Object Detection – white paper

August 2024

saiwa

simple artificial intelligence web application

OBJECT DETECTION

Object detection is a supervised machine learning and machine vision problem that detects instances of pre-trained classes of objects in images and videos. This means the models are trained using annotated data samples. Each image or video frame in the training dataset must be accompanied with an annotation file that includes the boundaries and classes of the objects it contains. Object detection has many applications in machine vision problems such as counting, tracking, recognition, and surveillance.

saiwa object detection service employs three recent deep neural networks: **Detecron2**, **YOLOv5** and **YOLOv7**. Each of these networks have its own applications, pros and cons. Below, you will find more technical information. Using saiwa simple UI you may try both networks pre-trained on Common Object in Context (COCO) dataset classes [1]. The COCO dataset is a large-scale object detection, segmentation, captioning and tracking dataset published by Microsoft[1].

[1] The COCO classes include the following 80 objects:
'person', 'bicycle', 'car', 'motorcycle', 'airplane', 'bus', 'train', 'truck', 'boat', 'traffic light', 'fire hydrant', 'stop sign', 'parking meter', 'bench', 'bird', 'cat', 'dog', 'horse', 'sheep', 'cow', 'elephant', 'bear', 'zebra', 'giraffe', 'backpack', 'umbrella', 'handbag', 'tie', 'suitcase', 'frisbee', 'skis', 'snowboard', 'sports ball', 'kite', 'baseball bat', 'baseball glove', 'skateboard', 'surfboard', 'tennis racket', 'bottle', 'wine glass', 'cup', 'fork', 'knife', 'spoon', 'bowl', 'banana', 'apple', 'sandwich', 'orange', 'broccoli', 'carrot', 'hot dog', 'pizza', 'donut', 'cake', 'chair', 'couch', 'potted plant', 'bed', 'dining table', 'toilet', 'tv', 'laptop', 'mouse', 'remote', 'keyboard', 'cell phone', 'microwave', 'oven', 'toaster', 'sink', 'refrigerator', 'book', 'clock', 'vase', 'scissors', 'teddy bear', 'hair drier', 'toothbrush'

Detectron2 was built by Facebook AI Research (FAIR) as an open-source fast and flexible object detection algorithm [2]. Detectron2 is a ground-up rewrite and successor of the previous Detectron version and it comes from the Mask R-CNN benchmark. Detectron2 includes high-quality implementations of state-of-the-art object detection algorithms, including: DensePose, panoptic feature pyramid networks, and numerous variants of the pioneering Mask R-CNN model [3] family, RetinaNet, Faster R-CNN, RPN, Fast R-CNN and R-FCN. Detectron efficiently detects objects in an image while simultaneously generating a high-quality segmentation mask for each instance. Figure 1 represents a few instances.

