Implement of YoloV3

Jianxiao Yang, Yachen Wang, Yuxi Ge, Chenjiayi He

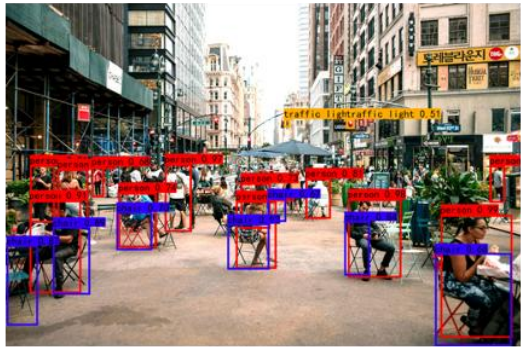
CS/ECE 523 Deep Learning Project
Boston University

Task:Object Detection

Given an input image, the model should identify and locate the objects present in the image, while providing bounding boxes around each object and assigning a class label to it.

Input

Output



Balance: fast and accurate

YoloV3:

designed to be fast, with a single-shot architecture that processes an image in one forward pass

Compared to former version

Detection at three Scales

Better at detecting smaller objects

Better, *not* Faster, Stronger

Related work

Before YOLO, there were two main classes of object detection methods: region-based methods and sliding window methods.

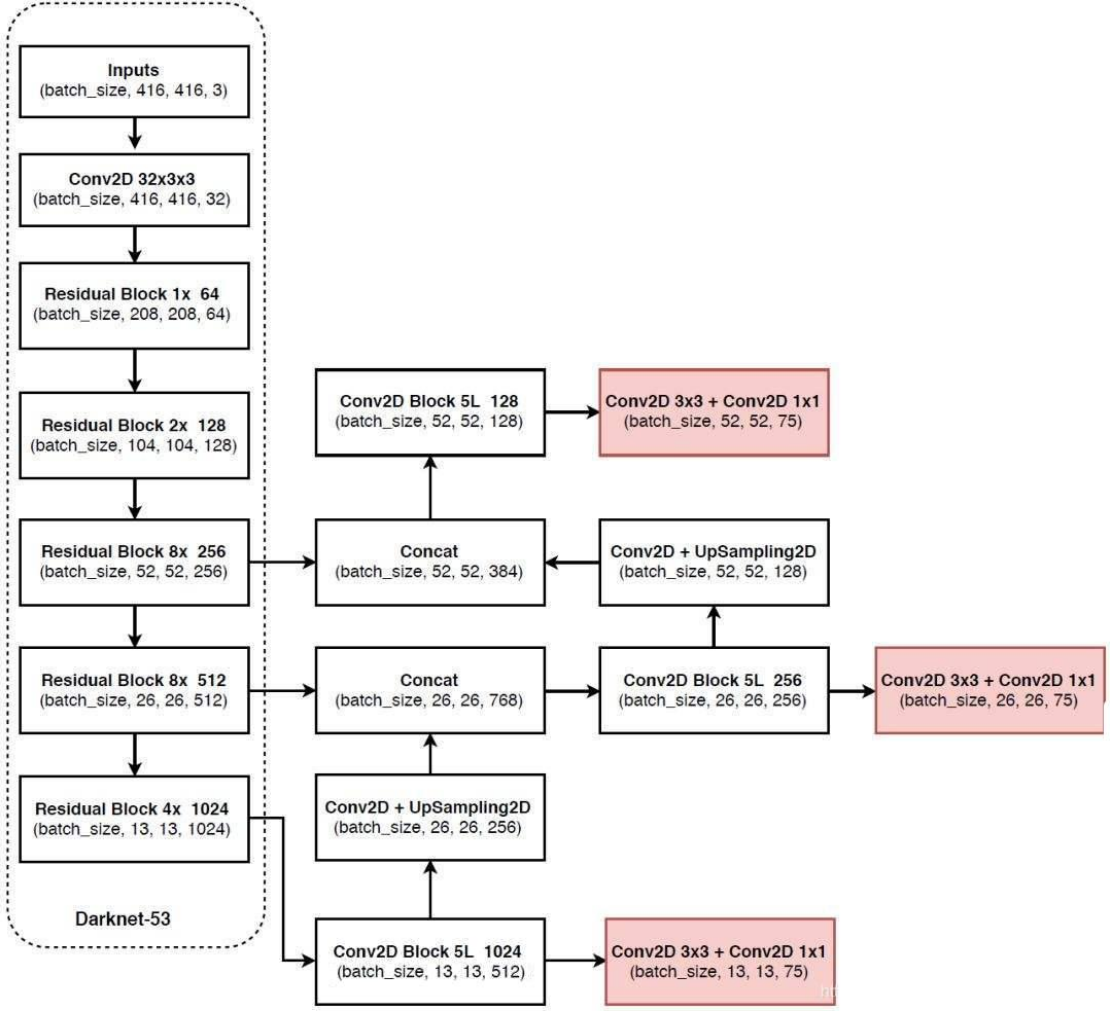
Region-based methods: such as R-CNN, Fast R-CNN, and Faster R-CNN. These methods first generate a set of potential object regions (box proposals) and then classify these regions. These methods achieve good results in accuracy, but are relatively slow and not suitable for real-time scenarios.

