INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact factor: 4.295
(Volume 4, Issue 2)
Available online at: www.ijariit.com

The comparison between various object detection algorithms

Chandramani Kumar
chandramanikumar871@gmail.com
Maharaja Agrasen Institute Of Technology. Rohini, Delhi

Kartik Chawla
kartik.chawla.167@gmail.com
Maharaja Agrasen Institute of Technology , Rohini, Delhi

Shreya Arora
shreyaarora@mait.ac.in
Maharaja Agrasen Institute of Technology , Rohini, Delhi

ABSTRACT

Object detection is a concept that has been handed over to the machines for some time now. It all started with pattern recognition back in the 1960's and the concept has been evolving ever since. Now it has come to such an extent that machines are successfully able to achieve real time object detection. But all this isn't done on its own and requires human help and instantiation in order to start working independently. Whenever there is a need for instantiation, there comes a need for an algorithm. Talking about algorithms, an algo is a finite series of steps which when performed lead to an outcome. Ever since the beginning of time, we have been using algorithms to do our daily chores. Ever since the beginning of machines, we have been using algorithms to program them and to define how they would react to various inputs. Similarly, when the concept of object detection came in, the series of development of algorithms began. There have been multiple developments over the past which have been able to do the job of object detection but the difference between these has always been of speed and accuracy as well as efficiency. Earlier algorithms were performed on still images that were captured out of cameras or taken out of video footages. Now, highly efficient algorithms have made it possible to provide real time object detection that can be performed on continuously moving video footages.

Object detection is not cakewalk and has a lot of challenges that it comes with. The goal to be achieved is overcome some challenges and reduce the effect of others to the bare minimum. It is true that there are tradeoffs involved in this process but based on the objective of the program, these tradeoffs can be easily managed. The main challenges that come up are mentioned below.

Keywords: *Object detection, Objects, R-CNN, Input, Output.*

1. Challenges that occur during Object Detection

a. Number of objects in an image

When an input is provided to an object recognition algorithm, the image is presented as data vectors and therefore, the number of objects that will be detected is unknown at the beginning. This means that the number of outputs is also unknown. Also, this tells us that there is going to be a need for calculations after the image has been processed, generally known as post-processing. This in turn adds to the complexity of the whole process. Such a problem is overcome using a sliding window where a sliding window takes various instances into view, uses the rational ones, discards some and merges some in order to get the final result.[1]

b. Size of the object(s)

Another big challenge is one of size. An image might have multiple objects and each of them aren't expected to be of the same size. Usually, we would want to look for variables that cover most part of the image but then there might be times when the small objects, that take up minute areas of an image need to be recognized too. This challenge might be overcome by using multiple sliding windows, each of different sizes.[2] This solution is quite simple but turns out to be inefficient as it requires the time to slide multiple windows each of different sizes. Each window takes a different amount of time to make a pass through the entire image and if the window is only a few pixels large, then the time taken grows exponentially.

The first framework that was devised to solve the problem of face detection, the Viola-Jones object detection framework, turned out to

be a milestone in object detection and is termed as the classical approach towards this concept as there are further improved methods to detect objects that are based on deep learning.[3]

The Viola-Jones framework was introduced in 2001 and was quite strict when it came to facial detection. It is a fact that computers cannot detect human faces and have to be given some direction as to how they will do that. So the constraint was that the human, in the image, must be facing the camera directly and the head should not be tilted in any direction for it to be detected.[4] This was not much practical but was still a bit of breakthrough as this was where other people needed to draw from. The deep learning approach is much different from this. In that approach, the vision of the computer is based on pre trained models. These pre trained models have a huge database of images on which they have already been trained and all the detection and recognition takes place through comparisons with these images. The categories that the detected objects are to be put in are numerous.[5] Further sections of this paper will be bent towards the different algorithms in deep learning approach.

2. DEEP LEARNING APPROACH TOWARDS OBJECT DETECTION

There have been multiple advancements in the field of object recognition using deep learning and of all those, one of the first ones was OverFeat which proposed a multi scale sliding window using Convolutional Neural Networks(CNN). But one of the more noticeable ones was Regions with CNN features or R-CNN.[6]

R-CNN

R-CNN was published at the UC Berkeley and its highlight was that it claimed to be an almost 50% improvement on the object detection challenge. The proposed theory was a three stage process.[7] The first stage was the extraction of possible objects using a region proposal method of which the most widely known was Selective Search. The second step involved the extraction of features from each region using a CNN. Finally, the last step was to classify each region using a Support Vector Machine (SVM). An SVM is a supervised learning model that is associated with learning algorithms in order to analyze and classify data.[8] These machine models have pre trained data which are classified into a category and when they're provided with new data, they make comparisons in order to classify the newly arrived data. The results obtained from R-CNN were great but there were various problems. In order to train it, proposals had to be generated for each training dataset, the CNN feature extraction had to be applied to each one of them and this took up a lot of space, after which the SVM had to be trained.[9]

Fast R-CNN

The next step up from R-CNN was Fast R-CNN which came in a year later. Just like R-CNN, Fast R-CNN used a Selective Search to generate object proposals but instead of extracting all of them independently and using SVM classifiers, it focused the CNN on the complete image and then made use of both ROI Pooling on the map of feature with a final feed forward network in order to carry on classification and regression.[10] This technique came out to be a lot faster and due to the RoI pooling, it became end to end differentiable and a lot easier to train. But there was still a big drawback to this method as it still relied on Selective Search.

