**Object Recognition: YOLO (You Only Look Once)**

YOLO is a clever convolutional neural network (CNN) for doing object detection in real-time. The algorithm applies a single neural network to the full image, and then divides the image into regions and predicts bounding boxes and probabilities for each region.

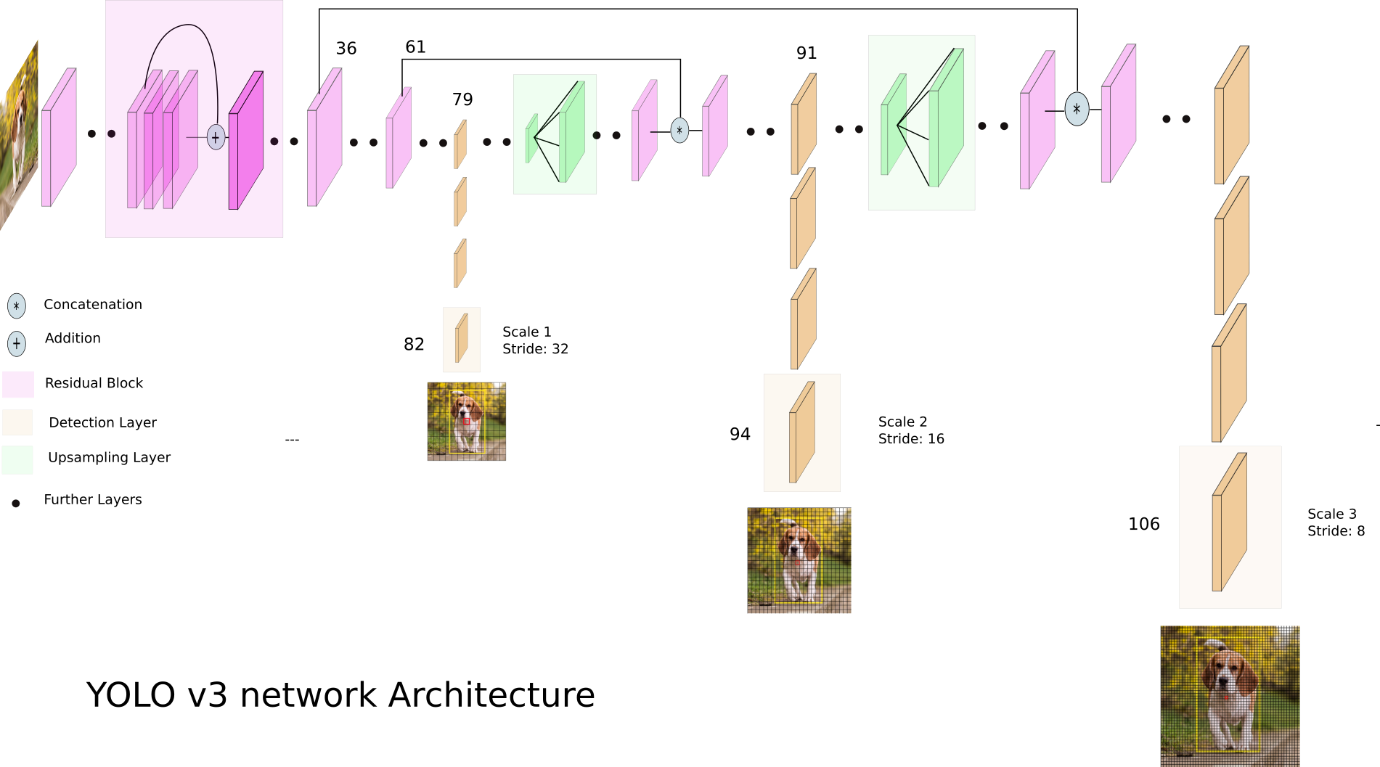
The algorithm “only looks once” at the image in the sense that it requires only one forward propagation pass through the neural network to make predictions. After non-max suppression (which makes sure the object detection algorithm only detects each object once), it then outputs recognized objects together with the bounding boxes.

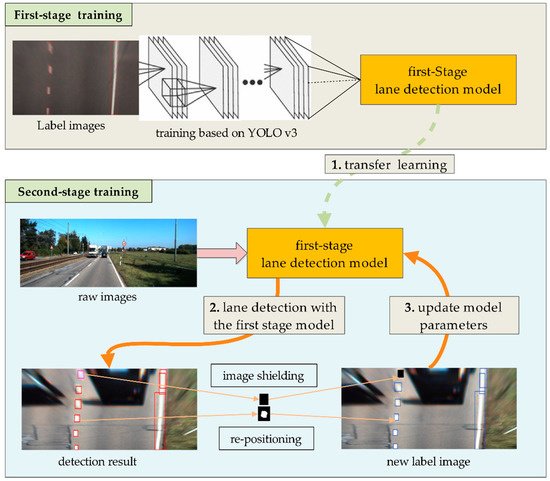
With YOLO, a single CNN simultaneously predicts multiple bounding boxes and class probabilities for those boxes. YOLO trains on full images and directly optimizes detection performance.

YOLO learns generalizable representations of objects so that when trained on natural images and tested on the artwork, the algorithm outperforms other top detection methods.