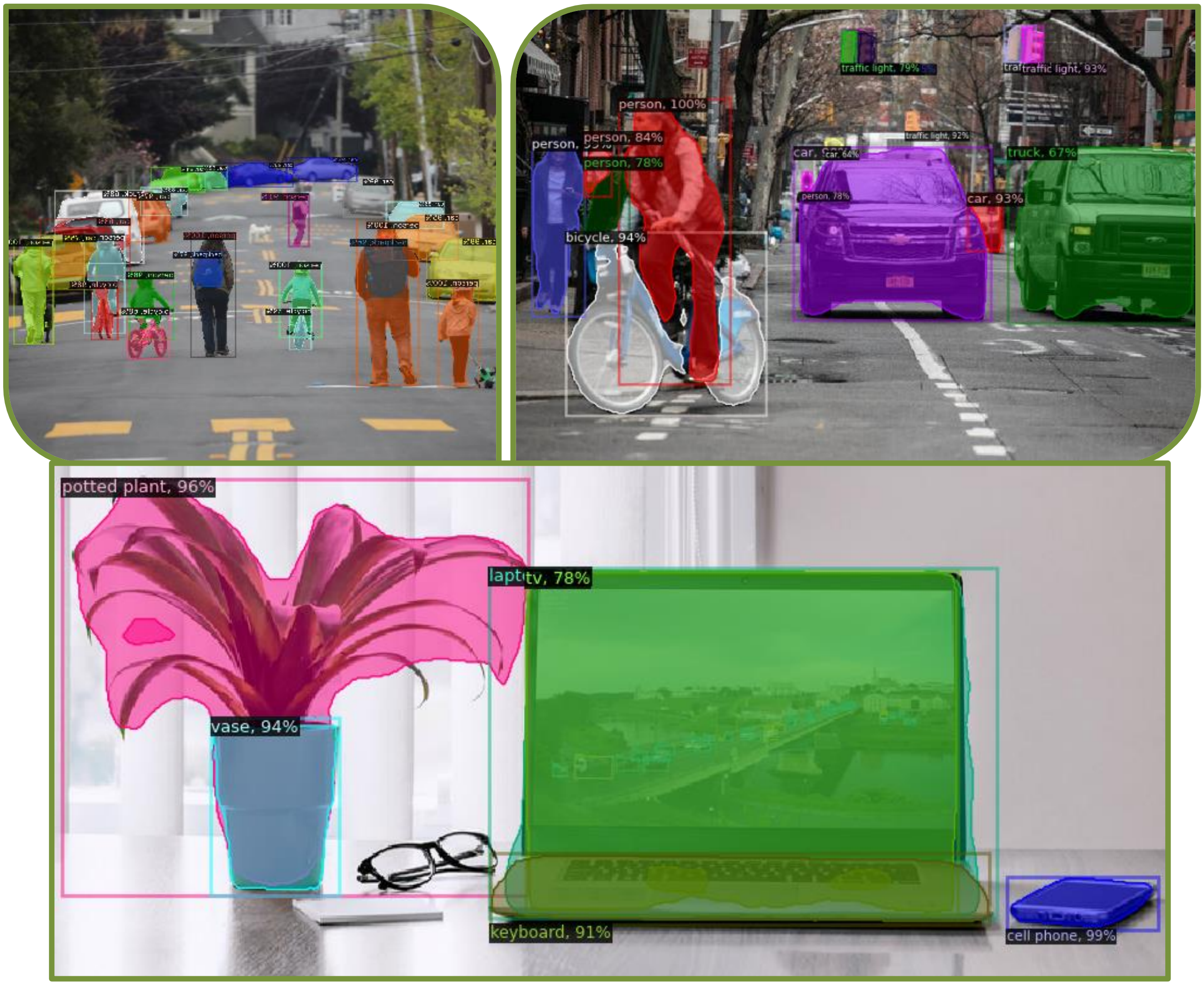


Figure 1. Detectron2 results on the COCO test set calculated using saiwa object detection online interface. Masks are shown in color, and bounding boxes, categories, and confidences are also shown.

An instance implementation of Mask R-CNN model as an ancestor of Detectron2 is shown in Figure 2 and extends the Faster R-CNN box backbones from the ResNet and FPN models. Here, Mask R-CNN extends Faster R-CNN by adding a branch to predict segmentation masks on each region of interest, in parallel with existing branch for classification and bounding box regression.

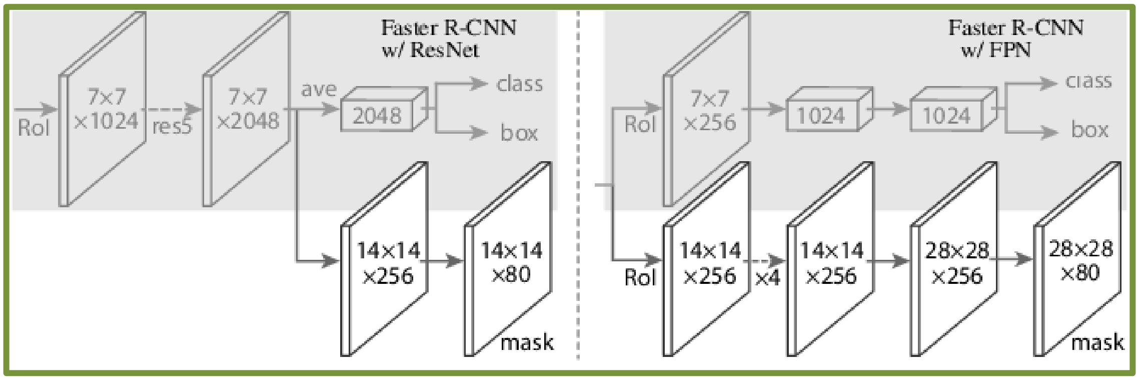
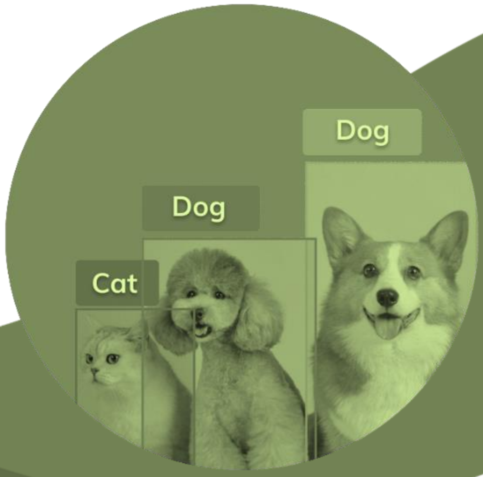


