Cigarette Smoking Detection

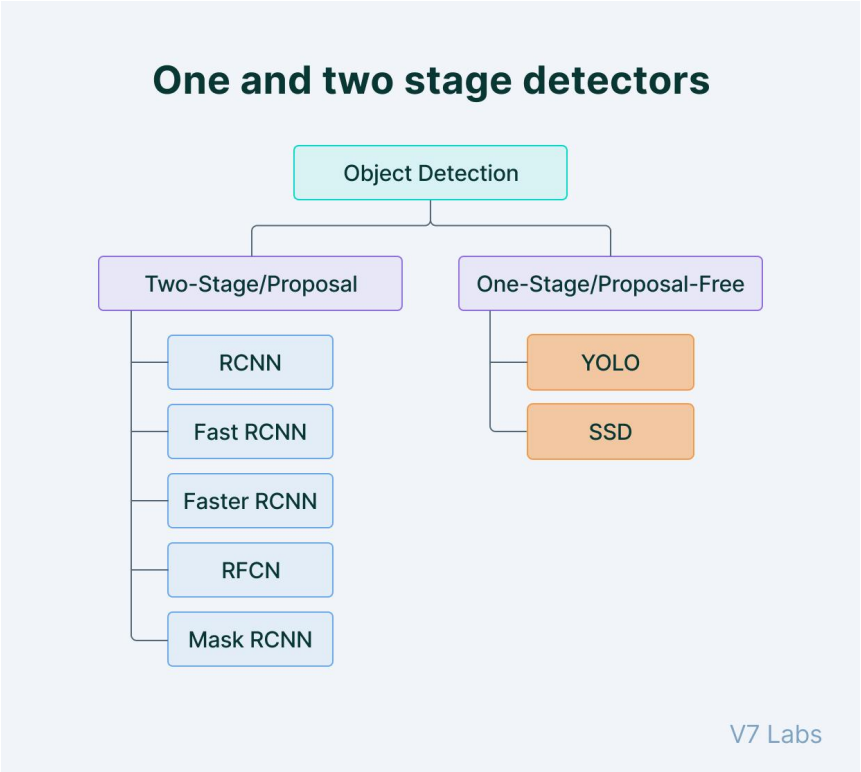
Object detection overview

Object detection is an advanced form of image classification where a neural network predicts objects in an image and points them out in the form of bounding boxes.

Object detection thus refers to the detection and localization of objects in an image that belong to a predefined set of classes.

Tasks like detection, recognition, or localization find widespread applicability in real-world scenarios, making object detection (also referred to as object recognition) a very important subdomain of Computer Vision.

Few popular Object detection models



YOLO

Compared to the approach taken by object detection algorithms before YOLO, which repurpose classifiers to perform detection, YOLO proposes the use of an end-to-end neural network that makes predictions of bounding boxes and class probabilities all at once.

Following a fundamentally different approach to object detection, YOLO achieves state-of-the-art results beating other real-time object detection algorithms by a large margin.

YOLO VS R-CNN

The YOLO v7 (released in 2022) algorithm achieves the highest accuracy among all other real-time object detection models – while achieving 30 FPS or higher using a GPU V100. Compared to the best performing Cascade-Mask R-CNN models, YOLOv7 achieves 2% higher accuracy at a dramatically increased inference speed (509% faster).