Sliding window method: such as DPM (Deformable Part Models). These methods slide a fixed-size window across different scales and positions, and classify each window. While this approach can handle multi-scale problems, it is computationally expensive due to the large number of windows that need to be traversed.

YOLO3 is an improvement based on YOLO and YOLOv2. The YOLO3 paper is "YOLOv3: An Incremental Improvement" by Joseph Redmon and Ali Farhadi. This paper improves on the previous two versions of YOLO, including higher accuracy, faster detection speed and better detection of multi-scale objects. YOLO3 adopts a new network structure called Darknet-53, and combines multi-scale detection and a new loss function to improve performance. We have successfully implemented YOLO3.

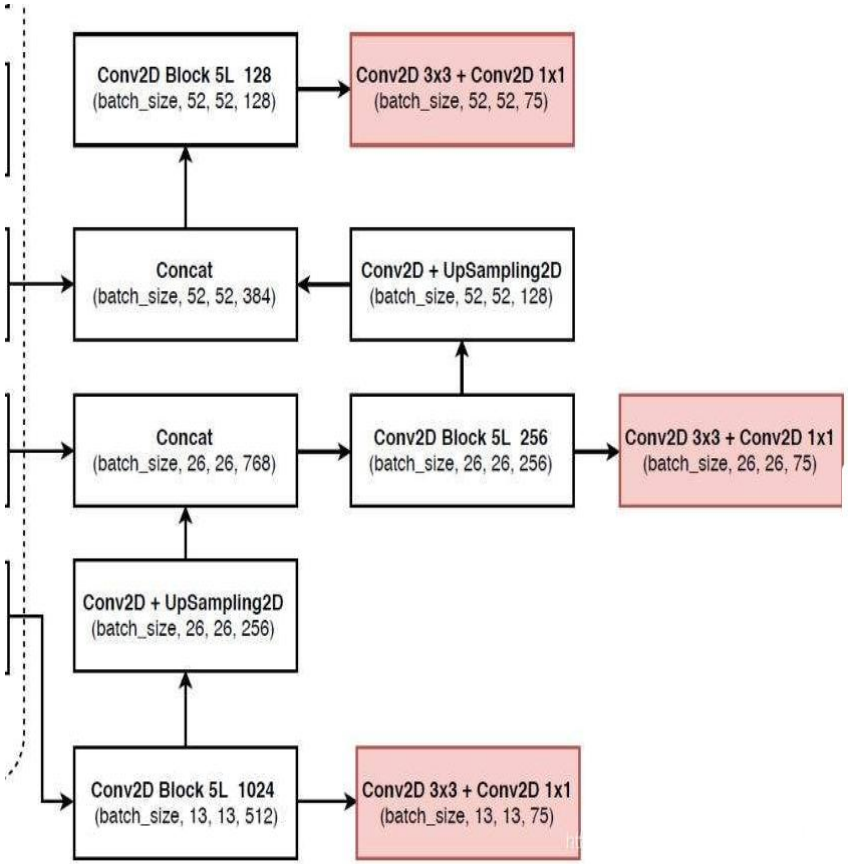