Figure 2. Mask R-CNN architecture. (printed from [3])

YOLOv5 is member of "you only live once" (YOLO) family [4]. In object detection task, the YOLO series play an important role in one-stage detectors (i.e. the algorithm requires only one single forward propagation through a neural network to detect objects). At a high level, the detection concept is to split the image into cells, each of which is responsible for predicting multiple bounding boxes along with their corresponding confidence scores. If the center of an object falls into a grid cell, that grid cell is responsible for detecting that object. The class with the maximum probability is selected and assigned to that particular grid cell.

YOLOv5 [5] is simply a PyTorch implementation of YOLOv4 and shows similar performance and shares similar design to its ancestor. YOLOv5 also proves to be significantly smaller, faster to train and more accessible to be used in a real-world applications. Figure 3 shows a few instances.



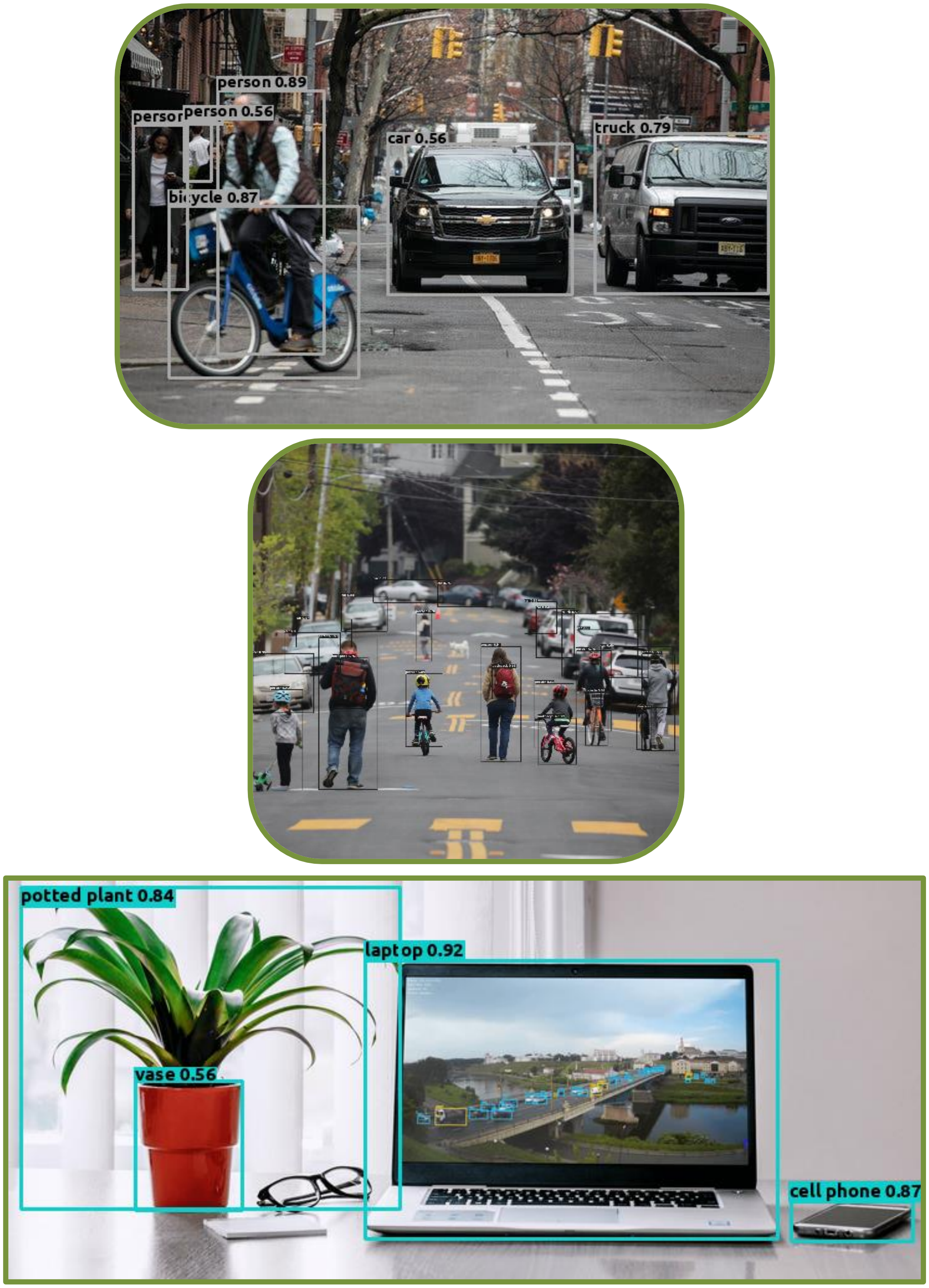


Figure 3. YOLOv5 results on the COCO test set calculated using saiwa object detection online interface. Bounding boxes are shown in color in addition to categories, and confidences.

