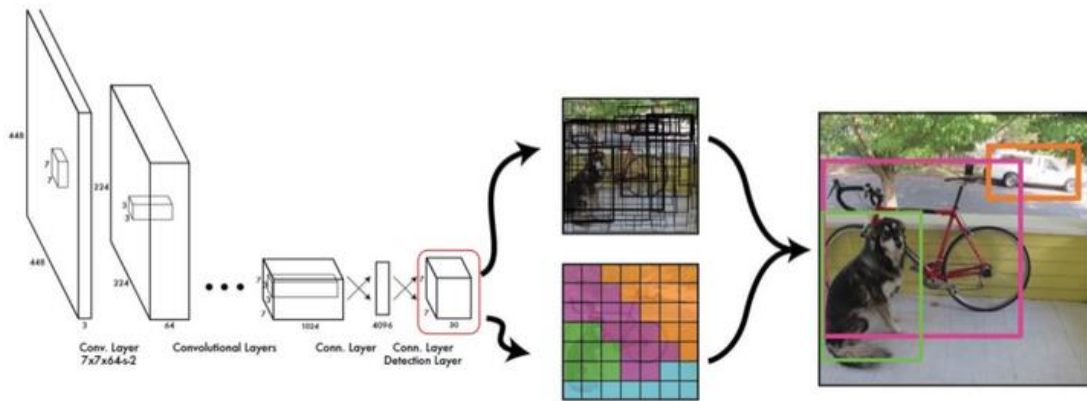
邮箱: lx@sdu.edu.cn
电话: 138-531-23559



单阶段(1-stage)检测模型代表作: Yolo



单阶段模型没有中间的区域检出过程，直接从图片获得预测结果，也被成为Region-free方法。



- 1.准备数据：将图片缩放，划分为等分的网格，每个网格按跟Ground Truth的IoU分配到所要预测的样本。
- 2.卷积网络：由GoogleNet更改而来，每个网格对每个类别预测一个条件概率值，并在网格基础上生成B个box，每个box预测五个回归值，四个表征位置，第五个表征这个box含有物体（注意不是某一类物体）的概率和位置的准确程度（由IoU表示）；
- 3.后处理：使用NMS（Non-Maximum Suppression，非极大抑制）过滤得到最后的预测框。

YOLO相比两阶段方法，其速度优势明显。但YOLO本身也存在一些问题，如划分网格较为粗糙，每个网格生成的box个数等限制了对小尺度物体和相近物体的检测。

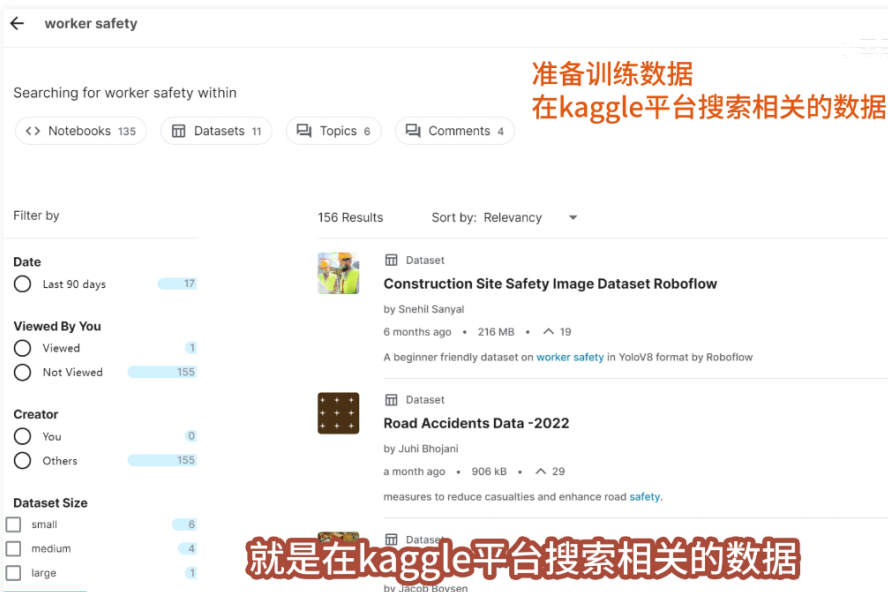
论文链接: [You Only Look Once: Unified, Real-Time Object Detection](#)

