Deep Models for Object Detection

This article attempts to introduce the concept of object detection as well several state-of-art deep models solving this problem to readers as simple as possible.

Introduction

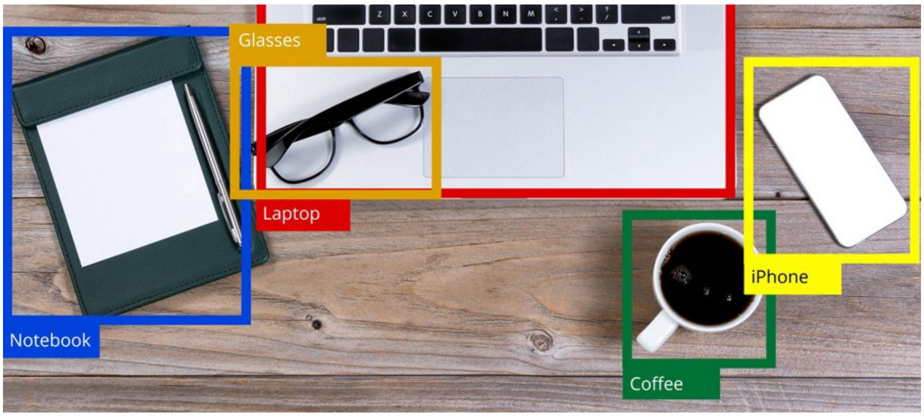
Computer Vision Brief Summary

Object detection is a classical task in computer vision. So, before we dive into details, let's take a look at the overall picture of computer vision and points out where we can place the object detection task in this picture. According to Wikipedia's Computer Vision entry, 'Computer vision is an interdisciplinary scientific field that deals with how computers can be made to gain high-level understanding from digital images or videos… It seeks automate tasks that the human visual system can do.' [5] For example, recognizing a dog from an image and circling it in the picture.

There are various tasks in computer vision filed, including recognition, motion analysis, scene reconstruction, image restoration, etc. The object detection task belongs to the recognition problem.

Object Detection Introduction

Basically, Object detection is a computer technology to detect semantic objects and use a bounding box to circle it in images. Object detection is a classical task in computer vision area and is fundamental for many image processing applications. For example, you have seen a picture of a lovely husky and want to find more pictures like this using an image search engine that takes pictures as input. To complete this task, the search engine has to find semantic objects as well as their categories in your pictures and the database so that it can match them accordingly. Another example is Face ID. To recognize your face, the software must find where your face is in the shot.

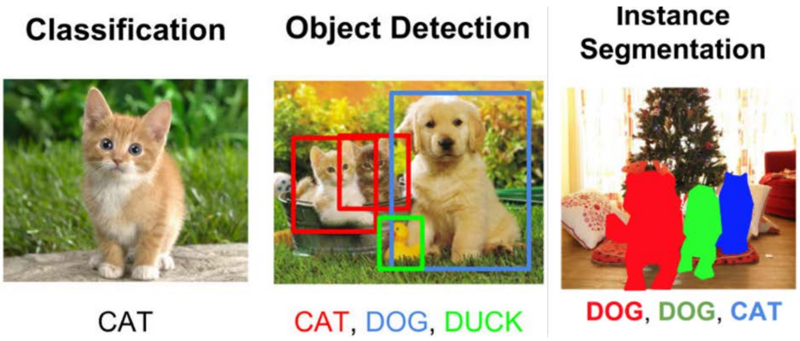


An object detection algorithm can find multiple semantic objects in one picture and circle the object with a bounding box as accurate as possible.

Pic from:

https://towardsdatascience.com/beginners-guide-to-object-detection-algorithms-6620fb31c375

Tasks similar to object detection include image classification and instance segmentation. From the picture below, we can see several differences between these three problems. The goal of image classification is to classify the picture according to its content with pre-defined categories. So, usually there is only one semantic object in the picture or the algorithm will get confused. The output of image classification algorithms is a category label. While in instance segmentation task, the algorithm not only recognizes and finds the position a semantic object, but also labels all pixels inside the object with its category.



