engineering proceedings

MDPI

Proceeding Paper

An Overview of Object Detection and Tracking Algorithms †

Kehao Du * and Alexander Bobkov

Department of Informatics and Control Systems, Bauman Moscow State Technical University, 105005 Moscow, Russia; alexander.bobkov@bmstu.ru

* Correspondence: duhuhaizi@gmail.com; Tel.: +7-925-793-0588

† Presented at the 15th International Conference "Intelligent Systems" (INTELS'22), Moscow, Russia, 14–16 December 2022.

Abstract: With the development of information technology, the vision-based detection and tracking of moving objects is gradually penetrating into all aspects of people's lives, and its importance is becoming more prominent, attracting more and more scientists and research institutions at home and abroad to participate in research in this field. With in-depth research into vision-based object detection and tracking, various superior algorithms have appeared in recent years. In this article, we attempt to compare some of the classic algorithms in this area of detection and tracking that have appeared recently. This article examines and summarizes two areas: the detection and the tracking of moving objects. First, we divide object detection into one-stage algorithms and two-stage algorithms depending on whether a region proposal should be generated, and we accordingly outline some commonly used object detection algorithms. Second, we separate object tracking into the KCF and SORT algorithms according to the differences in the underlying algorithms.

Keywords: region proposal; R-CNN; Fast R-CNN; Faster R-CNN; YOLO; MOSSE; KCF; SORT

1. Object Detection Algorithms

Object detection algorithms can be divided into one-stage algorithms and two-stage algorithms. Two-stage object detection algorithms appeared earlier, so we start with them.

1.1. Two-Stage Object Detection Algorithms

The task of object detection in an image usually involves drawing bounding boxes and labels for each object, i.e., classification and localization for each object, not merely the main object.

If we use a sliding window for image classification and object localization, we would need to apply convolutional neural networks to many different objects in the image. Since convolutional neural networks will recognize every object in the image as either an object or the background, we would need to use convolutional neural networks in a large number of places and scales, but this is computationally intensive.

To solve this problem, neural network researchers propose using the concept of regions so that we can identify "blobs" of areas in an image that contain objects, which can be performed much faster. The first model is regions with CNN features (R-CNN) [1], and the principles of its algorithm are shown in Figure 1.

In R-CNN, the input image is first scanned for possible features using a selective search algorithm, resulting in roughly 2000 region proposals; then, we run a convolutional neural network on these region proposals; finally, the output of each convolutional neural network is fed into a support vector machine (SVM), which uses linear regression to narrow the feature's bounding box.

Essentially, we transform object detection into an image classification task. However, there are also several problems: the learning rate is low, it requires a large amount of disk space, and the inference speed is also low.

check for updates

Citation: Du, K.; Bobkov, A. An Overview of Object Detection and Tracking Algorithms. *Eng. Proc.* **2023**, *33*, 22. https://doi.org/10.3390/engproc2023033022

Academic Editors: Askhat Diveev, Ivan Zelinka, Arutun Avetisyan and Alexander Ilin

Published: 13 June 2023

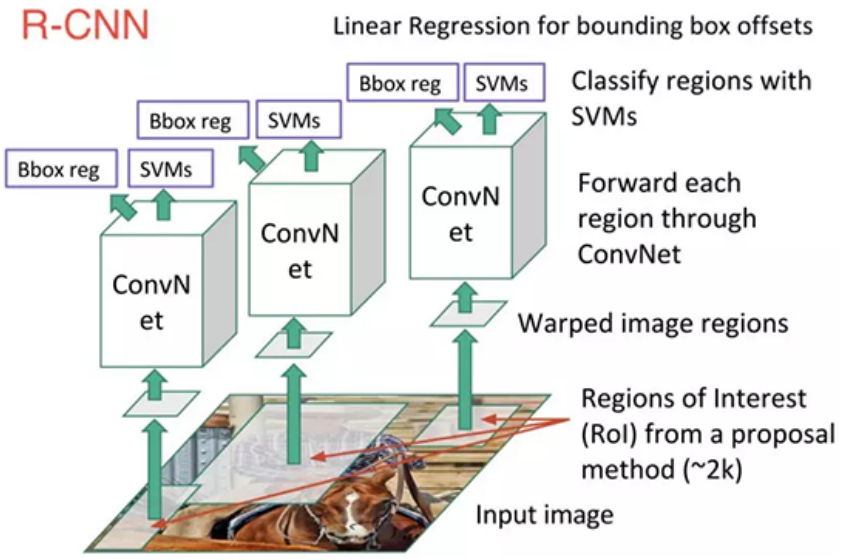

Figure 1. R-CNN architecture.