1. 准备训练数据

➤ 在kaggle平台(需翻墙)搜相关数据

搜索Safety Helmet或者 Worker safety

kaggle



<https://github.com/njvisionpower/Safety-Helmet-Wearing-Dataset>

1. 准备训练数据

➤ 在kaggle平台(需翻墙)搜相关数据

搜索Safety Helmet或者 Worker safety

Safety Helmet Detection

Improve work safety by detecting the presence of people a

Data Card Code (17) Discussion (7)

About Dataset



数据格式: PASCAL VOC

Construction Site Safety Image Dataset Roboflow

A beginner friendly dataset on worker safety in YoloV8 format by Roboflow

Data Card Code (3) Discussion (0)

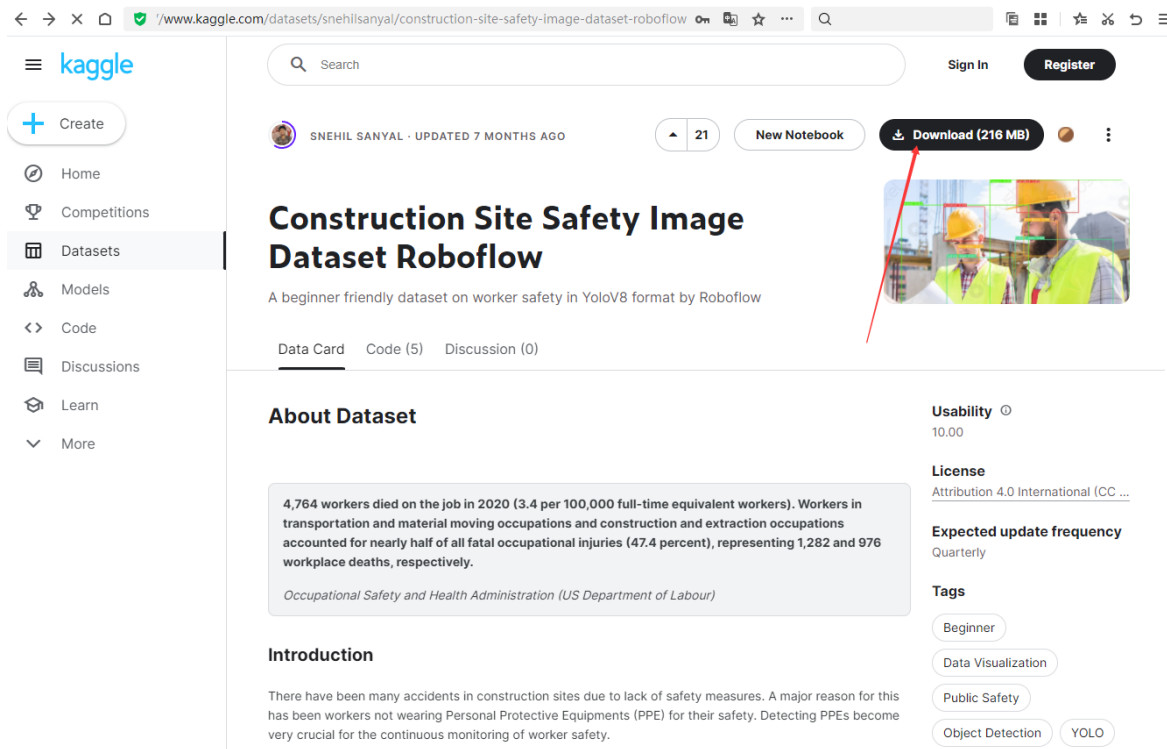


数据格式YoloV8, 可直接使用



1. 准备训练数据

➤ 在kaggle平台(需翻墙)下载数据



The screenshot shows the Kaggle website interface. The browser address bar displays the URL: www.kaggle.com/datasets/snehilsanyal/construction-site-safety-image-dataset-roboflow. The left sidebar contains navigation links: Create, Home, Competitions, Datasets (highlighted), Models, Code, Discussions, Learn, and More. The main content area shows the dataset page for 'Construction Site Safety Image Dataset Roboflow' by SNEHIL SANYAL, updated 7 months ago. The page has 21 votes and a 'New Notebook' button. A red arrow points to the 'Download (216 MB)' button. Below the title, a description states: 'A beginner friendly dataset on worker safety in YoloV8 format by Roboflow'. The 'About Dataset' section provides statistics: '4,764 workers died on the job in 2020 (3.4 per 100,000 full-time equivalent workers). Workers in transportation and material moving occupations and construction and extraction occupations accounted for nearly half of all fatal occupational injuries (47.4 percent), representing 1,282 and 976 workplace deaths, respectively.' The source is cited as 'Occupational Safety and Health Administration (US Department of Labour)'. The 'Introduction' section mentions that many accidents occur due to lack of safety measures and that detecting PPEs is crucial for continuous monitoring. On the right, the 'Usability' is 10.00, the 'License' is Attribution 4.0 International (CC ...), and the 'Expected update frequency' is Quarterly. The 'Tags' section includes Beginner, Data Visualization, Public Safety, Object Detection, and YOLO.

