Object Detection and Tracking in Practice

Daniele Giuliani

Università degli Studi di Trento, Povo, Italy

Abstract. Object tracking is the problem of detecting the movement of objects on a canvas and computing their trajectory as they move across the image plane. In the last few years, many improvement have been made in this fields due to the advent of deep learning. In this paper we examine a practical example of object tracking on a video sequence from the *Multi Object Tracking* (*MOT*) challenge.

Keywords: Computer Vision · Object Detection · Object Tracking

1 Introduction

The problem of object tracking consists in detecting the presence of objects inside a frame and computing their position as they move across the canvas. A specific version of this problem is *Multi Object Tracking* (*MOT*) which poses several challenges, such as the presence of an unknown amount of objects to track (possibly very large), the fact that objects can leave and enter the scene at any moment and most importantly occlusions between different objects.

Many different techniques have been developed to approach this problem, during this paper we will present an approach based on the widely used *tracking-by-detection* paradigm. Our final goal is to be able to perform pedestrian tracking in a video sequence from the *MOT* challenge.

2 Methods

In our practical application we used the popular *tracking-by-detection* paradigm which considers the detection and tracking problem separately. In the following sections we will describe the detection part which was performed using a Faster R-CNN, and the tracking part, performed both with *intersection-over-union* (*IOU*) and *centroid-based* approaches.

2.1 Object Detection

Object detection was carried out by using a Faster Region-based Convolutional Neural Network (*Faster R-CNN*).

This approach represents an evolution in detection done using convolutional neural networks. The problem with using traditional CNN to perform detection is that objects might appear at different positions and scales inside the frame, hence creating the need to select a very large number of regions to check and making computation unfeasible.

To bypass this problem, the *R-CNN* method [1] has been proposed, which uses a Region Proposal Network (*RPN*) in order to extract a limited number of regions to check. The *Faster R-CNN* [3] approach improves on this method by merging the *RPN* and the *CNN* in a single architecture. Mechanisms such as *attention* and *ROI pooling* have also been introduced to perform region proposal and detection in an end-to-end manner, resulting in a significant increase of processing speed.

The specific implementation that we used, is pretrained on the Pascal VOC dataset and uses ResNet-50 as a backbone. This was done due to the unavailability of a discrete graphic card powerful enough to perform the training process in a reasonable amount of time. This network allows us to extract detection for multiple classes with a confidence level associated to each one of them. In our case we are interested only in pedestrian detection, as a confidence level we used is $Conf_{thres} = 0.5$.

2.2 Object Tracking

To perform tracking we explored two different approaches.

The first one is tracking based on the intersection-over-union (IOU) of bounding boxes as described in [4]. This approach computes on each frame the IOU of the active tracks with the detections of the current frame and associates each time the one with the highest value.

The second approach is centroid-based tracking, where on each frame the detections are associated to active tracks based on the distance of their centroid. We further improved this approach by adding a reidentification model based on the similarity of color histograms of the bounding boxes content.

3 Results

During this section we will described the results obtained both for detection and for the different methods of tracking.

3.1 Detection

The use of the Faster R-CNN to perform detection resulted very effective. We tried using different confidence threshold, but adopting values lower than 0.5 resulted in the generation of a lot of false positives. The processing speed of the network resulted reasonably high: by running the network on CPU, using a laptop with an Intel i7-2670QM, we were able to to extract all the bounding boxes overnight ($\approx 5 - 6$ hours of CPU time).

In the bounding boxes extracted there are still some false positives. The network has also some difficulty detecting pedestrians while partially occluded by the lamp post as we can see in Figure 1, where the bounding box for the pedestrian occluded is not shown because the network failed to detect it.

3.2 Tracking

Unfortunately the IOU-based tracking approach did not perform effectively. In fact the number of different tracks detected was well above 300 and the average track length was approximately 14 frame, which is unacceptable given the fact that the total number of track is 19 as specified by the ground truth data available. This behaviour is due to the low frame rate of the video, creating a large displacement for a single person in between two consecutive frames. This translates into a low IOU score between the detection and the active tracks resulting into new tracks begin constantly created.

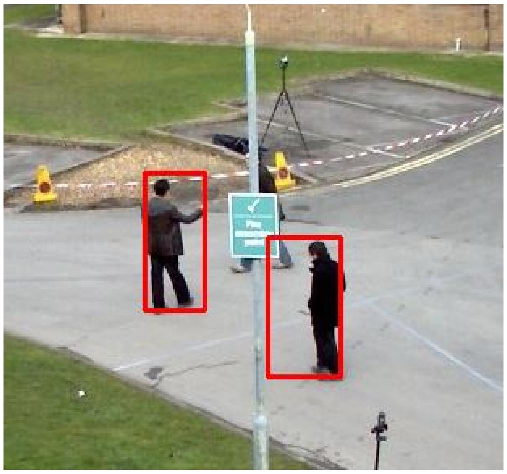

Fig. 1: Detected pedestrians by the Faster R-CNN in a frame where occlusion with other entities is present.

On the other hand, centroid-based tracking showed much better performance. The results are shown in Table 1, which presents a measure for the metrics commonly used when evaluating *MOT* algorithms. As we can see, the introduction of the reidentification model based on color histograms, improved significantly the *IDF1* measure, meaning that fewer identify switches were being performed. Even the MOTA score shows marginal improvement from the use of the color histograms. In the demo it can be observed that, most of the times, subjects are being reidentified even after passing behind the lamp post. Nevertheless this approach is far from perfect, the color histogram is a very approximative reidentification measure due to the fact that it is computed on all the content of the bounding box which includes both the person and the background. This means that the reidentification can still fail, especially when the subjects move to locations with very different background, such as from grass to asphalt and vice-versa, this results in bounding boxes, for the same track, jumping from one side of the image to the other.

4 Conclusions

Multiple Object Detection is a challenging task, even though achieving good result can be very difficult, we managed to create a working prototype which performs quite well and can still be improved upon. A possible improvement is to change the reidentification model with a better one, this could be done by implementing some background subtraction techniques in order to compute the color histogram only related to the person (and not to the background). Because of time constraints, we were not able to implement this improved version, but we are still satisfied with the performance of the proposed solution.

All the code base can be found in the Github repository [5], correlated with all the instruction necessary to execute the prototype. A demo of the prototype can also be found at: `https://drive.google.com/file/d/1dzz9wLGeHRYAbTfVUgLq5FORZrrtwGhW`

	IDF1	*IDP*	*IDR*	*Rcll*	*FP*	*FN*	*IDs*	*MOTA*
centroid	54.2%	53.6%	54.8%	94.1%	372	273	86	84.3%
centroid + color hist.	60.4%	61.0%	59.7%	92.2%	265	364	68	85.0%

Table 1: Scores obtained by centroid-based tracking with and without the use of of color histograms

References

1. Ross B. Girshick and Jeff Donahue and Trevor Darrell and Jitendra Malik, Rich feature hierarchies for accurate object detection and semantic segmentation (2013). http://arxiv.org/abs/1311.2524
2. Ross B. Girshick, Fast R-CNN (2015). http://arxiv.org/abs/1504.08083
3. Shaoqing Ren and Kaiming He and Ross B. Girshick and Jian Sun, Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks (2015). http://arxiv.org/abs/1506.01497
4. Bochinski Erik and Senst Tobias and Sikora Thomas, Extending IOU Based Multi-Object Tracking by Visual (2018). https://doi.org/10.1109/AVSS.2018.8639144
5. https://github.com/daniele122008/cv2assignment2