The network architecture of YOLOv5 (Figure 4) consists of three parts: backbone, neck and head. The model backbone is the element dedicated to taking the input image and extracting it feature maps which in YOLOv5 the Cross Stage Partial (CSP) networks are used. Model neck is primarily used to generate feature pyramids. Feature pyramids help the model identify the same object with different sizes and scales. Here, YOLOv5 employs PANet. Finally, the ultimate detection is performed by model head which generates final output results with confidence scores and bounding boxes.

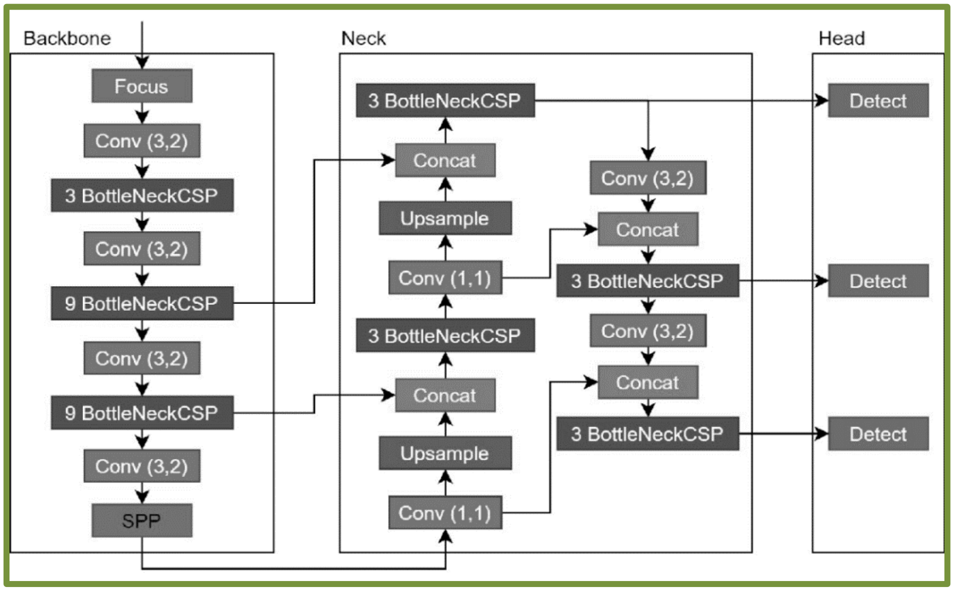


Figure 4. YOLOv5 default structure (printed from [6])

Yolov7 is member of YOLO family, just like Yolov5 [4]. As it has been shown in Figure 5, YOLOv7's speed and accuracy outperforms all known object detectors [7]. In order to obtain a real-time object detector, both the architecture and the training process have been significantly optimized in this model. The main features of YOLOv7 include the following:

- Multiple trainable bag-of-freebies methods, enabling real-time object detection to significantly improve accuracy without increasing inference costs,
- Addressing two issues: how to replace the original module with a re-parameterised one, and how to deal with the assignment to different output layers in a dynamic label assignment strategy,
- Real-time object detector extend and compound scaling methods,
- Faster inference speed and higher detection accuracy with approximately 40% fewer parameters and 50% fewer computations compared to known real-time object detectors.

Figure 6 shows a few detection results that are generated using saiwa object detection service interface.