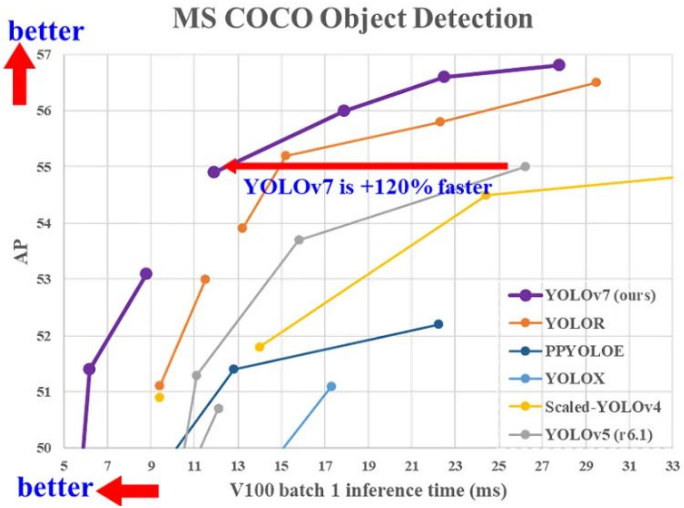


Image labeling

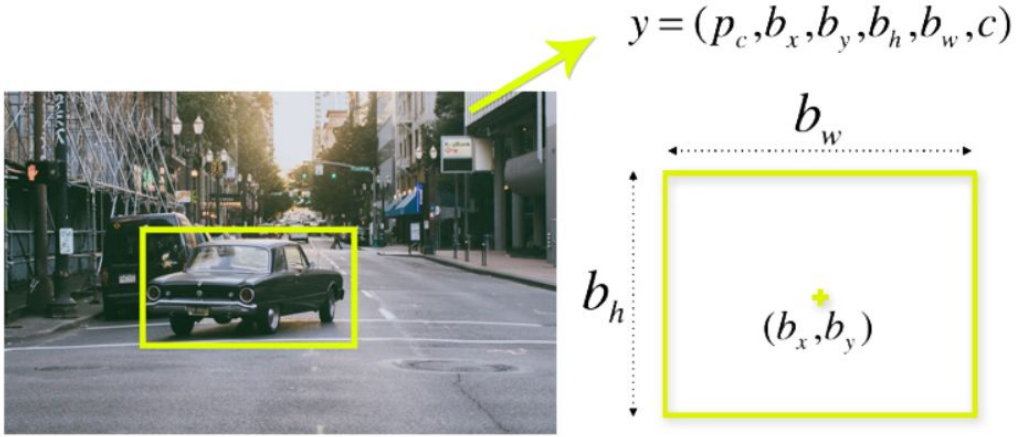
What is image annotation? Image annotation is the task of labeling digital images, typically involving human input and, in some cases, computer-assisted help. Labels are predetermined by a machine learning (ML) engineer and are chosen to give the computer vision model information about the objects present in the image.

Image labeling gives you insight into the content of images. When you use the API, you get a list of the entities that were recognized: people, things, places, activities, and so on. Each label found comes with a score that indicates the confidence the ML model has in its relevance.

Understanding YOLO object detection: the YOLO algorithm

To understand the YOLO algorithm, it is necessary to establish what is actually being predicted. Ultimately, we aim to predict a class of an object and the bounding box specifying object location. Each bounding box can be described using four descriptors

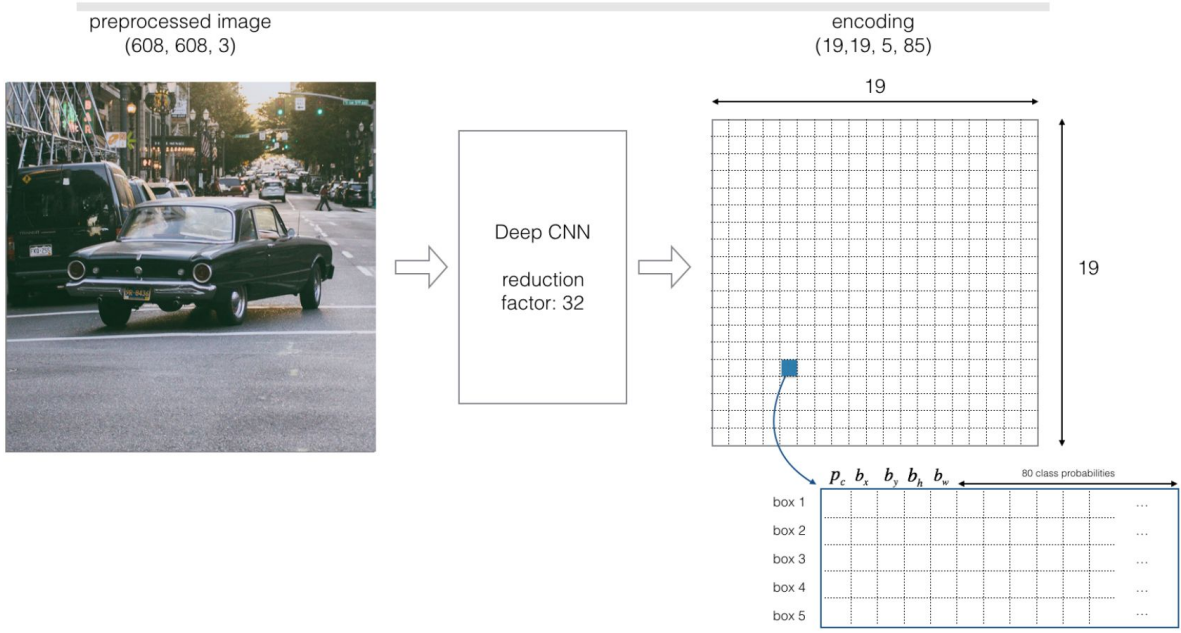
1. center of a bounding box (**bxby**)
2. width (**bw**)
3. height (**bh**)
4. value **cis** corresponding to a class of an object (e.g., car, traffic lights, etc.)