Approach

- ***Darknet-53***
  - *Backbone feature extraction network*
  - *Downsampling: The process of a series of convolutions that continually reduce the height and width*
  - *Using the Residual Net：Easy to optimize and able to improve accuracy*



Approach

- ***Predict from the extracted feature***
  - *Building an FPN (Feature Pyramid Network) to enhance feature extraction*
  - *Using a YOLO head to predict on three effective feature layers.*

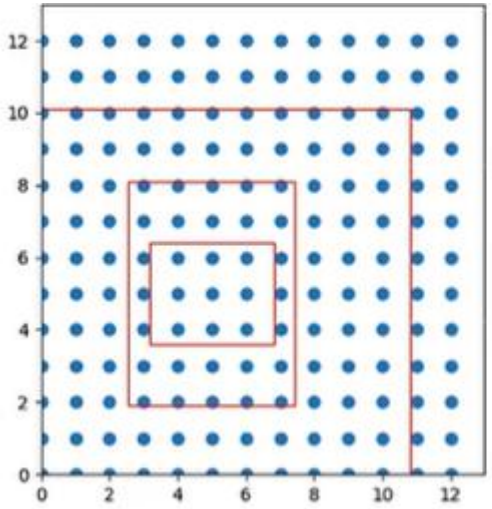


Approach

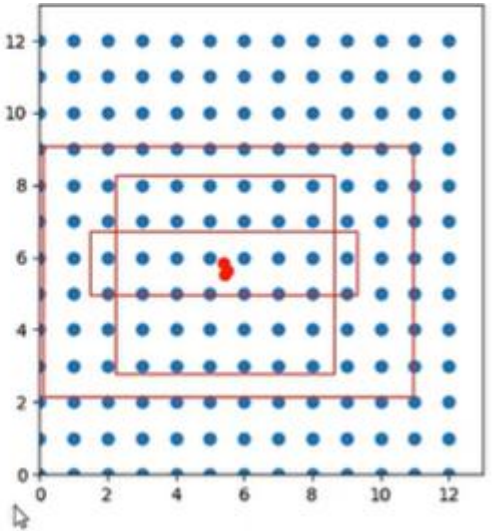
- ***Decoding part***

- *Center of the predicted boxes：*
  Obtained by adding the corresponding x_offset and y_offset to each grid point.

- *The width and height of the predicted boxes:*
  Calculated by combining the prior boxes with the given h and w values.