Pic ensembled from various resources include:

https://towardsdatascience.com/object-detection-using-deep-learning-approaches-an-end-to-end-theoretical-perspective-4ca27eee8a9a

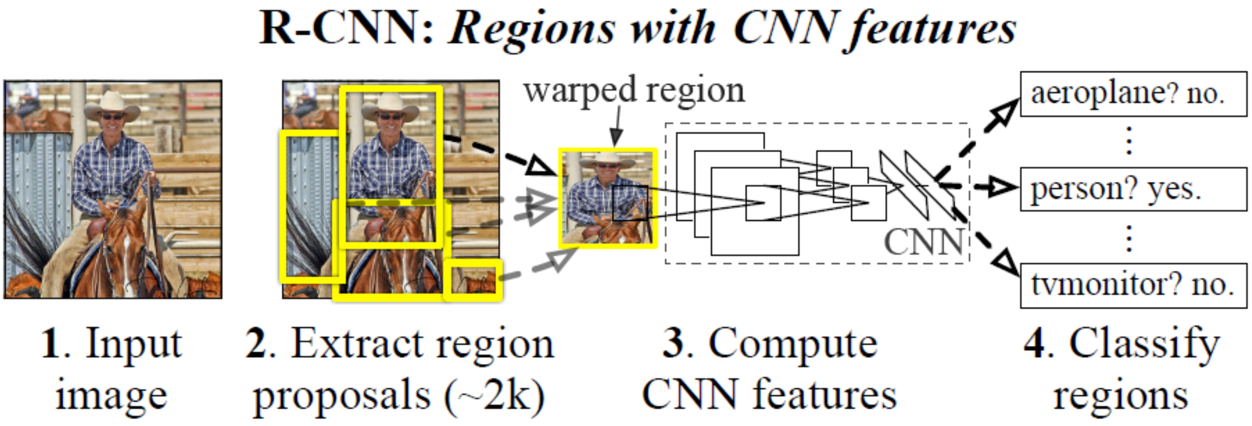
https://www.datacamp.com/community/tutorials/object-detection-guide

Different Models Analysis

Region Proposal Based Models

An intuitive solution to object detection task is using windows in various sizes to slide through the image and judge if the part inside the window contains an object as well as the corresponding category. Yet this method is computationally impractical, because there are too many windows needed to be tested. Thus, we use selective search algorithm to suggest much fewer windows (or say, bounding boxes) that may contain an object, and then use a classification model to confirm if there is an object in each window. More advanced models replace the selective search algorithm with a CNN to obtain both higher speed and accuracy. But let’s start with a basic model called R-CNN.

A. R-CNN



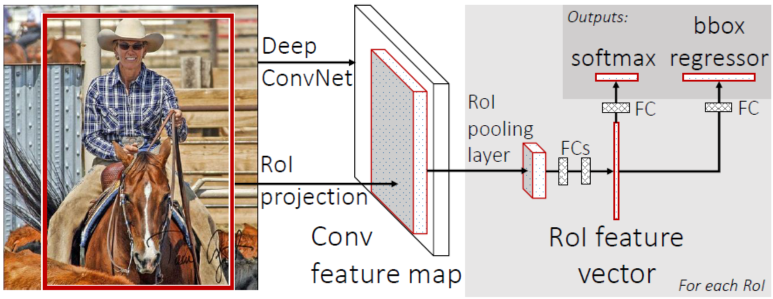
Pic from reference [1]

The picture above shows the work flow of R-CNN:

First, it applies selective search algorithm to the input image to find potential bounding boxes in various shape. Then, all of these boxes are warped to the same size to meet the input requirements of CNN. After that, the CNN model will take in these bounding boxes one by one and output the categories of them. To be specific, a ‘background’ category means there is no object in the bounding box while in other cases, the CNN should label the box with the category of the object contained.

B. Fast-R-CNN

R-CNN is still computationally expensive because around two thousand warped images need to be processed by the CNN before we can get the final result. An improved version of R-CNN is called Fast-R-CNN, whose work flow is shown below:

