Milestone of object detection : YOLO

Lan Yuchen

Nov 2021

1 Introduction

You Only Look Once(YOLO)[3] is an object detection approach which regards object detection task as a regression task, proposed by Joseph Redmon in 2016[3] Unlike traditional approaches like sliding window, as its name, YOLO would not be repeated calculate image information, and thus has much higher speed in object detection. YOLO attracts huge attention from academia and industry by its high calculation speed and freshness, then YOLOv2[1], YOLOv3[2] were proposed, both have different improvements on loss function, net structure, etc. This article would introduce the background, idea of YOLO, and improvement from YOLO to YOLOv3

2 Start from classification

Object detection is a task to find the location of objects in a specified environment and decide their category(e.g. people, cat, cup, basketball...). Therefore, we need firstly learn what is classification problem is. Classification in computer vision can be defined progress using image or video as input, then output a one-hot vector which size is 1 * n(n is several classes). There is only one position in the one-hot vector is 1 while others are 0, the mere position represent the category of input data. As Shown below:

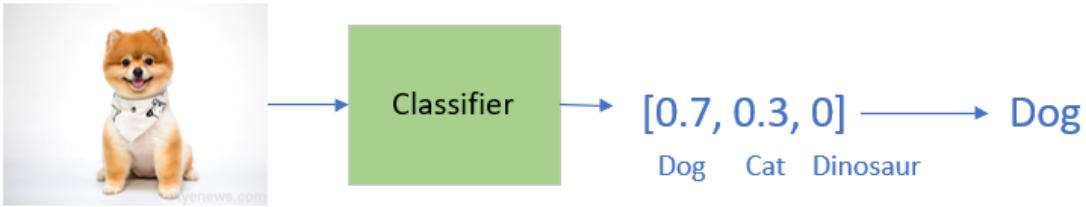

Figure 1: Example of classification

The performance of the classifier seems good, it know how the dog looks like. However, it may also become confused in some simple scenario even it clearly understand how a dog, cat, and dinosaur look like. One example just like the input shown below:

Figure 2: Dogs and cats

How this picture is classified, Cat? Dog? The classifier is confused. Some people may propose a solution quickly: "Just classify them respectively", and this is what the object detection model does, after applying the object detection model, the output would be like this:

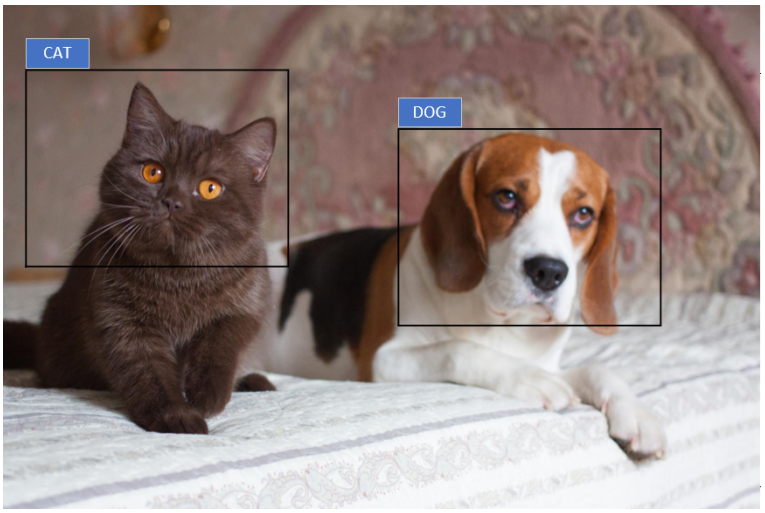

Figure 3: Object detection of dogs and cats

Every output frame can be defined to a vector, a common representation of this frame vector is [x,y,w,h], among them, x and y represent the coordination of the center point. w and h represent the width and height of this frame. Traditionally, the method of object detection is a sliding window, which can be regarded as a traversing classification. For each window would apply a binary classifier(Expected object/Background) and the output would be the frame with the highest possibility of the expected object and a one-hot vector of the category of the object inside frame

But there are three significant problem.

