International Journal of Trend in Scientific Research and Development (IJTSRD)
Volume: 3 | Issue: 3 | Mar-Apr 2019 Available Online: www.ijtsrd.com e-ISSN: 2456 - 6470

Object Detection: An Overview

P. Rajeshwari, P. Abhishek, P. Srikanth, T. Vinod

Anurag Group of Institutions, Telangana, India

How to cite this paper: P. Rajeshwari | P. Abhishek | P. Srikanth | T. Vinod "Object Detection: An Overview" Published in International Journal of Trend in Scientific Research and Development (ijtsrd), ISSN: 2456-6470, Volume-3 | Issue-3, April 2019, pp.1663-1665, URL: https://www.ijtsrd.com/papers/ijtsrd23422.pdf

IJTSRD23422

ABSTRACT

The goal of the project is to run an object detection algorithm on every frame of a video, thus allowing the algorithm to detect all the objects in it, including but not limited to: people, vehicles, animals etc. Object recognition and detection play a crucial role in computer vision and automated driving systems. We aim to design a system that does not compromise on performance or accuracy and provides real-time solutions. With the importance of computer vision growing with each passing day, models that deliver high-performance results are all the more dominant. Exponential growth in computing power as-well-as growing popularity in deep learning led to a stark increase in high-performance algorithms that solve real-world problems. Our model can be taken a step further, allowing the user the flexibility to detect only the objects that are needed at the moment despite being trained on a larger dataset.

KEYWORDS: *object detection, computer vision, deep learning, and neural networks.*

1. MOTIVATION

The motivation of building an Object Detection model is to provide solutions in the field of computer vision. The primary essence of object detection can be broken down into two parts: to locate objects in a scene (by drawing a bounding box around the object) and later to classify the objects (based on the classes it was trained on). A model that provides high-performance and accuracy to real-world data would offer solutions to pressing issues such as surveillance, face-detection and most of all, autonomous driving systems where any failure in a system may cause irreversible damage. Being able to classify only the object but not the location of the object accurately is not practical in the real world especially when the applications include autonomous vehicles, robotics and automation. Therefore, building an accurate model that provides high performance with low detection time is vital.

2. RELATED WORK

There are two deep learning based approaches for object detection: one-stage methods (YOLO – You Only Look Once, SSD – Single Shot Detection) and two-stage approaches (RCNN, Fast RCNN, Faster RCNN).

One-stage methods:

A. YOLO – You Only Look Once

It is a state-of-the-art, real-time object detection system which offers extreme levels of speed and accuracy. YOLO, as the name suggests, looks at the image only once, i.e., there is only a single network evaluation unlike the previous systems like the R-CNN approach which requires thousands of evaluations for a single image. This is the secret to the extreme speed of a YOLO model (almost 1000x faster than R-CNN and 100x faster than the Fast R-CNN model). In this approach, the model uses pre-defined set of boxes that look for objects in their regions. For the SxS grid cells drawn for each image, YOLO predicts X boundary boxes each with its own confidence score and each box can predict only one object. YOLO also generates Y conditional class probabilities (for the likeliness of each object class).

B. SSD – Single Shot Detection

Similar to YOLO, SSD's take only a single shot to detect all the classes in a scene that the model was trained on and therefore, like YOLO, is much faster than the traditional two-stage methods that require two shots (onefor generating region proposals and another for detecting objects of each proposal). SSD's implement techniques such as multi-scale features and default boxes which allow it to obtain similar levels of accuracy as that of a Faster R-CNN model using lower resolution images which further increases the speed of a Single Shot Detector.

SSD uses VGG16 to extrac feature maps from a scene and then uses a Conv4_3 convolution layer to detect objects from it.

Two-stage methods:

A. R-CNN

Instead of having to deal with a large number of regions, R-CNN uses divides a scene into 2000 regions called as region proposals which highly reduces the number of regions where detections need to be made. The 2000 regions are generated using a selective search algorithm. The selective search algorithm generally has 3 steps: to generate candidate regions, use greedy algorithm and recursively combine similar regions into larger regions; propose final candidate region proposals. The regions are then fed into an SVM to classify the presence of an object.

B. Fast R-CNN

The Fast R-CNN model was built to counter a few drawbacks of the previous R-CNN model. In this approach, similar to the previous approach, selective search is used to generate region proposals but the input image is fed to a CNN and a convolutional feature map is generated from it which is then used to identify the regions and combine them into larger squares by using a RoI pooling layer. A softmax layer is finally used to predict the class of the proposed region.

C. Faster R-CNN

Unlike R-CNN and Fast R-CNN, Faster R-CNN does not use Selective Search which is a slow process and instead came up with an approach to allow the network to learn the region proposals. Unlike Fast R-CNN where selective search algorithm is used on the feature map to identify region proposals, Faster R-CNN uses a separate network to predict the region proposals. The predicted proposals are then pooled into larger squares using the RoI pooling layer which is then finally used to classify the image.

3. PROBLEM STATEMENT

With the growing importance of computer vision and the need for autonomous vehicles, figuring out the problem of object detection is more crucial than ever. While image classification algorithms have developed a lot in the past few decades, image localization algorithms that offer extremely high accuracy and performance in real-time that and are able to detect a large number of classes in a short span of time are still rare. If such an algorithm is developed, the applications are seen not only in autonomous vehicles but also in robotics, surveillance, automation and analysis (people counting, traffic analysis, pedestrian detection).

4. SYSTEM SPECIFICATIONS

The system specifications on which the model was trained and evaluated are: Intel Core i7, 16 GB RAM, GPU – NVIDIA GeForce 1050 Ti.