The first updated version of R-CNN is Fast R-CNN [2], which significantly improves the average precision through two improvements, which are shown in Figure 2.

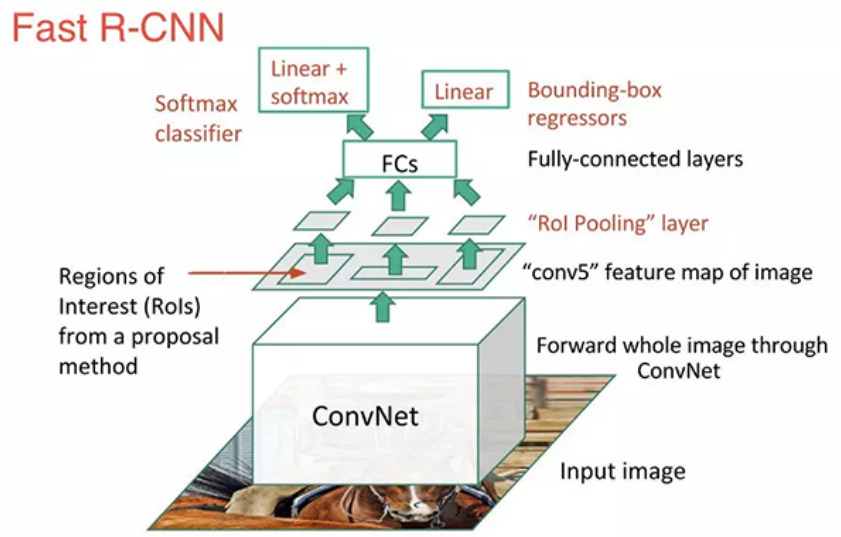

Figure 2. Fast R-CNN architecture.

Feature extraction is performed before creating a proposal of regions, so the convolutional neural network can only be run once for the entire image. We extend the neural network used for prediction by replacing the SVM with a SoftMax layer, instead of creating a new model.

Fast R-CNN is much faster than R-CNN because it only trains one CNN per image. However, the selective search algorithm still requires a long time to generate region proposals.

Faster R-CNN [3] is a typical case of object detection based on deep learning. The algorithm replaces the slow selective search algorithm with a fast neural network: by inserting a region proposal network (RPN), it predicts proposals based on features. The RPN decides "where" to look, and this reduces the amount of computation for the entire inference process. Once we have region proposals, we feed them directly into Fast R-CNN. Moreover, we add a pooling layer, several fully connected layers, a SoftMax classification layer, and a bounding box regressor, which are shown in Figure 3. In conclusion, Faster R-CNN is faster and more accurate.

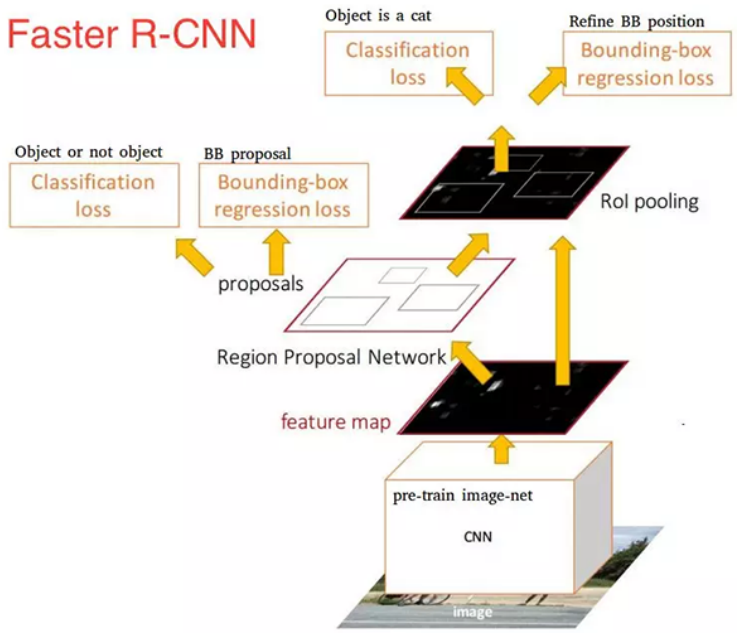

Figure 3. Faster R-CNN architecture.