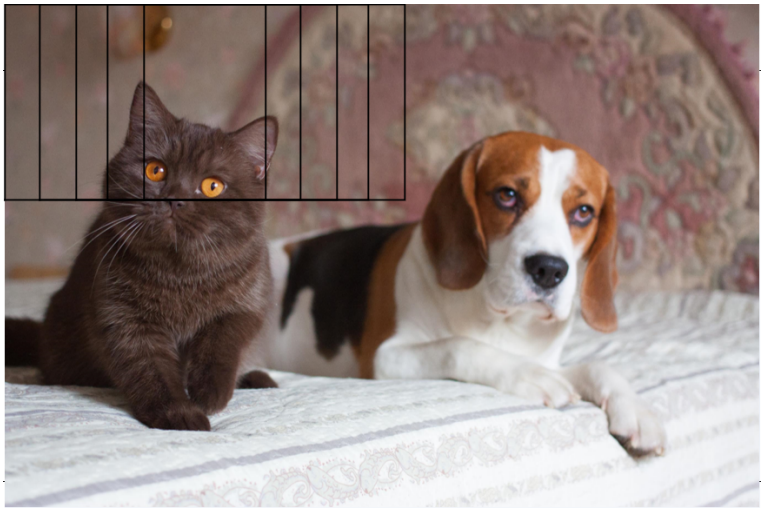

Figure 4: Sliding window object detection

1. Too much redundant calculation, especially when the object is relatively small compared with the background.
2. Still not solving the problem that cannot detect multiple objects.
3. Data is an imbalance, most of the time, the major part of the input is background.

So far, we learn how to do object detection with the idea of classification, which with several problems. Fortunately, Joseph Redmon makes improvement on this method, and proposed the milestone model, YOLO.

The basic thought of YOLO is to replace classification task with regression task, Joseph thinks that a vector [x, y, w, h, c] can represent the combination of output, among them, c is the confidence score of an object existing inside. YOLO would predict the location of the bounding box and the category relying on the learning ability of CNN. Also, YOLO would predict the classification of the category of the object inside the bounding box. Based on the definition of regression task, classification loss function and location loss function can be integrated to Sum of the Squared Errors(SSE)[3]. As shown below:

$$
\begin{aligned}
&\lambda_{\textbf{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] \\
&+ \lambda_{\textbf{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right] \\
&+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i \right)^2 \\
&+ \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i \right)^2 \\
&+ \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2 \qquad (3)
\end{aligned}
$$

Figure 5: YOLO loss function

The loss function consists of 4 parts. The first part is the loss of the center location. Among the equation, λ_{coord} represents the punctual coefficient of the prediction of the center point and $\mathbb{1}_{ij}^{obj}$ represent the existence of object, if there is an object its value would be one

$$\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right]$$

Figure 6: loss of the center location

The second part is the loss of width and height.

$$\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right]$$

Figure 7: loss of the width and height

The third part is the loss of confidence. λ_{noobj} is the punctual coefficient once the detected frame does not have object. The conidence score is calculated by $IOU * Pr(object)$
Among this definition, $Pr(object) = 1$ *if there is object else* 0 and IOU is defined as the intersection of predicted bounding box and the ground truth devided by the union of predicted bounding box and the ground truth.

$$+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i \right)^2$$

$$+ \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i \right)^2$$

Figure 8: Confidence loss

The last part is classification loss, using SSE instead of crossentropy. One thing to be noted is if there is no object, there would not be classification loss.

$$\sum_{i=0}^{S^2} \mathbb{1}_i^{\text{obj}} \sum_{c \in \text{classes}} \left(p_i(c) - \hat{p}_i(c)\right)^2$$

Figure 9: Classification loss

So far so good, the idea of YOLO has already been introduced, by converting the problem from classification to regression, there would be no more redundant calculation. Only look once!