1. 准备训练数据

➤ 在kaggle平台(需翻墙)下载数据

The screenshot shows the Kaggle dataset page for 'Construction Site Safety Image Dataset Roboflow' by SNEHIL SANYAL. The page includes a sidebar with navigation links (Home, Competitions, Datasets, Models, Code, Discussions, Learn, More), a search bar, and a 'Download (216 MB)' button highlighted with a red arrow. The dataset description states it is a beginner-friendly dataset on worker safety in YOLOV8 format. The 'About Dataset' section mentions that 4,764 workers died on the job in 2020. The 'Introduction' section discusses the importance of safety measures in construction sites.

www.kaggle.com/datasets/snehilsanyal/construction-site-safety-image-dataset-roboflow

Search Sign In Register

SNEHIL SANYAL · UPDATED 7 MONTHS AGO 21 New Notebook Download (216 MB)

Construction Site Safety Image Dataset Roboflow

A beginner friendly dataset on worker safety in YoloV8 format by Roboflow

Data Card Code (5) Discussion (0)

About Dataset

4,764 workers died on the job in 2020 (3.4 per 100,000 full-time equivalent workers). Workers in transportation and material moving occupations and construction and extraction occupations accounted for nearly half of all fatal occupational injuries (47.4 percent), representing 1,282 and 976 workplace deaths, respectively.

Occupational Safety and Health Administration (US Department of Labour)

Introduction

There have been many accidents in construction sites due to lack of safety measures. A major reason for this has been workers not wearing Personal Protective Equipments (PPE) for their safety. Detecting PPEs become very crucial for the continuous monitoring of worker safety.

Usability 10.00

License Attribution 4.0 International (CC ...)

Expected update frequency Quarterly

Tags Beginner Data Visualization Public Safety Object Detection YOLO

迅雷下载 safemat css-data

名称

test 82个测试数据
train 2605个训练数据
valid 114个验证数据

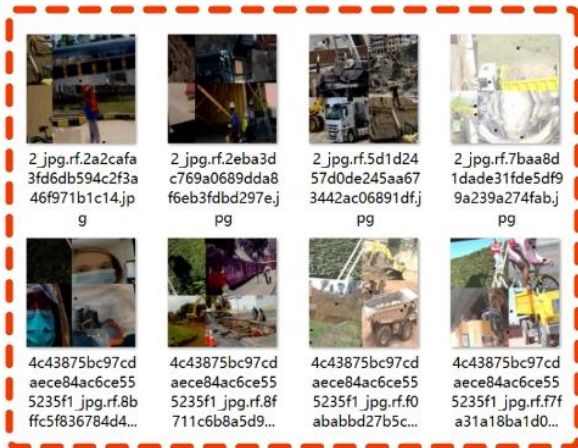
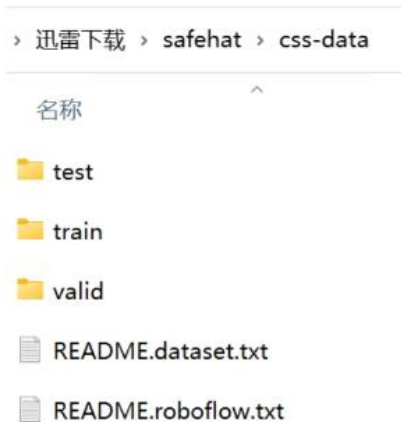
README.dataset.txt