To learn more about **PP-YOLO**.In addition, we have to predict the pc value, which is the probability that there is an object in the bounding box.

Understanding YOLO

Instead, we are splitting our image into cells, typically using a 19×19 grid. Each cell is responsible for predicting 5 bounding boxes (in case there is more than one object in this cell). Therefore, we arrive at a large number of 1805 bounding boxes for one image. Rather than seizing the day with #YOLO and Carpe Diem, we're looking to seize object probability. The exchange of accuracy for more speed isn't reckless behavior, but a necessary requirement for faster real-time object detection.



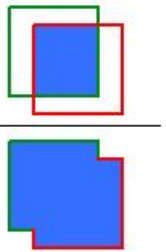
Intersection over Union (IoU)

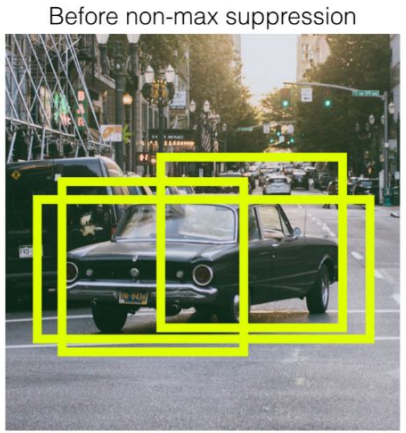
Intersection over Union is a popular metric to measure localization accuracy and calculate localization errors in object detection models.

To calculate the IoU with the predictions and the ground truth, we first take the intersecting area between the bounding boxes for a particular prediction and the ground truth bounding boxes of the same area. Following this, we calculate the total area covered by the two bounding boxes—also known as the *Union*.

The intersection divided by the Union, gives us the ratio of the overlap to the total area, providing a good estimate of how close the bounding box is to the original prediction.

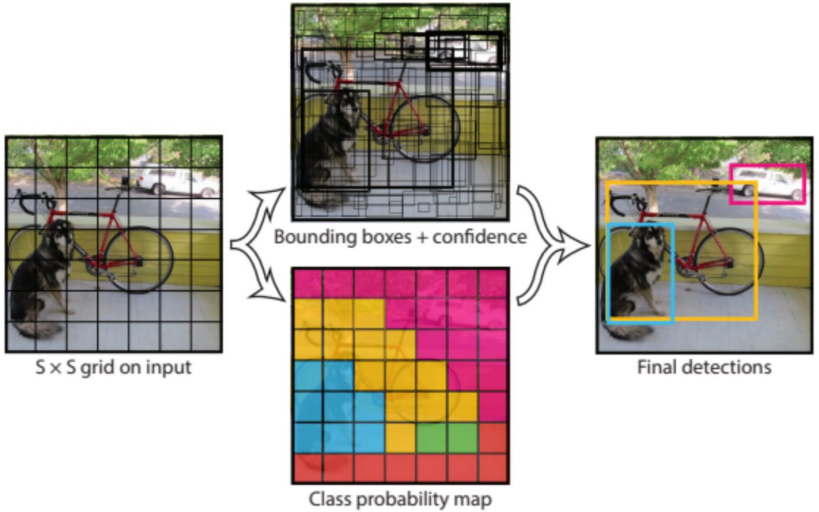
$$IOU = \frac{\text{area of overlap}}{\text{area of union}} =$$





How does YOLO work?

The YOLO algorithm works by dividing the image into N grids, each having an equal dimensional region of SxS. Each of these N grids is responsible for the detection and localization of the object it contains. Correspondingly, these grids predict B bounding box coordinates relative to their cell coordinates, along with the object label and probability of the object being present in the cell.This process greatly lowers the computation as both detection and recognition are handled by cells from the image, but—It brings forth a lot of duplicate predictions due to multiple cells predicting the same object with different bounding box predictions.