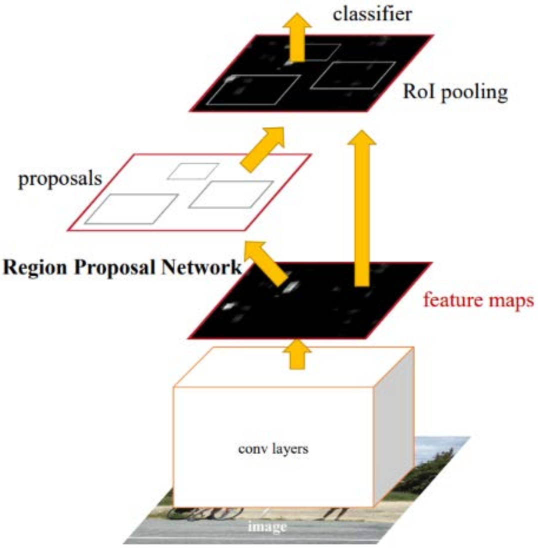


Pic from reference [2]

Instead of suggesting ROI (i.e. bounding boxes/windows) on the original image, Fast-R-CNN first input the whole image into CNN and applies selective search to the extracted features in a later stage. After combining these features with a pooling layer, categories are identified for each of these ROI. Then, ROIs in feature maps will be mapped to the origin image to get the bounding boxes. This method is much faster than R-CNN due to shared layers of CNN. In another word, only one feedforward process is needed for one image in contrast to two thousand times of feedforward in R-CNN.

C. Faster-R-CNN

Faster-R-CNN is even faster than Fast-R-CNN because it replaces the selective search algorithm with a CNN. Thus the 'bounding box suggesting' step and 'bounding box classification' step can be done at the same time, which can be seen from the picture below. According to [3], Faster-R-CNN only takes about 0.1 seconds to process an image using VGG16 and 2 seconds using ResNet101.



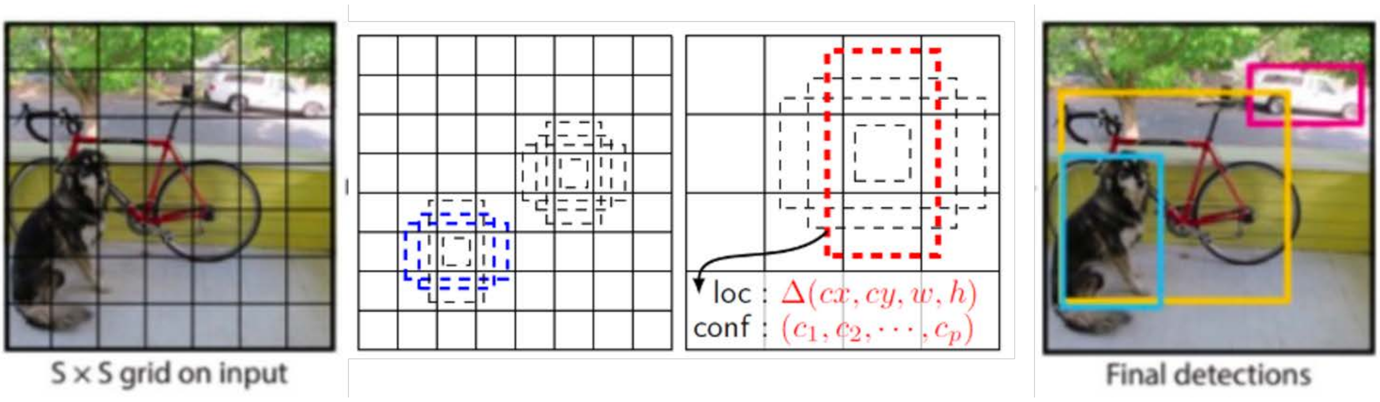
Pic from https://www.datacamp.com/community/tutorials/object-detection-guide

Except for the models mentioned above, there are also state-of-art region-proposal-based models like RFCN, FPN, SPP-Net, etc. If you are interested in more details, [3] will be a good review.

Regression Based Methods

YOLO and SSD

Unlike region-proposal-based methods, regression-based methods use gridding and anchors to reduce the amount of calculation. As the picture shown below, the image will be gridded in to several parts. Then various anchors (i.e. different sized windows at the center of each part) of the image will be feed into a large CNN, which will output not only the category of the anchor, but also four offsets for adjusting the position of the anchor. In the end, final detections can be gained based on these results. YOLO and SSD are all models based on this idea. They have quick speed because no bounding box proposal need to be made. It also because only a few grids need to be processed by the CNN. According to [3], only about 0.02 seconds is needed to process a single image, which means the regression-based methods can be applied to an online scenario.