README.roboflow.txt



2. 训练数据

2605个训练数据



train/image保存了图片数据



train/labels保存了数据的标记结果

2. 训练数据



数据图片

01094.jpg.rf.15d2452cfbd2f25016ddb1bb90f0e19a.txt					
1	0	0.4984375	0.1109375	0.021875	0.0625
2	0	0.41640625	0.1328125	0.0578125	0.0546875
3	3	0.41640625	0.18671875	0.025	0.0296875
4	7	0.39609375	0.2953125	0.1125	0.2015625
5	7	0.48828125	0.2703125	0.0421875	0.184375
6	5	0.48515625	0.30546875	0.0484375	0.4484375
7	5	0.4078125	0.315625	0.1375	0.428125
8	8	0.6828125	0.14921875	0.346875	0.2984375
9	8	0.534375	0.21171875	0.05	0.1578125
10	3	0.07578125	0.9703125	0.0828125	0.059375
11	0	0.0859375	0.83671875	0.171875	0.2359375
12	5	0.1375	0.85703125	0.275	0.2859375
13	7	0.609375	0.76484375	0.2	0.4703125
14	5	0.68046875	0.76484375	0.3421875	0.4703125

标记数据

- txt标记文件中每一行表示一个目标物体类别序号和位置

2. 训练数据

width
height

$\text{width} = 0.021 * \text{image_width}$

$\text{height} = 0.062 * \text{image_height}$



数据图片

```
01094_.jpg. rf. 15d2452cfbd2f25016ddb1bb90f0e19a. txt
1 0 0.4984375 0.1109375 0.021875 0.0625
2 0 0.41640625 0.1328125 0.0578125 0.0546875
3 3 0.41640625 0.18671875 0.025 0.0296875
4 7 0.3 class_id: 类别编号
5 7 0.4 75
6 5 0.4 x_center: 边界框中心点x坐标 1375
7 5 0.4
8 8 0.6 y_center: 边界框中心点y坐标 75
9 8 0.5
10 3 0.0 width: 边界框的宽度 75
11 0 0.0 height: 边界框的高度 75
12 5 0.1
13 7 0.609375 0.76484375 0.2 0.4703125
14 5 0.68046875 0.76484375 0.3421875 0.4703125
```

标记数据

➤ 数据只有2605张图片，包括20000多个标记结果

3. 搭建环境



```
(base) C:\Users\49034>conda create -n safehat python=3.10  
Collecting package metadata (current_repodata.json): done  
Solving environment: done
```

```
(base) C:\Users\49034>activate safehat
```

```
(safehat) C:\Users\49034>pip install ultralytics
```

创建环境并安装python: conda create -n safehat python=3.10

激活环境: activate safehat

安装YOLO: pip install ultralytics

3. 搭建环境



```
safehat.yaml
1  # 模型训练时使用的yaml配置文件。该文件说明了数据的地址和待训练类别
2  # 配置三个路径，分别对应训练数据、验证数据、测试数据的地址
3
4  # 训练数据用于模型的训练
5  train: C:\Users\49034\Desktop\yolo\safehat\data\train\images\
6  # 验证数据用于模型训练过程中的评估和参数调试
7  val: C:\Users\49034\Desktop\yolo\safehat\data\valid\images\
8  # 测试数据用于模型完成训练后的测试
9  test: C:\Users\49034\Desktop\yolo\safehat\data\test\images\
10
11 # number of classes
12 nc: 10
13
14 # class names
15 #names: ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9']
16 #Hardhat - 安全帽
17 #Mask - 口罩
18 #NO-Hardhat - 无安全帽
19 #NO-Mask - 无口罩
20 #NO-Safety Vest - 无安全背心
21 #Person - 人
22 #Safety Cone - 安全锥
23 #Safety Vest - 安全背心
24 #machinery - 机械(挖掘机)
25 #vehicle - 车辆
26 names: [Hardhat, Mask, NO-Hardhat, NO-Mask, NO-Safety Vest, Person, Safety Cone, Safety Vest, machinery, vehicle]
27
```