Most of these cells and bounding boxes will not contain an object. Therefore, we predict the value pc, which serves to remove boxes with low object probability and bounding boxes with the highest shared area in a process called **non-max suppression**.

YOLO Architecture

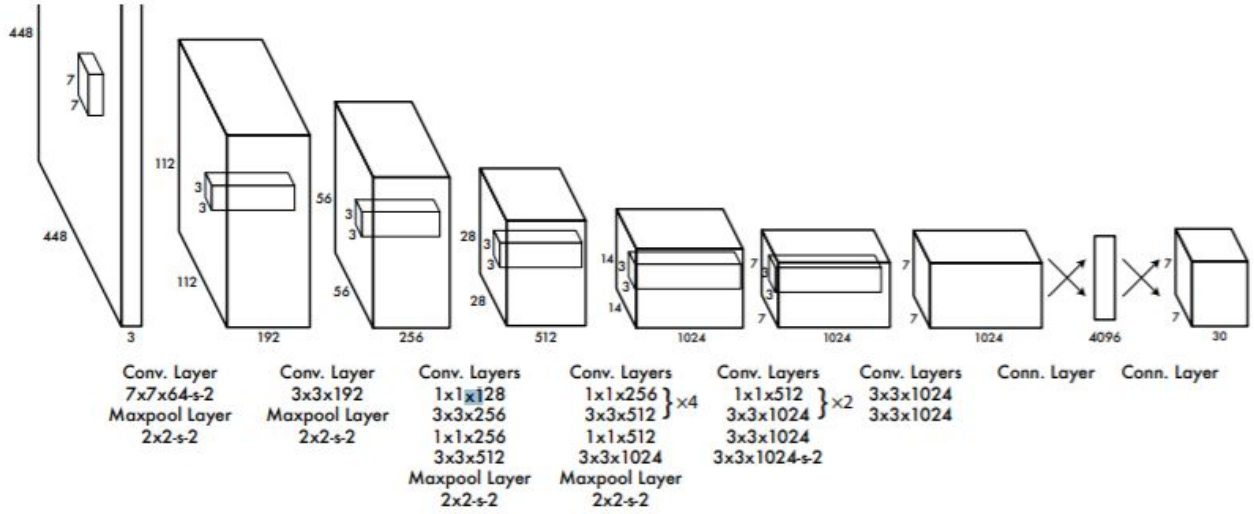
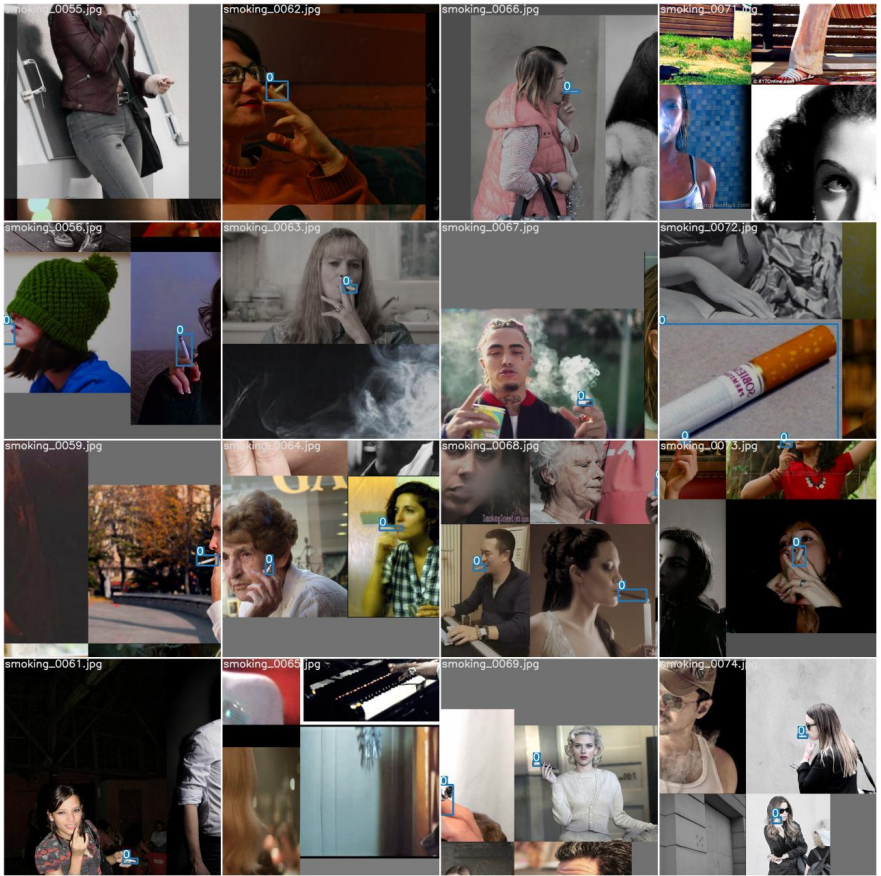