You Only Look Once (YOLO)

Compared to other high end object detection algorithms like fast R-CNN, which perform detection on various region proposals and therefore end up performing prediction many times on different areas in an image, YOLO architecture is more like Fully Convolutional Neural Network (FCNN).[11] It is a state of the art object detection system. A single convolutional network is applied on the whole image. The network divides the image into a grid. This is further divided into regions and bounding boxes with probabilities of different region and category. These bounding boxes are then predicted by the weight of the probability due to comparisons using the SVM. As a single convolutional neural network is responsible for the prediction of multiple bounding boxes, the job is done much faster than other algorithms. YOLO runs on full images and directly increases performance. This model of object detection has several advantages over other methods of object detection.[12]

First, YOLO is very fast. There is no need for a complex pipeline. The neural network is simply run on the image when it comes as input. With this technique, it is easy to perform real time object detection with minimum latency.[13]

The next advantage is that YOLO detects the object and compares it globally as compared to other algorithms. This is because it looks at the entire image at a time rather than using the Sliding window that has eyes only a single region at a time. Other algorithms might mistake background differences or patches as objects because they don't see the complete picture at once. YOLO on the other hand, makes a lot less mistakes when it comes to background errors.[14]

In addition to the above two advantages, it can also be said that YOLO is adaptable and it more generalizable. It has been trained on natural images and cases and when it comes across an unexpected input, it is able to process it well rather than giving an error. This is why performs better than many of the object detection algorithms.

YOLO is a great algo when we talk about real time detection but as mentioned before in this paper, there are multiple tradeoffs as all the challenges can never be fully overcome and the motive is to reduce each of them to the least possible problematic value.

It imposes strong spatial constraints on bounding box predictions since each grid cell only predicts two boxes and can have only one class. This spatial constraint limits the number of nearby objects from being detected, like a flock of birds. Moreover, the model relies on predicting bounding boxes from objects, it has problems in generalizing objects in new or unusual aspect ratios. Moreover, the model struggles to predict small objects and usually ends up making small errors in small objects. A small error in a big object can be overlooked but even a minute error in a small object is quite significant.

3. SINGLE SHOT MULTIBOX DETECTOR

This is one of the latest proposals that have been made in the field of object detection. This technique is such that it detects objects in an image using a single deep neural network. This is similar to YOLO algo. Single Shot Multibox Detector or SSD, converts the output space of bounding boxes and makes it discrete such that it is turned into a set of bounding box priors over various aspect ratios and scales per feature map location. At the time of prediction, the network generates certainties that each prior relates to the objects of interest and generates adjustments to the previous to better match the object shape and boundary. Moreover, the network combines predictions from different feature maps with distinct resolutions to normally handle objects of different sizes. [15]

POINT OF DIFFERENCE	R-CNN	Fast R-CNN	YOLO
Extent of Real Time Object Detection	Low	Better than R-CNN but worse than YOLO	Best in its class
Latency during real time detection	High	Medium	Low
Type of Search	Selective Search	Selective Search	Single deep learning neural network
Number of Background errors	High	High	50% lower than the other two
Speed	Slow	Medium	Fast
Nearby object detection capability	Yes	Yes	No

Overall, we cannot yet declare any one of the object detection methods as the best because each of them can be used to serve a different purpose. Some algorithms are aimed towards successfully detecting all objects, small or large in the image in view. On the other hand, some are based on speed and are aimed towards performing in all situations. Therefore, some are good for real time object detection while others are useful for performing facial or still object detection. This tells us that the best algorithm to use depends on the need of the situation.

4. REFERENCES

[1] Antonio Gulia and Sujit Pal, “Deep Learning By Keras”, 2017, pp. 32-45.
[2] Tryo Labs, “Object Detection in the age of Deep Learning”, 2016, pp. 3-4.
[3] Ankit Sachan, “Guide to Deep Learning”, 2015, pp. 145-167.
[4] Pierre Sermanet, “Integrated Recognition, Localization using Convolutional Networks”, 2013.
[5] Jason Brownlee, “What is Deep Learning?”, 2016, pp. 1-4.
[6] Pedro Domingos, “A few useful things to know about Machine Learning”, 2012.
[7] Jianxin Wu, “Introduction to Convolutional Neural Networks”, 2017.
[8] Ross Girshick, Jeff Donahue, Trevor Darrell and Jitendra Malik, “Region Based Convolutional Neural Networks”, 2014.
[9] Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun, “Faster R-CNN: Towards Real Time Object Detection with Region Proposal Networks”, 2015.
[10] Ross Girshick, “Fast R-CNN”, 2015, pp. 3-7.
[11] Matt Zeiler, Rob Fergus, “Visualising and Understanding Neural Networks”, 2013.
[14] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi, “You Only Look Once: Unified, Real Time Object Detection”, 2015.
[15] Wei Liu, “Single Shot Multibox Detector”, 2015.