3. 训练



yolo_train.py 3

```
1  from ultralytics import YOLO #导入Yolo模块
2
3  # 加载yolov8的预训练模型，这个模型是yolov8使用了coco数据集训练的通用目标检测模型，
4  # 我们将它作为基础模型，在该模型的基础上，训练安全帽模型
5  model = YOLO('yolov8n.pt') # 用于加载模型
6
7  # 训练用户自定义的数据集，数据的配置保存在safehat.yaml中，epochs等于100表示100轮迭代
8  model.train(data='safehat.yaml', epochs=100)
9
10 # 使用验证集验证效果
11 model.val()
12
13
14
15
16
17
18
19
20
21
```


2. 训练文件



```
yolo_train.py 3
1  from ultralytics import YOLO #导入Yolo模块
2
3  # 加载yolov8的预训练模型，这个模型是yolov8使用了coco数据集训练的通用目标检测模型，
4  # 我们将它作为基础模型，在该模型的基础上，训练安全帽模型
5  model = YOLO('yolov8n.pt') # 用于加载模型
6
7  # 训练用户自定义的数据集，数据的配置保存在safehat.yaml中，epochs等于100表示100轮迭代
8  model.train(data='safehat.yaml', epochs=100)
9
10 # 使用验证集验证效果
11 model.val()
12
13
14
15
16
17
18
19
20
21
22
23
```

(safehat) C:\Users\49034\Desktop\yolo\safehat>python yolo_train.py

Ultralytics YOLOv8.0.170 Python-3.10.12 torch-2.0.1+cpu CPU (13th Gen Intel Core i7-13700H) 16GB
engine\trainer: task=detect, mode=train, model=yolov8n.pt, data=safehat.yaml, epochs=100, patience=50
=False, device=None, workers=8, project=None, name=None, exist_ok=False, pretrained=True, optimizer=a
ls=False, rect=False, cos_lr=False, close_mosaic=10, resume=False, amp=True, fraction=1.0, profile=Fa
out=0.0, val=True, split=val, save_json=False, save_hybrid=False, conf=None, iou=0.7, max_det=300, ha
e, save_txt=False, save_conf=False, save_crop=False, show_labels=True, show_conf=True, vid_stride=1,
augment=False, agnostic_nms=False, classes=None, retina_masks=False, boxes=True, format=torchscript,
simplify=False, opset=None, workspace=4, nms=False, lr0=0.01, lrf=0.01, momentum=0.937, weight_decay
_bias_lr=0.1, box=7.5, cls=0.5, dfl=1.5, pose=12.0, kobj=1.0, label_smoothing=0.0, nbs=64, hsv_h=0.01
ale=0.5, shear=0.0, perspective=0.0, flipud=0.0, fliplr=0.5, mosaic=1.0, mixup=0.0, copy_paste=0.0, c
in3

Overriding model.yaml nc=80 with nc=10

	from	n	params	module	arguments
0	-1	1	464	ultralytics.nn.modules.conv.Conv	[3, 16, 3, 2]
1	-1	1	4672	ultralytics.nn.modules.conv.Conv	[16, 32, 3, 2]
2	-1	1	7360	ultralytics.nn.modules.block.C2f	[32, 32, 1, True]
3	-1	1	18560	ultralytics.nn.modules.conv.Conv	[32, 64, 3, 2]
4	-1	2	49664	ultralytics.nn.modules.block.C2f	[64, 64, 2, True]
5	-1	1	73984	ultralytics.nn.modules.conv.Conv	[64, 128, 3, 2]
6	-1	2	197632	ultralytics.nn.modules.block.C2f	[128, 128, 2, True]
7	-1	1	295424	ultralytics.nn.modules.conv.Conv	[128, 256, 3, 2]
8	-1	1	460288	ultralytics.nn.modules.block.C2f	[256, 256, 1, True]
9	-1	1	164608	ultralytics.nn.modules.block.SPPF	[256, 256, 5]
10	-1	1		ultralytics.nn.modules.conv.Conv	[None, 2, 'nearest']
11	[-1, 6]	1		ultralytics.nn.modules.conv.Concat	[1]