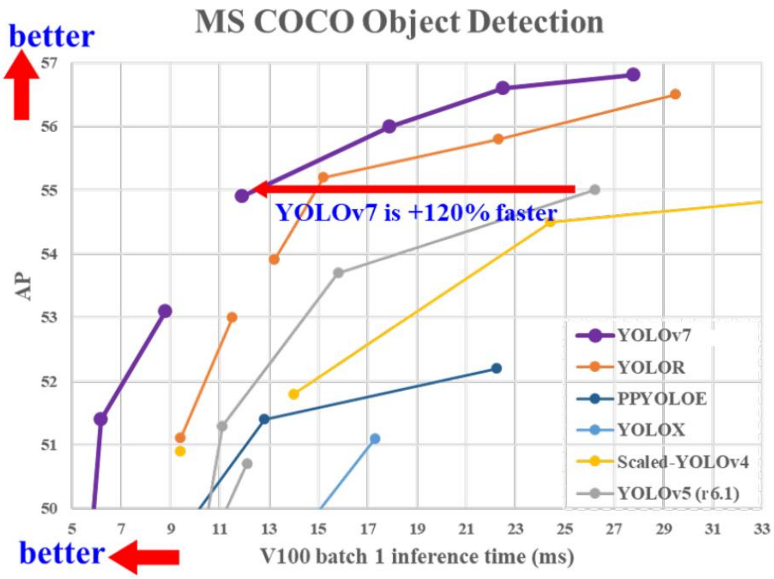


Figure 5. The comparison results of YOLOv7 with other real-time object detectors show that the proposed methods achieve state-of-the-art performance (printed from [8]).

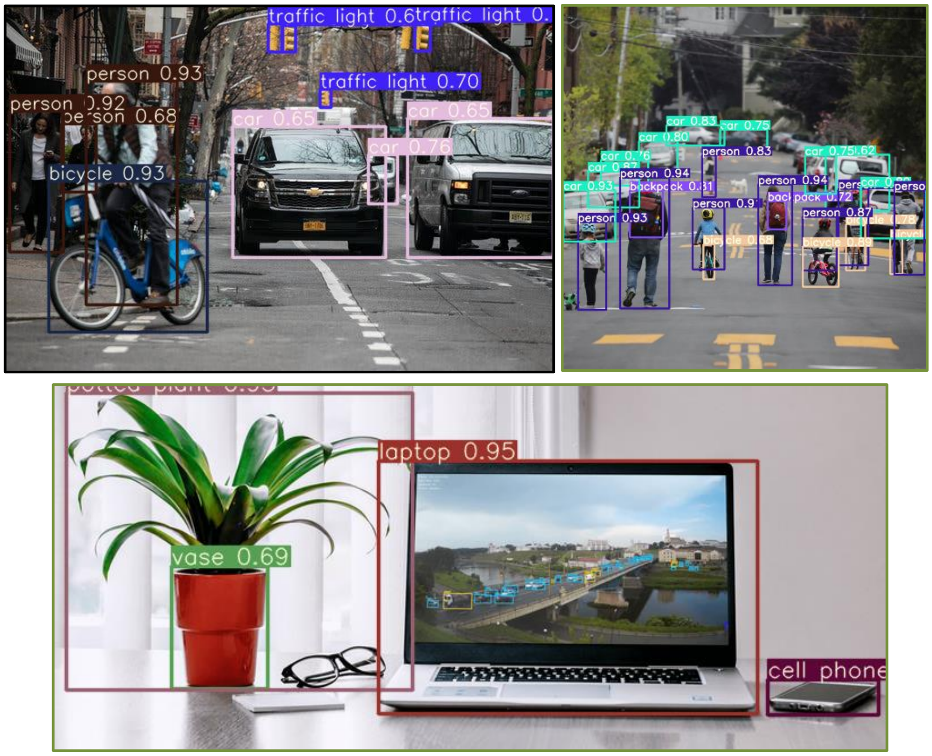


Figure 6. YOLOv7 results on the COCO test set calculated using saiwa object detection online interface. Bounding boxes are shown in color in addition to categories, and confidences.

References:

[1] Lin, Tsung-Yi, et al. "Microsoft coco: Common objects in context.", European conference on computer vision. Springer, Cham, 2015.

[2]Wu,Y.,etal."Detectron2".Availableonline https://github.com/facebookresearch/detectron2, 2019.

[3] He, Kaiming, et al. "Mask r-cnn." Proceedings of the IEEE international conference on computer vision. 2017.

[4] Redmon, Joseph, et al. "You only look once: Unified, real-time object detection." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[5] Ultralytics-Yolov5.github.com/ultralytics/yolov5, 2021.

[6] Benjumea, Aduen, et al. "YOLO-Z: Improving small object detection in YOLOv5 for autonomous vehicles." arXiv preprint arXiv:2112.11798, 2021.

[7] Wang, Chien-Yao, et al. "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors." arXiv preprint arXiv:2207.02696 (2022).

[8] YOLOv7. https://github.com/WongKinYiu/yolov7, 2022.

info@ saiwa.ai

720 Guelph Line Burlington, ON L7R 4E2 Building
+15148131809
www.linkedin.com/company/saiwa
www.instagram.com/saiwa.ai