1.2. One-Stage Object Detection Algorithms

Although Faster R-CNN has advanced greatly in terms of computational speed, it cannot meet the real-time detection requirements, so some researchers have proposed a regression-based method to directly regress the position and category of the object from the image. Two classic methods are YOLO [4] and SSD [5]. This article only discusses YOLO and its one-stage algorithm.

Different from the two-stage (two stages) detection algorithms represented by the R-CNN series, YOLO discards the region proposals and directly completes feature extraction, bounding box regression, and classification in the same non-branched convolutional network, which makes the network structure simpler and the detection speed almost 10 times higher than that of Faster R-CNN. This allows deep learning object detection algorithms to meet the needs of real-time detection tasks with improved processing power.

The specific steps of the algorithm are shown in Figure 4. We divide the input image into an S × S grid; each grid is responsible for determining what the objects in the grid are. We output the parameter information of the rectangles around the objects and the confidence score. The confidence score here refers to which features are contained in the grid and the predictive accuracy of each feature.

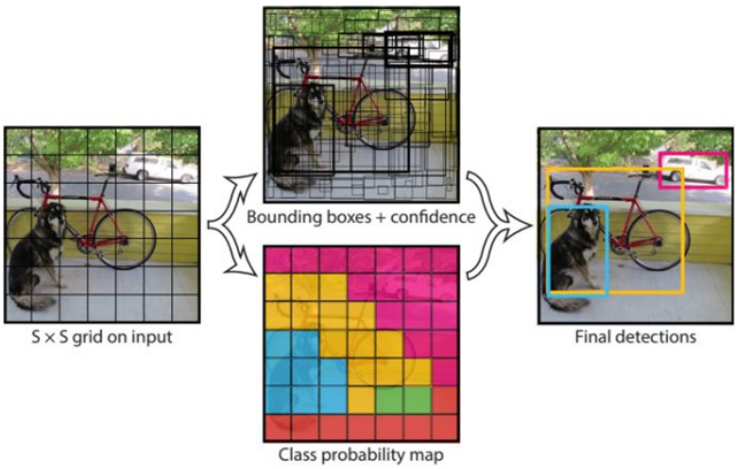

Figure 4. You Only Look Once (YOLO) model.

Since YOLO only looks at the image once, we do not use sliding windows. The feature map extracted after image convolution is divided into S × S blocks and then each block is classified with a superior classification model and each mesh is processed using non-maximum suppression to remove overlapping blocks; finally, we obtain our result. It can be seen that the whole process is very simple: there is no need for region proposals to find the object, and direct regression completes the determination of the position and category.

2. Object Tracking Algorithms

The next category concerns object tracking algorithms. It is generally considered that visual object tracking falls into two categories: generative model methods and discriminative model methods. Currently, the most popular is the discriminative method, also called tracking by detection. All the algorithms presented in this article are related to tracking by detection.

Apart from classical tracking algorithms, object tracking algorithms are divided into two categories: correlation-filter-based tracking (MOSSE) [6] and deep-learning-based tracking algorithms (SORT) [7].

The notion of correlation first appeared in the field of signal processing to measure the relationship between signals, including autocorrelation and cross-correlation. Researchers have introduced cross-correlation into the field of computer vision to analyze the relationships between factors; it is used to measure the degree of local matching of different images in a certain area. Correlation filtering is a method of searching for objects in an image frame based on the principle of cross-correlation.

Before the advent of correlation filtering and deep learning, the traditional visual tracking algorithm was slow in research and had poor tracking performance. MOSSE contributes to the development of the correlation filter tracking algorithm and also reflects the potential of correlation filter technology in the tracking domain for the first time. The frame rate of MOSSE is shown in Table 1.

Table 1. Compares the frame rates of MOSSE and other algorithm.

Algorithm	Frame Rate	CPU
Frag Track	realtime	Unknown
GBDL	realtime	3.4 Ghz Pent. 4
IVTL	7.5 fps	2.8 Ghz CPU
MILTrack	25 fps	Core 2 Quad
MOSSE Filters	669 fps	2.4 Ghz Core 2 Duo

Based on the principle of correlation filtering, aiming to measure the similarity of local pixels, MOSSE develops a filter that can realize the sum of the least squares of errors. After tracking the initialization of the first frame of a video sequence, MOSSE can generate a relatively stable filter to use a matching pattern to measure the similarity in subsequent video images. The tracking speed can reach 669 fps when limited; with the advantage of a high tracking speed, the method has good reliability. MOSSE uses the Fast Fourier Transform (FFT) for image and filter processing, converting the convolution of images and filters in the frequency domain into a mathematical product operation. After filtering for image detection, the point with the highest cross-correlation value, i.e., the center point of the object, will be found.