ultralytics yolov8版本



yolo_test.py

```
1  from ultralytics import YOLO #在代码中导入yolo模块
2
3  # 导入训练好的模型best.pt
4  model = YOLO('best.pt')
5
6  # 随意找一些测试数据
7  # 图片数据和视频数据都可以，直接将数据传入接口就可以了
8  model.predict('01094_jpg.rf.15d2452cfbd2f25016ddb1bb90f0e19a.jpg', save=True)
9  model.predict('indianworkers.mp4', save=True)
10
11 # 自己构造一些数据
12 # 在识别自己构造的数据时，传入了classes = [0, 2],
13 # 代表只输出0和2，也就是安全帽是否佩戴这两个类别
14 # line_width = 30表示指定识别框的字体大小为30
15 model.predict('myself1.jpg', save=True, classes = [0, 2], line_width = 30)
16 model.predict('myself2.jpg', save=True, classes = [0, 2], line_width = 30)
17 model.predict('myself1.MP4', save=True, classes = [0, 2], line_width = 30)
18 model.predict('myself2.MP4', save=True, classes = [0, 2], line_width = 30)
```



```
(safehat) C:\Users\49034\Desktop\yolo\safehat>python yolo_test.py
```

小黑黑讲AI bilibili

```
image 1/1 C:\Users\49034\Desktop\yolo\safehat\01094.jpg.rf.15d2452cfbd2f25016ddb1bb90f0e19a.jpg: 640x640 4 Hardhats, 2 NO-Masks, 5 Persons, 3 Safety Vests, 2 machinerys, 141.5ms
```

```
Speed: 2.5ms preprocess, 141.5ms inference, 6.0ms postprocess per image at shape (1, 3, 640, 640)
```

```
Results saved to runs\detect\predict
```

WARNING inference results will accumulate in RAM unless 'stream=True' is passed, causing potential out-of-memory errors for large sources or long-running streams and videos. See <https://docs.ultralytics.com/modes/predict/> for help.