The prior box

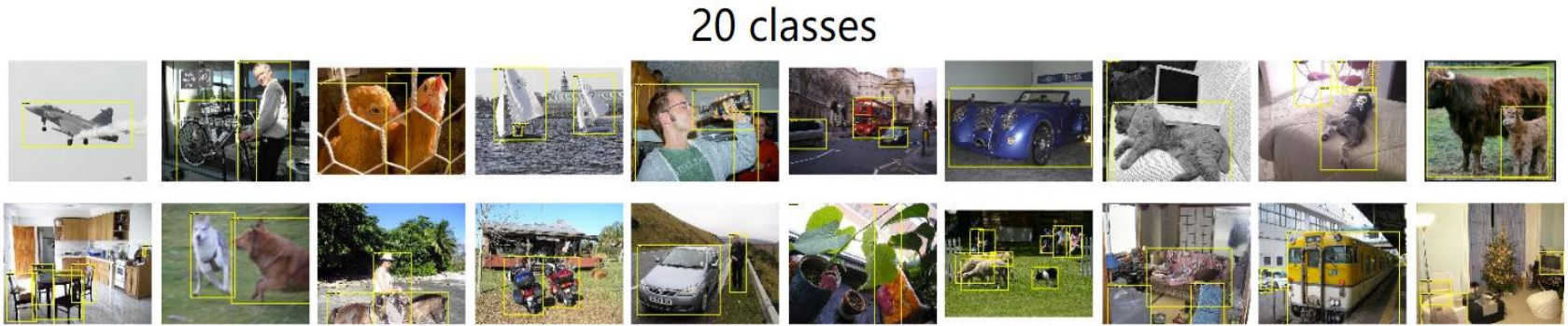


The predicted box

Dataset (s)

We use the VOC2007 dataset for training and testing.[Dataset link: here]

- *A benchmark dataset for object recognition and detection tasks in computer vision.*
- *Consists of images from 20 object categories, including people, animals, vehicles and so on.*
- *Contains 9,963 images for training and validation, and 4,952 images for testing.*
- *Each image in the VOC2007 dataset is annotated with object bounding boxes and class labels.*



Evaluation metric(s)

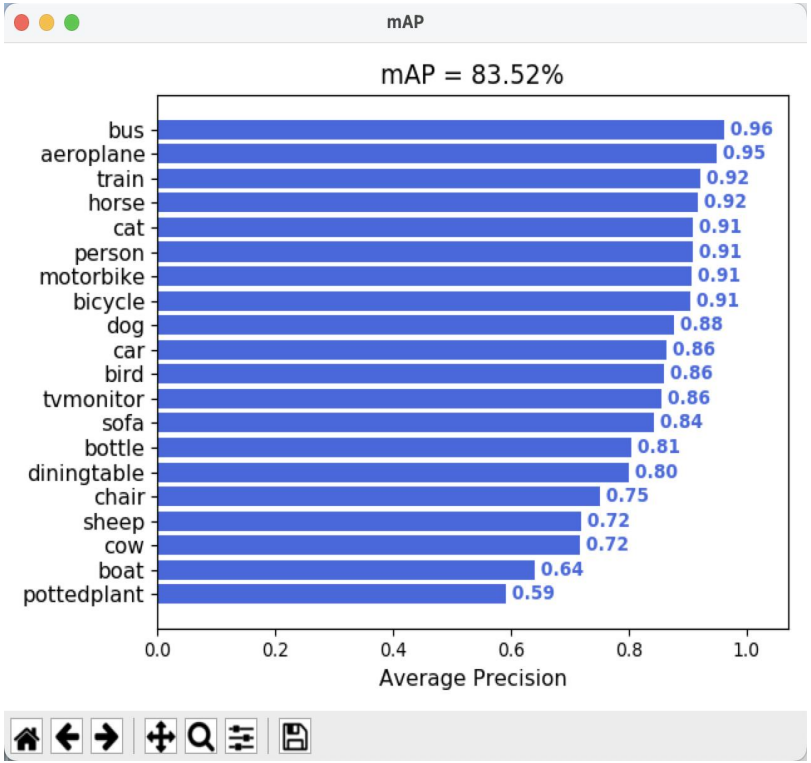
- TP: the number of predicted boxes whose IoU>0.5
- FP: the number of predicted boxes whose IoU<=0.5
- FN: the number of unpredicted boxes

Precision: TP/(TP+FP)

Recall: TP/(TP+FN)