Hold on. This model still cannot detect multiple objects. The output vector is still one bounding box. We may get the solution as soon as possible: Output multiple vectors. But how many vectors do we need? 1? 100? To solve this problem, Joseph chooses to let different cells to responsible for different bounding boxes [x,y,w,h,c]. Just as shown below:

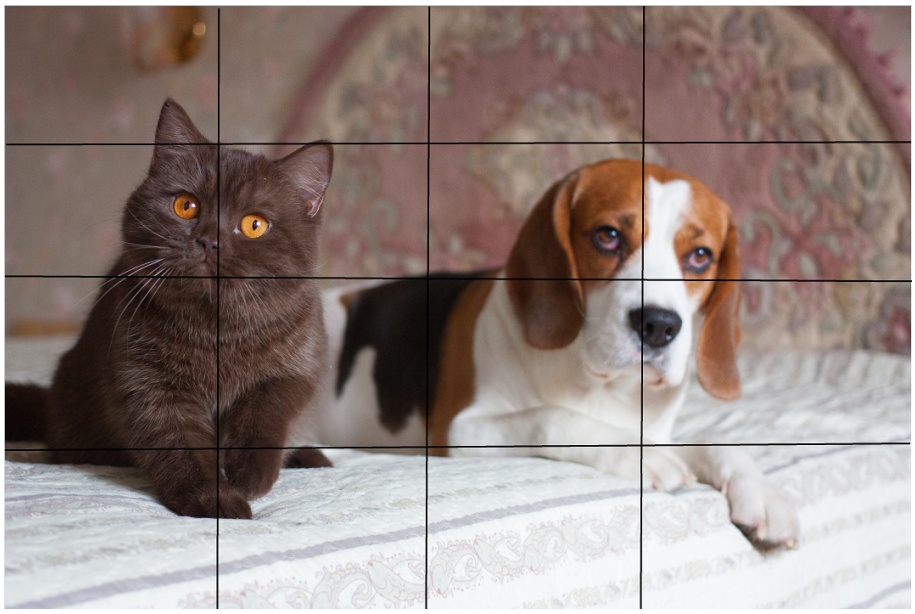

Figure 10: Multiple object

There is 16 cell, each cell would predict B bounding boxes [x,y,w,h,c] as candidates, and choose the one with the highest confidence score as the bounding box belonging to this cell. Since objects may occupy multiple cells, just like cat and dog in figure 10, multiple cell's bounding boxes may detect the same object, so how to choose? The answer is to choose the bounding box belonging to a cell that the object's center is located at[3]. This process may be a little confusing, how the cell knows it needs to predict the object which centers located in it? So let me introduce it in two stages. First, in the training stage, cells with the center of the object would be labeled with ground truth bounding box information, and thus

YOLO's cells are taught to predict the object in which the center is located. Then in the test stage, YOLO'cell would know that it needs to predict the object which centers located in it. Someone may still ask, training is not a one-size-fits-all, it is still possible that both two cells regard the object's center as located in itself. Fortunately, Joseph still has solution: Non-max suppression(NMS)[3]. This algorithm can be explained as 2 steps: 1. Choose the bounding box with the highest confidence score of 2. Calculating the IOU between this bounding box with other bounding boxes, once the result is higher than a threshold, suppress another bounding box.

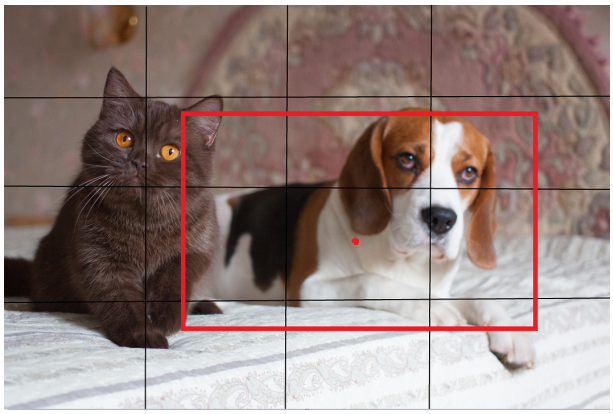

Figure 11: Dog detection

Very good! So far YOLO can detect multiple objects in one picture with much less time of calculation than a traditional approach, and this is YOLOv1, a milestone of object detection. However, everything just started, there are still 4 problems[3] of YOLO that need to solve.
1. Accuracy of predicting bounding box is not high enough
2. Many objects in the picture would be ignored
3. Cannot detect multiple objects located in the same cell
4. Weak in detecting small object

For solving these questions, one year after YOLOv1 was proposed, Joseph Redmond proposed YOLOv2[1], which is also named YOLO9000 since it can predict objects from 9000 categories. There are 10 improvements on YOLOv2 according to the paper[1], among them, The refreshing structure–Anchor and the new way of location prediction are practical to solve problems 1 and 2.

