Comparing Algorithms for Detecting People

Daniel Lindholm, Jacob Borg,
Nikolaj Thorsen Nielsen

December 10, 2020

Contents

1 Abstract

In this paper we show the difference in performance between three different person detection algorithms, seeing how big the difference between some of the 3 different methods we are using can be. Ranging from 1 to 16 second per image.

2 The Problem

In order to be able to do object tracking in real time on video, a fast algorithm is needed to process multiple images per second, and an algorithm that is good enough at detecting the needed object, in our case, a person. These two characteristics are important since they are needed in order to reliably track a moving object.

3 Benchmark

All the benchmarks are made on the same computer with an intel i5-4460 at 3.20 GHz and 8 GB ram, running the different algorithms on the same images and are all written in Python. The times have been calculated using pythons time.time()

4 Alternatives

Citing an internet link.[1]

4.1 Faster R-CNN (Region-Based Convolutional Neural Network)

It works using selective search to extract regions of interest in a picture, and then resizes these regions to a specific size, and runs these sub-images through a trained neural network to see if a known object can be found on any of them. https://www.analyticsvidhya.com/blog/2018/10/a-step-by-step-introduction-to-the-basic-obj
#:~:text=Instead%20of%20working%20on%20a,these%20boxes%20are%20called%
20regions

4.2 YOLO (You only look once)

What makes YOLO different from other algorithms in that it is only passed through a neural network once, and then the result is processed by a non-max suppression algorithm that makes sure the same object hasn't been found multiple times. https://medium.com/@ODSC/overview-of-the-yolo-object-detection-algorithm-7b52a745
~:text=YOLO%20sees%20the%20entire%20image,outperforms%20other%20top%
20detection%20methods

4.3 HOG (Histogram of oriented gradients)

HOG is used in computer vision, as a feature descriptor for object detection. First a gradient orientation is calculated for small areas of the image, then HOG is able to extract features from the gradient orientation data, this is done by HOG being able to detect edges and the direction of the edges. https://www.
analyticsvidhya.com/blog/2019/09/feature-engineering-images-introduction-hog-feature-descr

5 Findings

6 Conclusion

Yolo is by far the fastest of the different algorithms we tested. So if speed is the primary factor yolo would be a great choice. HOG is both slower and less precise, it's better for finding shapes, but because people can have different stances and angles, it makes it very difficult for this approach. RCNN was the only algorithm that found all the people in the pictures, but it did however also

have a false-positive. So if speed is not an issue and completeness is the most important thing, RCNN is a great choice.

Overall for tracking people in a stream of images, we found that YOLO's speed and accuracy was the best fit, as it didn't have any false positives and also was on average 6 times faster than RCNN.

7 Discussion and Future Work

The current code is run on the CPU, in the future it could be interesting to run some of these algorithms on a dedicated graphics card, for example you can use a newer nVidia card with integrated CUDA cores to speed up the processing time by as much as 10times????

Some of the different techniques these selected algorithms can be run using other weights that can make them faster, but less precise.

References

[1] Mark Senn. Using LaTeX for your thesis. `http://engineering.purdue.edu/~mark/puthesis`, 2009 (accessed February 3, 2014).

List of Figures

List of Tables