5. LIMITATIONS

The primary reason for the need of an efficient model that provides high-performance in the real-world is the high cost of failure. The limitations of an object detection model are:

A. Occlusion

Occlusion is when an object is covered or not entirely visible, i.e., only a partial image of the object is visible. Occlusion, sometimes, stumps even the human brain so it is only natural that it causes trouble to even the best object detection models.

B. Image Illumination:

In the real-world, one cannot guarantee an uninterrupted supply of proper lamination in an image or a video. Any object detection would face trouble detecting objects in a poorly lit, dim environment.

C. Object Scale:

It may be difficult for the model to notice the difference between objects of various sizes. This is the Object Scale problem of computer vision. 4) View Point Variation

An object appears differently when looked at from various view-points. If the object is rotated or viewed at from a different angle, the entire perception appears different.

D. Clutter or Background Noise

In the real-world, finding a perfect scene is near impossible and the model has to constantly remove background noise in order for it to detect objects more accurately.

E. Large groups of small objects

Similar to humans, it is also difficult for an object detection model to accurately detect and track large groups of small objects in a scene.

6. METHODOLOGY

We use a Neural Network-based regression approach to detect objects and a classification algorithm to classify the objects that are detected. We implement the YOLO model that uses a fully convolutional neural network to generate S*S grids across the image, bounding boxes for each grid and the class probabilities for each of the bounding box. The entire process is streamlined into a regression problem thus allowing blinding speeds with extremely low latency.

The primary difference of our approach over other models is a global prediction system, i.e., unlike the previous approaches of sliding window and region proposal-based techniques, we view the image in its entirety in a single pass. The S*S grids that are generated for the image are responsible for detecting the object inside them, i.e., the grid cell that contains the center of the particular object is responsible for detecting the object. Every grid cell has to predict the bounding boxes for objects in them and the confidence scores for the boxes.

$$\text{Confidence} = \Pr(\text{Object}) * \text{IOU}$$

Where Pr(Object) is the probability that an object exists in the bounding box and IOU is the intersection over union.
To predict the conditional class probability, we use

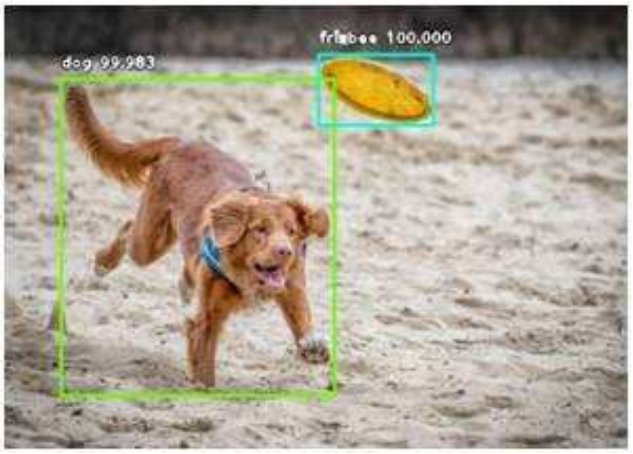
$$\Pr(\text{Classi}|\text{Object})*\Pr(\text{Object})*\text{IOU} = \Pr(\text{Classi})*\text{IOU}$$

7. EXPECTED RESULTS

Using the model, we expect to detect various objects that the model is trained to detect objects with extreme accuracy but also with a faster detection time. The models need to be improved to detect objects with more and more accuracy as even a small mistake in an autonomous vehicle would cause irreversible damage. Using our model, we expect to overcome or at least reduce the limitations mentioned above. The expectations with the system are:

1. Detect Objects with high accuracy and high confidence.
2. Detect objects despite being only partially visible (occlusion).
3. Detect objects when present in large groups irrespective of the size of the object.
4. Detect objects with low illumination.

8. RESULTS AND ANALYSIS



The model is able to predict the objects in the image correctly (with near perfect confidence in case of the dog and perfect confidence in case of the Frisbee). The model was trained on the Microsoft's COCO (Common Objects in COntext) dataset which consists of 80 classes (common everyday house-hold items).

The model accurately predicts almost all the people in the scene with the confidence scores of the objects that it detects, i.e., the probability of the object belonging to the class according to the model. The model shows above 80% confidence scores for people in the front and shows above 70% confidence score for people that are in behind and not shown properly. This shows that despite suffering from occlusion, the model predicts that a person is present in the scene with greater than 70% confidence. Not only does the model detect objects accurately when large groups of small objects are present, the model also confidently detects objects suffering from occlusion.

9. CONCLUSION

After analyzing the results, we can conclude that the model offers high performance and accuracy in the real-world and also overcomes most of the limitations mentioned earlier. The model offers satisfactory results to most, if not all of our expectations as mentioned in the EXPECTED RESULTS section.

Despite this, our model still needs to improve its accuracy in detecting objects

REFERENCE

[1] http://people.csail.mit.edu/klbouman/pw/papers_and_presentations/ObjectRecognition Detection-11-25-12.pdf

[2] http://ethesis.nitrkl.ac.in/4836/1/211CS104 9.pdf

[3] https://github.com/OlafenwaMoses/ImageAI/tree/master/imageai/Detection

[4] http://cocodataset.org/

[5] https://www.google.com/search?q=Retinanet&oq=Retinanet&aqs=chrome..69i57.1090j0j1&sourceid=chrome&ie=UTF-8

[6] https://www.analyticsvidhya.com/blog/2018/10/a-step-by-step-introduction-to-the-basic-object-detection-algorithms-part-1/

[7] https://www.tensorflow.org/lite/models/object_detection/overview

[8] https://www.hackerearth.com/blog/machine-learning/introduction-to-object-detection/

[9] https://www.edureka.co/blog/tensorflow-object-detection-tutorial/