Figure 3: The Architecture. Our detection network has 24 convolutional layers followed by 2 fully connected layers. Alternating 1×1 convolutional layers reduce the features space from preceding layers. We pretrain the convolutional layers on the ImageNet classification task at half the resolution (224×224 input image) and then double the resolution for detection.

Our Data

Training (994)



Validation(193)

Average Precision (AP)

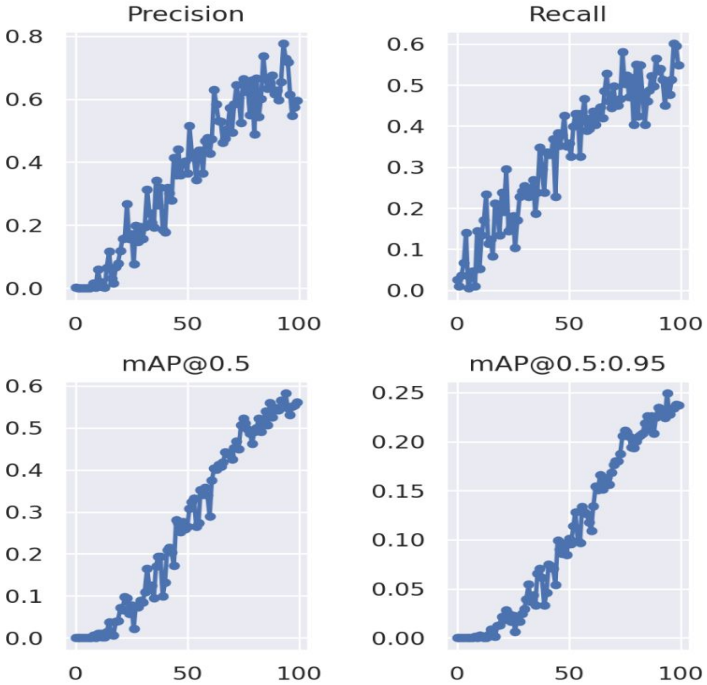
Average Precision is calculated as the area under a precision vs recall curve for a set of predictions.

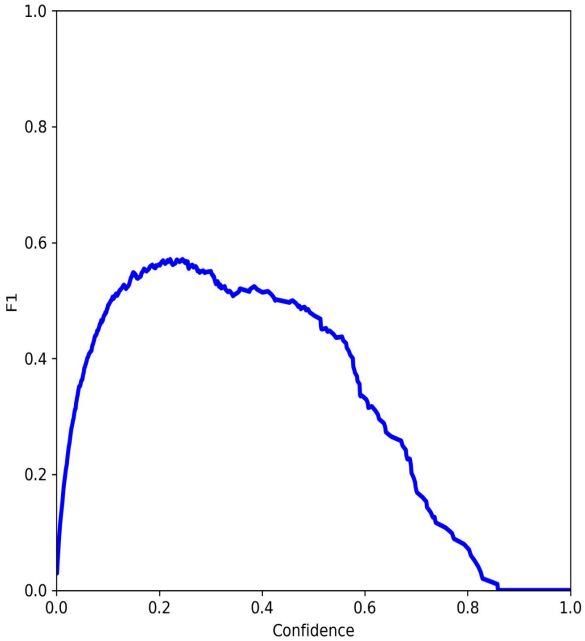
Recall is calculated as the ratio of the total predictions made by the model under a class with a total of existing labels for the class.

On the other hand, Precision refers to the ratio of true positives with respect to the total predictions made by the model.