Let me introduce anchor first, simply, anchor is a predefined bounding box, just like shown below. improved[1].

We can notice that there is two frames, the black one is the ground truth bounding box, yes, the anchor does not equal the predefined bounding box. Then the red one is the anchor box. Based on these two box, the new way to predict location can be defined–calculating the offset between anchor box to ground truth box. The reason for choosing this way of location prediction is a conclusion that predicting offset is easier for neural networks, which come from faster-RCNN[4] and the performance of predicting bounding box is indeed improved. In other words, problem 1 was solved. Then moved to problem 2, to handle this question, we need 5 anchors... Hold on! Someone complain: "You make me confused! How do these anchors

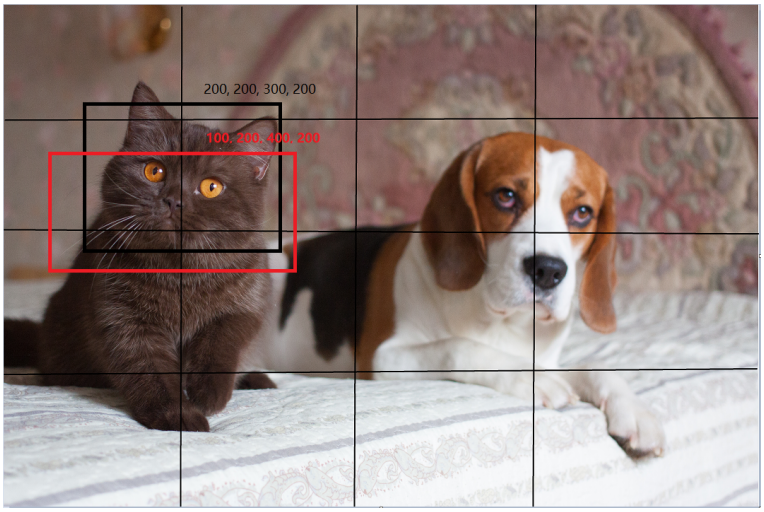

Figure 12: Anchor

come? I don't even know how to define one anchor but now we need to have 5..." All right, this is my fault, I would introduce how to get these anchors first. The answer is clustering. Joseph uses all information of the ground truth box to do clustering and get 5 prior bounding boxes(Anchors). During the process of object detection, each anchor would be responsible for predicting an object, thus can detect much more objects. Using the Figure of birds below as an example, if we have 16 cells and set 2 bounding boxes for each cell we can detect at most 32 birds, which is less than the actual number, but if we have 5 anchors we can detect at most of 90 objects.

Figure 13: birds

In fact, there is no need to require every anchors can detect object, so there is a confidence score threshold, we only select the box with confidence score higher then threshold.

This is YOLOv2, faster, better, and stronger[1]. Hold on, Stronger? We still have problem do not be solved! How to detect multiple objects in one cell? and how to detect small objects? Yes, YOLOv2 is not stronger enough, and Joseph also realize this point, so he proposed

YOLOv3, which also is the last model he proposed. In this model, these two problems have been solved. The performance is shown in two pictures below