YOLOv3 uses a variant of Darknet, a framework to train neural networks, which originally has 53 layers. For the detection task, another 53 layers are stacked onto it, accumulating to a total of a 106-layer fully convolutional architecture. This explains the reduction in speed in comparison with the second version, which only has 30 layers.

* **YOLOv3 Architecture Diagram:**



[](https://www.mdpi.com/sensors/sensors-18-04308/article_deploy/html/images/sensors-18-04308-g001-550.jpg)YOLOv3 makes detections at three different scales. The detection is done by applying 1 x 1 detection kernels on feature maps of three different sizes at three different places in the network. The shape of the detection kernel is 1 x 1 x (B x (5 + C) ). Here B is the number of bounding boxes a cell on the feature map can predict, “5” is for the 4 bounding box attributes and one object confidence, and C is the number of classes. In YOLO v3 trained on COCO, B = 3 and C = 80, so the kernel size is 1 x 1 x 255. The feature map produced by this kernel has identical height and width to the previous feature map and has detection attributes along with the depth as described above.

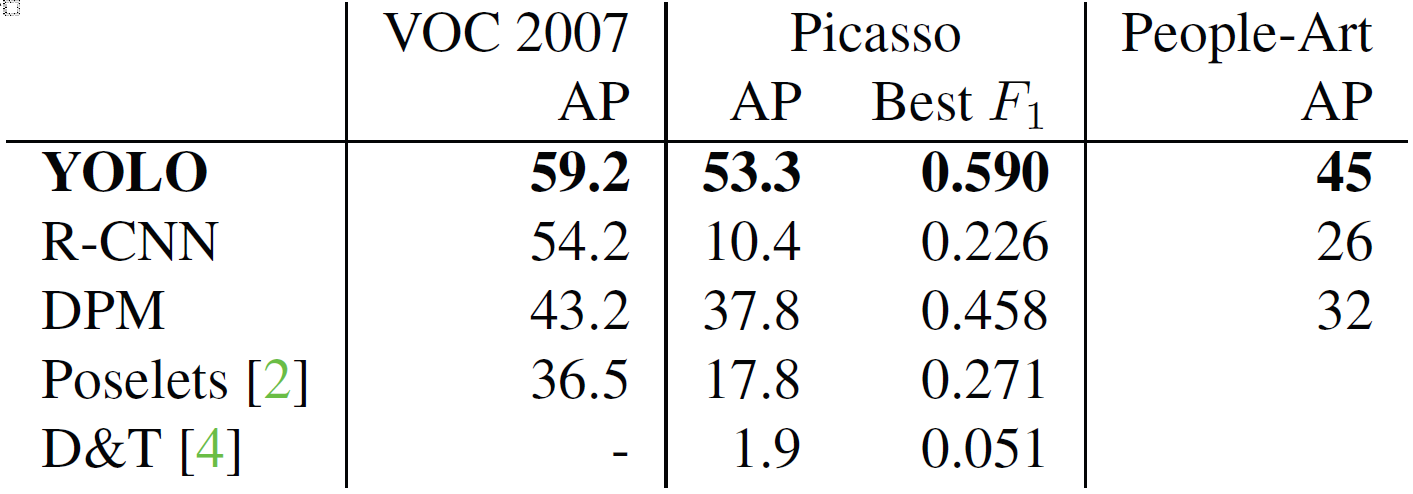
|  |  |
| --- | --- |
| **Sample frame from the input video** | **Sample frame from the output video** |
|  |  |
|  |  |
|  |  |
|  |  |

* [**YOLOv3 configuration parameters**](https://learnopencv.com/training-yolov3-deep-learning-based-custom-object-detector/)**:**

|  |  |
| --- | --- |
| Batch hyper-parameter: | batch=64  subdivisions=16 |
| Width, Height, Channels: | width=416  height=416  channels=3 |
| Momentum and Decay: | momentum=0.9  decay=0.0005 |
| Learning Rate, Steps, Scales, Burn In: | learning\_rate=0.001  policy=steps  steps=3800  scales=.1  burn\_in=400 |
| Data augmentation | angle=0  saturation = 1.5  exposure = 1.5  hue=.1 |
| Number of iterations | max\_batches=5200 |

**From Literature Survey:**

**[You Only Look Once: Unified, Real-Time Object Detection (IEEE):](https://ieeexplore.ieee.org/document/7780460)**

YOLO, a new approach to object detection. Prior work on object detection repurposes classifiers to perform detection. Instead, they frame object detection as a regression problem to spatially separated bounding boxes and associated class probabilities. A single neural network predicts bounding boxes and class probabilities directly from full images in one evaluation. Since the whole detection pipeline is a single network, it can be optimized end-to-end directly on detection performance. Their unified architecture is extremely fast. Our base YOLO model processes images in real-time at 45 frames per second. A smaller version of the network, Fast YOLO, processes an astounding 155 frames per second while still achieving double the mAP of other real-time detectors. Compared to state-of-the-art detection systems, YOLO makes more localization errors but is less likely to predict false positives in the background. Finally, YOLO learns very general representations of objects. It outperforms other detection methods, including DPM and R-CNN, when generalizing from natural images to other domains like artwork.