Lab 6 – Object Detection YOLO

1. Load a pretrained YOLO model (or train one).

   https://docs.ultralytics.com/models/yolov8/

   https://opencv-tutorial.readthedocs.io/en/latest/yolo/yolo.html?highlight=yolo

   https://pyimagesearch.com/2018/11/12/yolo-object-detection-with-opencv/

   https://towardsdatascience.com/yolo-object-detection-with-opencv-and-python-21e50ac599e9

   https://www.geeksforgeeks.org/object-detection-with-yolo-and-opencv/

   https://github.com/shayantaherian/Object-detection

   https://github.com/RafaelsNeurons/Pretrained-YOLO-Object-Detection

   https://encord.com/blog/yolo-object-detection-guide/

   https://blog.eduonix.com/2022/01/real-world-implementations-of-yolo-algorithm/

   https://keylabs.ai/blog/comparing-yolov8-and-yolov7-whats-new/

2. Use this model to detect objects in images. For an image or a video draw the bounding box around the detected objects, the name of the objects and the confidence scores.

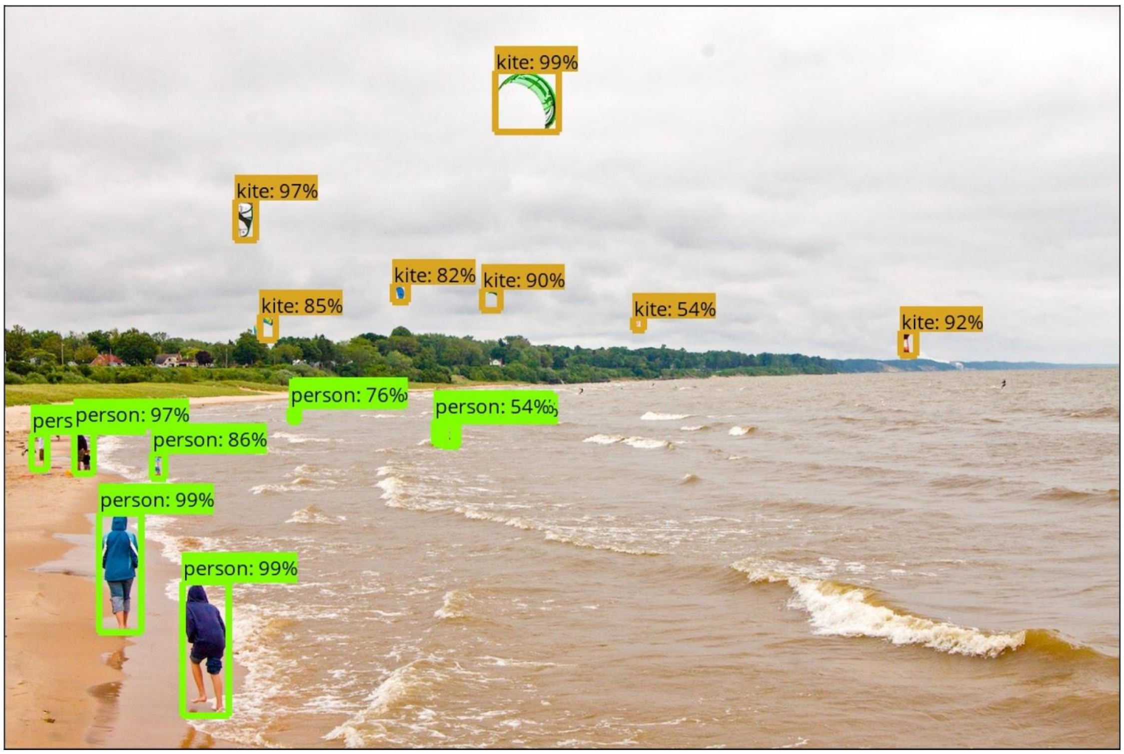
3. Take the picture in a room with at least three persons. Use YOLO to count the number of persons in the image.

4. Shoot a video in the street (with moving cars, traffic signs, persons). Use YOLO to annotate the detected objects, draw bounding boxes of 3 or 4 detected objects with highest confidence score.

5. Consider the following dataset:
   https://www.kaggle.com/datasets/trainingdatapro/cars-video-object-tracking
   Count the number of cars in each frame (car, minivan, separately). Track a certain vehicle (at your choice), from the entering frame to the exit frame. Mark the vehicle, specify the number of the entering frame and of the exiting frame.

6. Split the dataset in 70% train images (and masks) and 30% test images. Use as many images as your computer allows you. Train the YOLO model on the training images and test it on the test images. Evaluate the quality of the trained model by counting the number of well identified cars.

Object detection



(https://research.google/blog/supercharge-your-computer-vision-models-with-the-tensorflow-object-detection-api/)

Object detection in an image or video is a very important topic of Computer Vision. An object detector is a task that needs to locate the object and also identify the type of the object in an image or video. Object detection applications are an important part in autonomous driving, crowd counting and traffic monitoring, video surveillance, quality control in manufacturing, visual product search and so. The newest and the most efficient methods for object detection use deep learning approaches.

The deep learning based object detection methods can be divided into two main categories: single-shot detectors and two-stage detectors, depending on the number of types the image is analyzed.

- Two-Stage Detectors: first, they will propose candidate region and then classify the region into categories. Some of the two stage detectors are R-CNN (Regions with Convolutional Neural Networks), Fast R-CNN, Faster R-CNN, Mask R-CNN.

- Single-stage Detectors: these algorithms use a single pass through the image to accurately forecast the bounding boxes and class probabilities for every area of the picture. YOLO (You Only Look Once) and SSD (Single Shot MultiBox Detector) are two examples. Single-shot detectors are generally less accurate than the other detectors. It has problems in detecting small objects. These types of algorithms can be used to detect objects in real time situations.

(https://www.v7labs.com/blog/yolo-object-detection)
YOLO divides an input image into an S × S grid. If the center of an object falls into a grid cell, that grid cell is responsible for detecting that object. Each grid cell predicts B bounding boxes and confidence scores for those boxes. These confidence scores reflect how confident the model is that the box contains an object and how accurate it thinks the predicted box is. YOLO predicts multiple bounding boxes per grid cell. At training time, we only want one bounding box predictor to be responsible for each object. YOLO assigns one predictor to be “responsible” for predicting an object based on which prediction has the highest current IOU with the ground truth. This leads to specialization between the bounding box predictors. Each predictor gets better at forecasting certain sizes, aspect ratios, or classes of objects, improving the overall recall score. One key technique used in the YOLO models is non-maximum suppression (NMS). NMS is a post-processing step that is used to improve the accuracy and efficiency of object detection. In object detection, it is common for multiple bounding boxes to be generated for a single object in an image. These bounding boxes may overlap or be located at different positions, but they all represent the same object. NMS is used to identify and remove redundant or incorrect bounding boxes and to output a single bounding box for each object in the image.