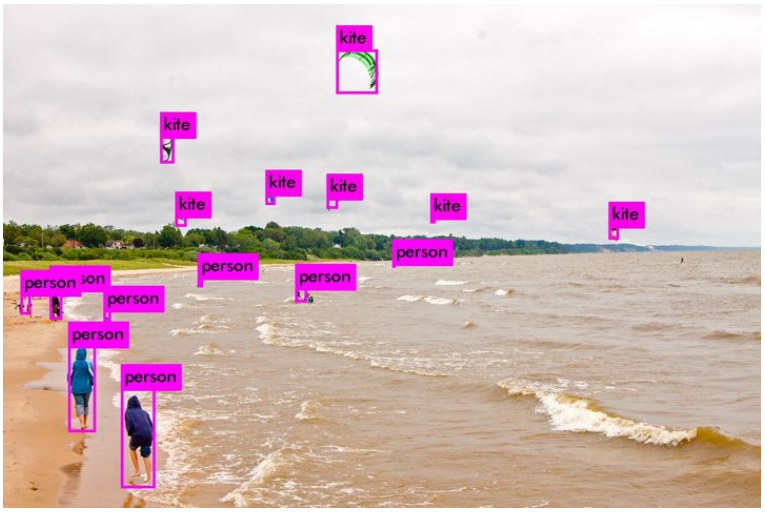

Figure 14: small objects

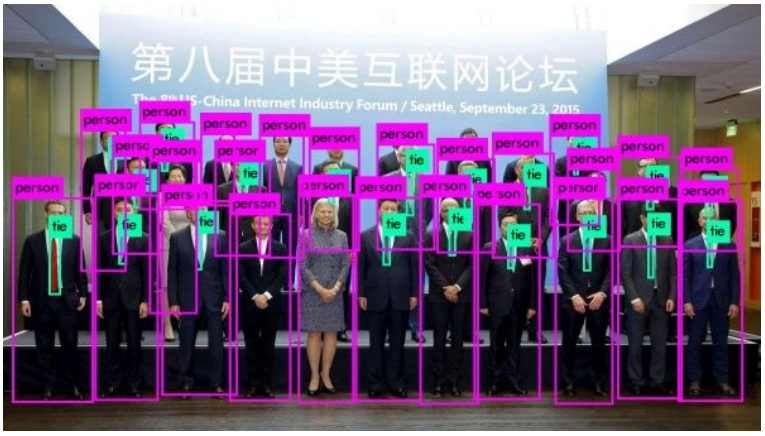

Figure 15: multiple object

The respectively solutions for these two problems are a new loss function and implement of multiple-scale feature maps[2]. Let me explain them further.

First, new loss function, Joseph use multiple sigmoid function to replace the softmax function, thus can detect different objects in one bounding box or detect different labels of one object. Second, multiple-scale feature maps, YOLOv3 append a down-sampling layer and two up-sampling layers to detect objects with different size. In other words, down-sampling layer would have bigger respective field, suitable for detecting large object, vice versa. Just like picture shown below.

Review the developing process of YOLO, We can find that the proposal and development of YOLO was come from some basic knowledge, like using sigmoid to conduct multiple classification, like converting object detection task to regression task. They were just another

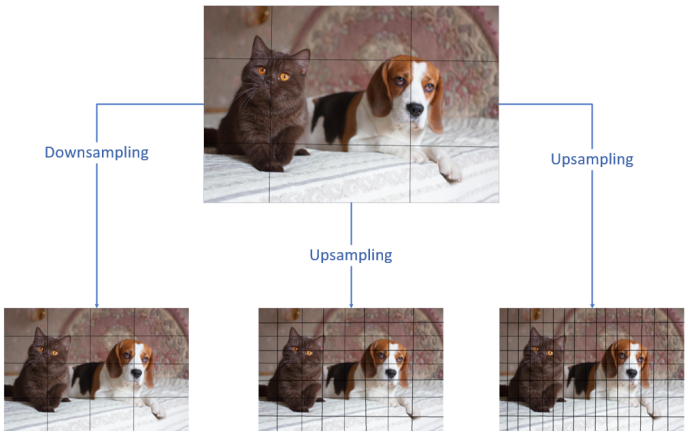

Figure 16: Multiscale

way of using existing method. All of us can be inspired from YOLO that basic knowledge is important.

References

[1] Joseph Redmon and Ali Farhadi. "YOLO9000: better, faster, stronger". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 7263–7271.

[2] Joseph Redmon and Ali Farhadi. "Yolov3: An incremental improvement". In: *arXiv preprint arXiv:1804.02767* (2018).

[3] Joseph Redmon et al. "You only look once: Unified, real-time object detection". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 779–788.

[4] Shaoqing Ren et al. "Faster r-cnn: Towards real-time object detection with region proposal networks". In: *Advances in neural information processing systems* 28 (2015), pp. 91–99.