Example:

```
results = model(source=..., stream=True) # generator of Results objects
for r in results:
    boxes = r.boxes # Boxes object for bbox outputs
    masks = r.masks # Masks object for segment masks outputs
    probs = r.probs # Class probabilities for classification outputs
```

```
video 1/1 (1/648) C:\Users\49034\Desktop\yolo\safehat\indianworkers.mp4: 384x640 3 Hardhats, 2 Persons, 4 Safety Vests, 2 machinerys, 85.3ms
video 1/1 (2/648) C:\Users\49034\Desktop\yolo\safehat\indianworkers.mp4: 384x640 3 Hardhats, 2 Persons, 4 Safety Vests, 2 machinerys, 86.0ms
video 1/1 (3/648) C:\Users\49034\Desktop\yolo\safehat\indianworkers.mp4: 384x640 3 Hardhats, 2 Persons, 3 Safety Vests, 2 machinerys, 78.1ms
video 1/1 (4/648) C:\Users\49034\Desktop\yolo\safehat\indianworkers.mp4: 384x640 3 Hardhats, 2 Persons, 2 Safety Vests, 2 machinerys, 76.2ms
video 1/1 (5/648) C:\Users\49034\Desktop\yolo\safehat\indianworkers.mp4: 384x640 3 Hardhats, 2 Persons, 2 Safety Vests, 2 machinerys, 73.7ms
video 1/1 (6/648) C:\Users\49034\Desktop\yolo\safehat\indianworkers.mp4: 384x640 3 Hardhats, 2 Persons, 4 Safety Vests, 2 machinerys, 70.5ms
video 1/1 (7/648) C:\Users\49034\Desktop\yolo\safehat\indianworkers.mp4: 384x640 3 Hardhats, 2 Persons, 4 Safety Vests, 2 machinerys, 74.6ms
video 1/1 (8/648) C:\Users\49034\Desktop\yolo\safehat\indianworkers.mp4: 384x640 3 Hardhats, 2 Persons, 3 Safety Vests, 2 machinerys, 71.4ms
video 1/1 (9/648) C:\Users\49034\Desktop\yolo\safehat\indianworkers.mp4: 384x640 2 Hardhats, 2 Persons, 3 Safety Vests, 2 machinerys, 77.7ms
video 1/1 (10/648) C:\Users\49034\Desktop\yolo\safehat\indianworkers.mp4: 384x640 2 Hardhats, 2 Persons, 3 Safety Vests, 3 machinerys, 85.3ms
video 1/1 (11/648) C:\Users\49034\Desktop\yolo\safehat\indianworkers.mp4: 384x640 2 Hardhats, 2 Persons, 2 Safety Vests, 3 machinerys, 83.8ms
video 1/1 (12/648) C:\Users\49034\Desktop\yolo\safehat\indianworkers.mp4: 384x640 2 Hardhats, 2 Persons, 3 Safety Vests, 2 machinerys, 1 vehicle, 81.6ms
video 1/1 (13/648) C:\Users\49034\Desktop\yolo\safehat\indianworkers.mp4: 384x640 2 Hardhats, 2 Persons, 4 Safety Vests, 3 machinerys, 84.1ms
video 1/1 (14/648) C:\Users\49034\Desktop\yolo\safehat\indianworkers.mp4: 384x640 3 Hardhats, 2 Persons, 3 Safety Vests, 3 machinerys, 84.0ms
video 1/1 (15/648) C:\Users\49034\Desktop\yolo\safehat\indianworkers.mp4: 384x640 3 Hardhats, 2 Persons, 3 Safety Vests, 3 machinerys, 78.5ms
```



```
(safehat) C:\Users\49034\Desktop\yolo\safehat>python yolo_test.py
```

```
image 1/1 C:\Users\49034\Desktop\yolo\safehat\01094_jpg.rf.15d2452cfbd2f25016ddb1bb90f0e19a.jpg: 640x640 4 Hardhats, 2 NO-Masks, 5 Persons, 3 Safety Vests, 2 machinerys, 141.5ms  
Speed: 2.5ms preprocess, 141.5ms inference, 6.0ms postprocess per image at shape (1, 3, 640, 640)  
Results saved to runs\detect\predict
```

WARNING inference results will accumulate in RAM unless 'stream=True' is passed, causing potential out-of-memory errors for large sources or long-running streams and videos. See <https://docs.ultralytics.com/modes/predict/> for help.

