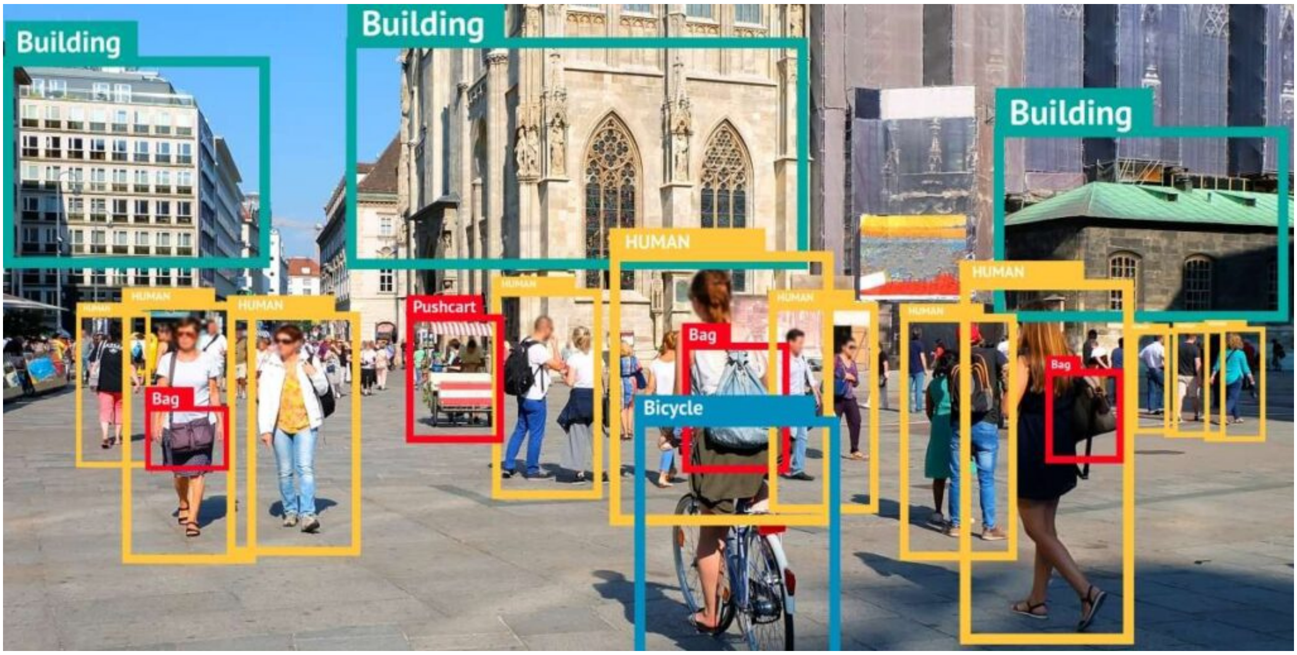
Dataintelliage

Object detection

Real time object detection using yolo

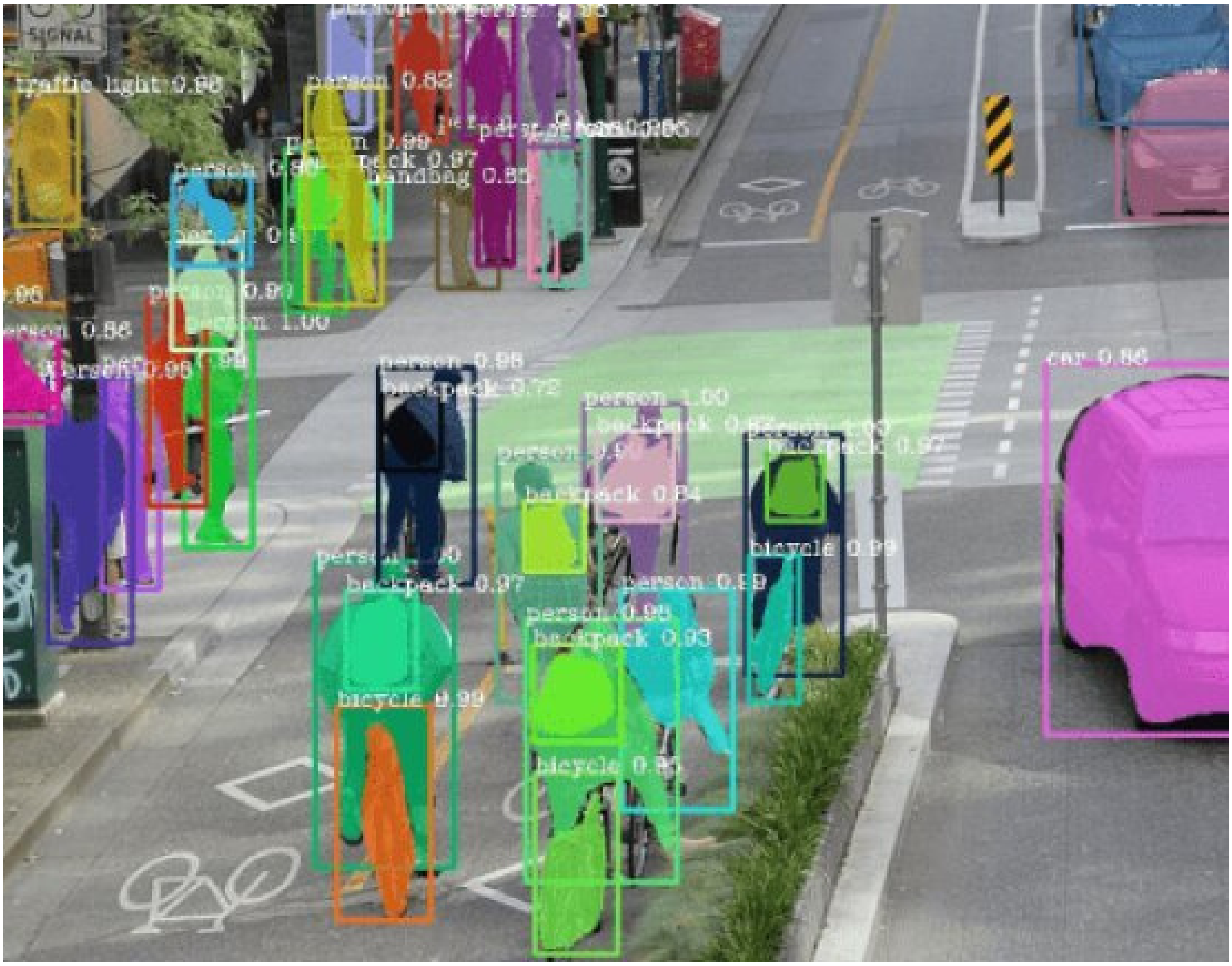
1. What is object detection?

Object detection is an important computer vision task used to detect instances of visual objects of certain classes (for example, humans, animals, cars, or buildings) in digital images such as photos or video frames. The goal of object detection is to develop computational models that provide the most fundamental information needed by computer vision applications.



2.Object segmentation

Object Segmentation is one such method that is being used in these intelligence systems and still, every day more and more papers and algorithms are developing. By applying Object Detection models, we will only be able to build a bounding box corresponding to each class in the image. But it will not tell anything about the shape of the object as the bounding boxes are either rectangular or square in shape. Image segmentation will create pixel-wise masks for each object hence it will be useful to understand granular details about the object.



Object detection

In [ ]:

```

from ultralytics import YOLO

# Load an official or custom model
model = YOLO('yolov8n.pt')  # Load an official Detect model


# Perform tracking with the model
#results = model.track(source="2.mp4", save=True)  # Tracking with default tra
results = model.track(source="D:\people on the street.jpg", save=True)
#results = model.track(source="https://youtu.be/LNwODJXcvt4", show=True, track

```

In [ ]:

```
from ultralytics import YOLO

# Configure the tracking parameters and run the tracker
model = YOLO('yolov8n.pt')

results = model.track(source="D:\mixkit-times-square-during-a-sunny-day-4442-m
```