The area under the precision vs recall curve gives us the Average Precision per class for the model. The average of this value, taken over all classes, is termed as mean Average Precision (mAP).

Average Precision (AP) First 100 epochs

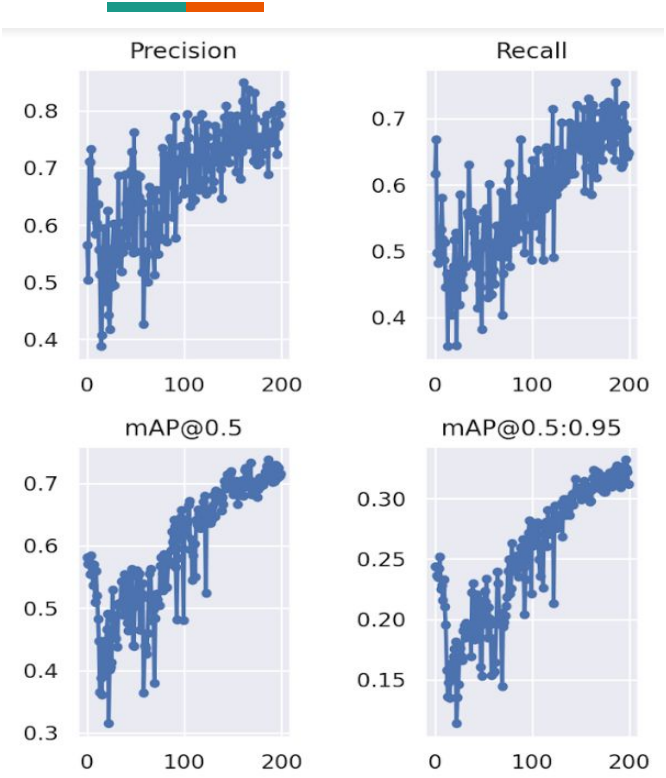


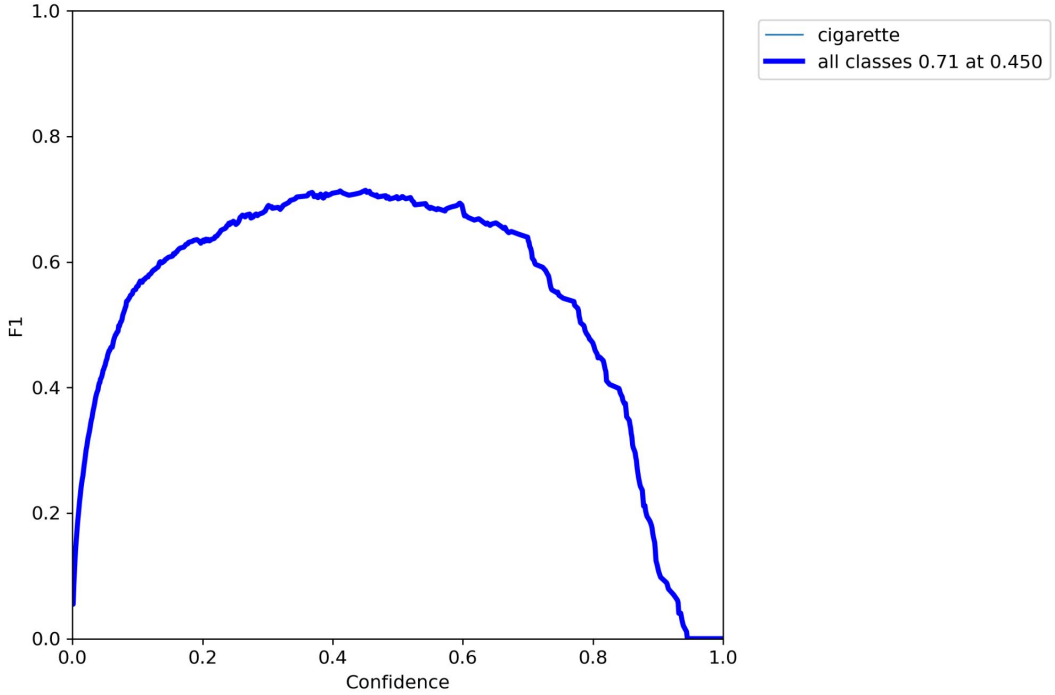


F1 score

mAP at 0.5 confidence is 0.58

Next 200 epochs





Further Improvement

need of further improvement in model...