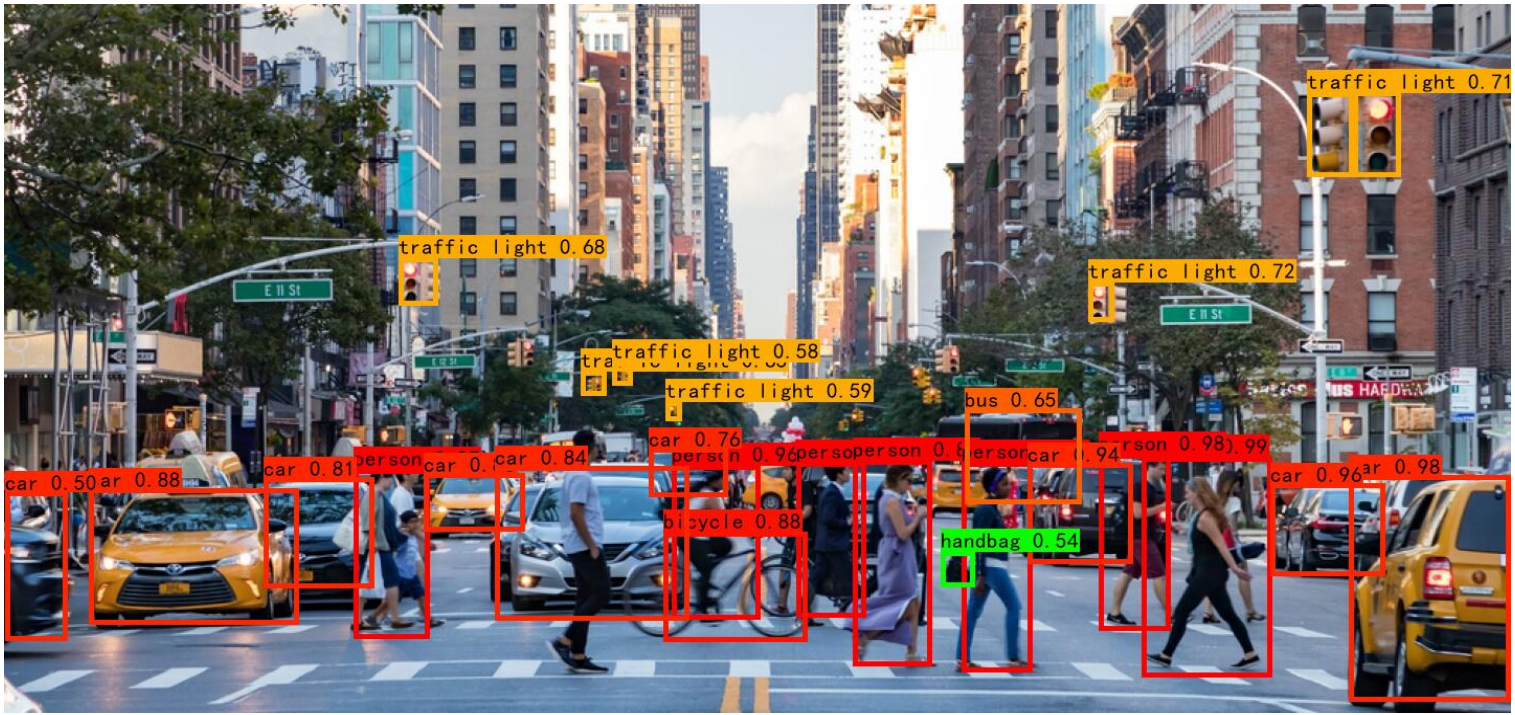
AP: The area under the Precision-Recall curve

mAP: the mean value of AP for each category



Results

- Successfully detecting every object in the image and
- Providing the bounding boxes around every object with the class label and the probability.



Conclusion

What we have done?

- *We successfully trained our model,preprocessed out data and tested out model on the dataset.*
- *We can successfully detected the objects on an image and labeled it at the same time*

What we learned?

- *We better understand its network structure ——Darknet-53; the loss function, and also the training strategy of this model.*
- *Overall, YOLOv3 is a powerful object detection algorithm for target detection.*