In [4]:

```python
from ultralytics import YOLO

# Load an official or custom model
model = YOLO('yolov8n-seg.pt')  # Load an official Segment model

# Perform tracking with the model
#results = model.track(source="HeartofStoneNetflix.mp4", save=True) # Tracking
results = model.track(source="1.mp4", save=True, tracker="bytetrack.yaml")  #
```

```
    WARNING  stream/video/webcam/dir predict source will accumulate result
s in RAM unless `stream=True` is passed,
    causing potential out-of-memory errors for large sources or long-runni
ng streams/videos.

    Usage:
        results = model(source=..., stream=True)  # generator of Results o
bjects
        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 outp
uts

video 1/1 (1/172) D:\yolov8_latest\object_tracking_using_ultralytics_yolo
\1.mp4: 384x640 1 person, 1 tie, 37.4ms
video 1/1 (2/172) D:\yolov8_latest\object_tracking_using_ultralytics_yolo
\1.mp4: 384x640 1 person, 1 tie, 33.0
```

In [6]:
```
from ultralytics import YOLO

# Load an official or custom model
model = YOLO('yolov8n-pose.pt')  # Load an official Pose model

# Perform tracking with the model
results = model.track(source="1.mp4", save=True) # Tracking with default track
#results = model.track(source="https://youtu.be/LNwODJXcvt4", show=True, track
```

```

    WARNING  stream/video/webcam/dir predict source will accumulate result
s in RAM unless `stream=True` is passed,
    causing potential out-of-memory errors for large sources or long-runni
ng streams/videos.

    Usage:
        results = model(source=..., stream=True)  # generator of Results o
bjects
        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 outp
uts

video 1/1 (1/9918) D:\yolov8_latest\object_tracking_using_ultralytics_yolo
\1.mp4: 384x640 (no detections), 11.0ms
video 1/1 (2/9918) D:\yolov8_latest\object_tracking_using_ultralytics_yolo
```

In [5]:

```
from ultralytics import YOLO

# Load an official or custom model
model = YOLO('best.pt')  # Load a custom trained model

# Perform tracking with the model
results = model.track(source=, save=True)  # Tracking with default tracker
#results = model.track(source="tejas.mp4", save=True, tracker="bytetrack.yaml",
```

```
    WARNING  stream/video/webcam/dir predict source will accumulate result
s in RAM unless `stream=True` is passed,
    causing potential out-of-memory errors for large sources or long-runni
ng streams/videos.

    Usage:
        results = model(source=..., stream=True)  # generator of Results o
bjects
        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 outp
uts

video 1/1 (1/9918) D:\yolov8_latest\object_tracking_using_ultralytics_yolo
\1.mp4: 384x640 1 tank, 13.4ms
video 1/1 (2/9918) D:\yolov8_latest\object_tracking_using_ultralytics_yolo
\1.mp4: 384x640 1 tank, 13.0
```

In [1]:

```
import cv2
from ultralytics import YOLO
model = YOLO('yolov8n.pt')

video_path = 0
cap = cv2.VideoCapture(video_path)

while cap.isOpened():

    success, frame = cap.read()

    if success:
        result = model(frame, save=True)

        annotation_frame = result[0].plot()

        cv2.imshow("YOLOv8 inference", annotation_frame)

        if cv2.waitKey(1) & 0xFF == ord('q'):
            break

    else:
        break

cap.release()
cv2.destroyAllWindows()
```

```

0: 480x640 1 person, 167.6ms
Speed: 7.7ms preprocess, 167.6ms inference, 79.0ms postprocess per image a
t shape (1, 3, 480, 640)
Results saved to runs\detect\predict13

0: 480x640 1 person, 17.9ms
Speed: 11.4ms preprocess, 17.9ms inference, 0.0ms postprocess per image at
shape (1, 3, 480, 640)
Results saved to runs\detect\predict13

0: 480x640 1 person, 10.3ms
Speed: 0.0ms preprocess, 10.3ms inference, 1.4ms postprocess per image at
shape (1, 3, 480, 640)
Results saved to runs\detect\predict13

0: 480x640 1 person, 9.8ms
Speed: 0.0ms preprocess, 9.8ms inference, 1.2ms postprocess per image at s
hape (1, 3, 480, 640)
Results saved to runs\detect\predict13
```