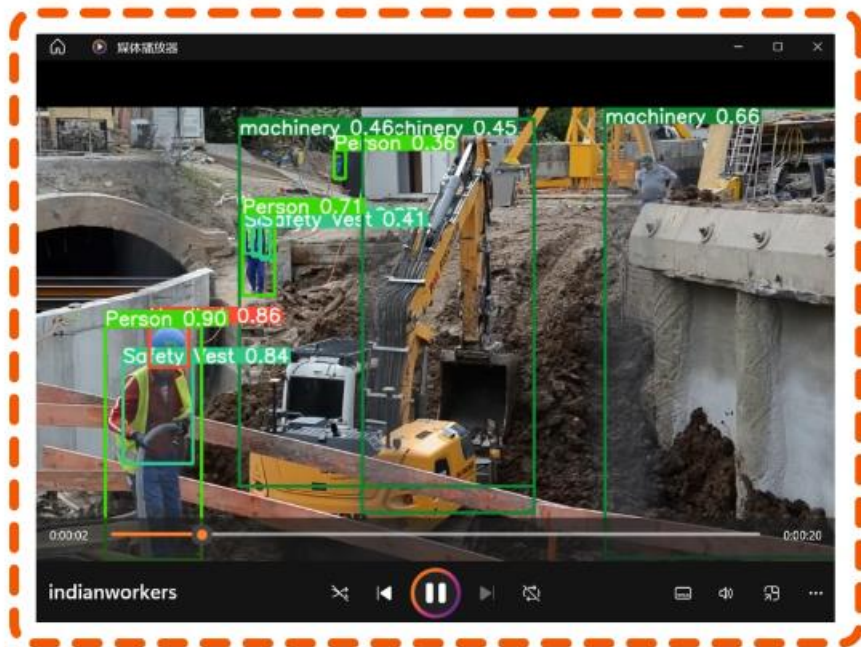
Example:

```
results = model(source=..., stream=True) # generator of Results objects  
for r in results:  
    boxes = r.boxes # Boxes object for bbox outputs  
    masks = r.masks # Masks object for segment masks outputs  
    probs = r.probs # Class probabilities for classification outputs
```

```
video 1/1 (1/648) C:\Users\49034\Desktop\yolo\safehat\indianworkers.mp4: 384x640 3 Hardhats, 2 Persons, 4 Safety Vests, 2 machinerys, 85.3ms  
video 1/1 (2/648) C:\Users\49034\Desktop\yolo\safehat\indianworkers.mp4: 384x640 3 Hardhats, 2 Persons, 4 Safety Vests, 2 machinerys, 86.0ms  
video 1/1 (3/648) C:\Users\49034\Desktop\yolo\safehat\indianworkers.mp4: 384x640 3 Hardhats, 2 Persons, 3 Safety Vests, 2 machinerys, 78.1ms  
video 1/1 (4/648) C:\Users\49034\Desktop\yolo\safehat\indianworkers.mp4: 384x640 3 Hardhats, 2 Persons, 2 Safety Vests, 2 machinerys, 76.2ms  
video 1/1 (5/648) C:\Users\49034\Desktop\yolo\safehat\indianworkers.mp4: 384x640 3 Hardhats, 2 Persons, 2 Safety Vests, 2 machinerys, 73.7ms  
video 1/1 (6/648) C:\Users\49034\Desktop\yolo\safehat\indianworkers.mp4: 384x640 3 Hardhats, 2 Persons, 4 Safety Vests, 2 machinerys, 70.5ms  
video 1/1 (7/648) C:\Users\49034\Desktop\yolo\safehat\indianworkers.mp4: 384x640 3 Hardhats, 2 Persons, 4 Safety Vests, 2 machinerys, 74.6ms  
video 1/1 (8/648) C:\Users\49034\Desktop\yolo\safehat\indianworkers.mp4: 384x640 3 Hardhats, 2 Persons, 3 Safety Vests, 2 machinerys, 71.4ms  
video 1/1 (9/648) C:\Users\49034\Desktop\yolo\safehat\indianworkers.mp4: 384x640 2 Hardhats, 2 Persons, 3 Safety Vests, 2 machinerys, 77.7ms  
video 1/1 (10/648) C:\Users\49034\Desktop\yolo\safehat\indianworkers.mp4: 384x640 2 Hardhats, 2 Persons, 3 Safety Vests, 3 machinerys, 85.3ms  
video 1/1 (11/648) C:\Users\49034\Desktop\yolo\safehat\indianworkers.mp4: 384x640 2 Hardhats, 2 Persons, 2 Safety Vests, 3 machinerys, 83.8ms  
video 1/1 (12/648) C:\Users\49034\Desktop\yolo\safehat\indianworkers.mp4: 384x640 2 Hardhats, 2 Persons, 3 Safety Vests, 2 machinerys, 1 vehicle, 81.6ms  
video 1/1 (13/648) C:\Users\49034\Desktop\yolo\safehat\indianworkers.mp4: 384x640 2 Hardhats, 2 Persons, 4 Safety Vests, 3 machinerys, 84.1ms  
video 1/1 (14/648) C:\Users\49034\Desktop\yolo\safehat\indianworkers.mp4: 384x640 3 Hardhats, 2 Persons, 3 Safety Vests, 3 machinerys, 84.0ms  
video 1/1 (15/648) C:\Users\49034\Desktop\yolo\safehat\indianworkers.mp4: 384x640 3 Hardhats, 2 Persons, 3 Safety Vests, 3 machinerys, 78.5ms
```

识别结果会保存在当前目录下的runs/detect/predict中

runs/detect/predict中





谢谢!