S × S grid on input
loc : $\Delta(cx, cy, w, h)$
conf : $(c_1, c_2, \cdots, c_p)$
Final detections

Pic ensembled from various resources including https://towardsdatascience.com/beginners-guide-to-object-detection-algorithms-6620fb31c375 and https://towardsdatascience.com/review-ssd-single-shot-detector-object-detection-851a94607d11

Pros and Cons

The table below tells us about some advantages and disadvantages on each model, which can be concluded as follow:

1. Region proposal methods
   Pros:
   - Achieved state-of-art accuracy
   - Models like Faster-R-CNN takes only seconds of time to process an image.

   Cons:
   - Not fast enough for an online application
2. Regression-based methods
   Pros:
   - High speed
   - Not only can be used in real-time applications, but also can be online refined.

   Cons:
   - Lower accuracy compared with state-of-art region proposal methods.

	Running Time	Performance on COCO small – middle - large	Performance on VOC 2007 areo – bottle - cat	Performance on VOC 2012 areo – bottle - cat
RFCN	0.17	--	**92.3** – **75.2** – **95.8**	--
R-CNN (VGG16)	--	--	79.6 – 41.9 – 84.6	73.4 – 44.6 – 79.8
Fast-R-CNN	--	7.3 – 32.1 – 52.0	82.3 – 38.7 – 89.3	77.0 – 38.3 – 86.7
Faster-R-CNN (VGG16)	0.11	12.0 – 38.5 – 54.4	87.4 – 59.6 – 91.3	84.3 – **65.7** – **88.9**
YOLOv2	**0.03**	9.8 – 36.5 – 54.4	88.8 – 51.8 – 93.1	--
SSD300	**0.02**	9.6 – 37.6 – 56.5	91.0 – 55.4 – 93.4	80.9 – 57.6 – 88.6
SSD512	0.05	**14.0** – **43.5** – **59.0**	91.4 – 63.1 – 93.9	**86.6** – **66.3** – **89.1**

Resources: [3]

Recommendation for using these models

1. Choose the best model according what your task is. If your system needs to detect objects with high accuracy, it's better to use a region-based model like Faster-R-CNN or RFCN. On the other hand, if your system needs to do real time object detection, YOLO or SSD may be a better choice.
2. RFCN usually has the best accuracy among all region-based methods, followed by Faster-R-CNN, while SSD usually outperform YOLO in most datasets [3].
3. Various CNN structures can be used in object detection models like VGG and Resent. Choose one according to the computing power you have.
4. Also, decide how much training data you would like to use according to the computing power you have. It's definitely that the more data you use, the more accurate the model will be. Yet it takes longer time for the model to converge too.

Conclusion

This article first introduces several common tasks in computer vision and points out that the object detection plays a basic and important role in CV. After that, we discuss about the differences between image classification, object detection and instance segmentation, which helps readers to get a better understanding of the task.

The article also talks about two classes of methods for object detection, i.e. the region-based methods and regression-based methods. Advantages and disadvantages of these models are also included.

In the end, we have made some recommendations for readers who want to use these models to do object detection.

References

[1] Ross Girshick, Jeff Donahue, Trevor Darrell, Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014, pp. 580-587

[2] Ross Girshick. Fast R-CNN. The IEEE International Conference on Computer Vision (ICCV), 2015, pp. 1440-1448

[3] Z. Zhao, P. Zheng, S. Xu and X. Wu, "Object Detection with Deep Learning: A Review," in *IEEE Transactions on Neural Networks and Learning Systems*. doi: 10.1109/TNNLS.2018.2876865

[4] Stanford University School of Engineering (2017, Aug 11) [Video File]. Lecture 11 | Detection and Segmentation. Retrieved from: https://www.youtube.com/watch?v=nDPWywWRIRo&t=3970s

[5] 2019, Oct 12. Computer Vision [Wikipedia]. Retrieved from https://en.wikipedia.org/wiki/Computer_vision

[6] pkulzc, vivek rathod, Neal Wu (2019, Jul) [GitHub]. detection_model_zoo.md. Retrieved from https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc

[7] Sik-Ho Tsang (2018, Nov 3). Review: SSD – Single Shot Detector (Object Detection). Retrieved from https://towardsdatascience.com/review-ssd-single-shot-detector-object-detection-851a94607d11