Considering that the correlation filter algorithm using raw pixels for feature extraction is too simple, the accuracy of the algorithm is not high. An improved CS correlation filter algorithm is proposed, also based on correlation filtering, which additionally provides higher accuracy while maintaining the speed advantage of correlation filtering. KCF [8] is an improvement of CSK [9], and CSK is also an improvement and addition based on MOSSE.

However, in terms of tracking, in order to reduce the computational complexity of the model, KCF proposes a cyclic matrix to reduce the computational complexity of the

cyclic shifts, and the Fourier transform is used to switch the calculation process between the frequency domain and the spatial domain, so that matrix multiplication with higher computational complexity becomes computation point multiplication with less computational complexity. During the KCF tracking process, the detection model also learns to measure the similarity between the predicted object and the tracking object. For each object predicted by the tracking model, the detection model first measures the similarity between the predicted object and the tracking object, and then decides whether to update the tracking track.

Another algorithm, SORT, differs from the correlation filter algorithm in that it does not use any signs of the tracked object during tracking, but only uses the position and size of the detection frames to estimate the movement and match the data. It does not implement any re-detection algorithm, focusing instead on frame-to-frame comparisons.

SORT mainly consists of three parts: object detection, a Kalman filter, and the Hungarian algorithm. Object detection mainly uses algorithms such as YOLO and R-CNN. Kalman filtering can identify the "optimal" estimate using the value predicted by the mathematical model and the measured value of the observation to find the "optimum" estimate (optimal here refers to the smallest standard error).

The Hungarian algorithm is a data association algorithm; in essence, the tracking algorithm should solve the data association problem. Suppose that there are two sets S and T, and set S has m elements and set T has n elements. The Hungarian algorithm seeks to match elements in S with elements in T in pairs (although they may not be the same). Combined with tracking scenario S and T, we have frame t and frame $t-1$. The task of the Hungarian algorithm is to match frame t with frame $t-1$.

SORT, as reflected in its name, is simple and can correspond to real time. On the one hand, its principle is straightforward and easy to implement; on the other hand, it is extremely fast due to its simplicity, as shown in Figure 5. At the same time, thanks to the introduction of object detection technology, the tracking accuracy is also significantly improved, which means that it can achieve a good compromise between accuracy and speed.

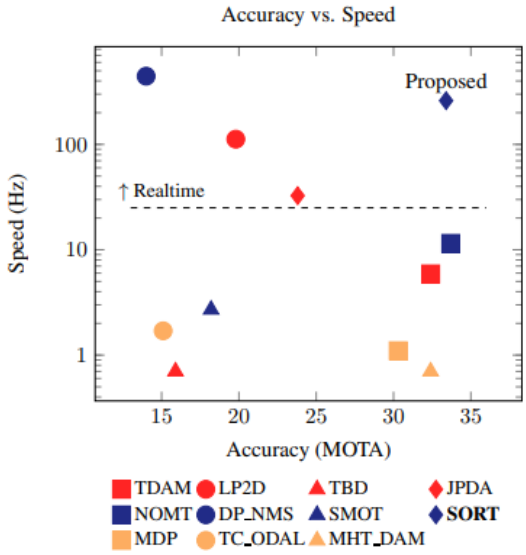

Figure 5. Benchmark performance of SORT.

3. Conclusions

In this article, modern algorithms for object detection and tracking are considered. Object detection is presented in terms of one-stage algorithms and two-stage algorithms. Object tracking via the SORT and KCF algorithms is introduced. Algorithms for object detection and tracking based on different concepts are compared, the unique and advantageous qualities of each concept are identified, and a promising direction for the development of

future detection and tracking algorithms is explored. For example, self-driving cars always require improvements in processing speed and accuracy.

Author Contributions: Conceptualization, A.B.; methodology, A.B.; software, K.D.; validation, K.D.; writing—original draft preparation, K.D.; writing—review and editing, K.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the 2014 Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
2. Girshick, R. Fast R-CNN. In Proceedings of the 2015 International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
3. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015; Volume 28.
4. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the 2016 Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
5. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single shot multibox detector. In Proceedings of the 2016 European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 11–14 October 2016; pp. 21–37.
6. Bolme, D.S.; Beveridge, J.R.; Draper, B.A.; Lui, Y.M. Visual object tracking using adaptive correlation filters. In Proceedings of the 2010 Computer Vision and Pattern Recognition (CVPR), San Francisco, CA, USA, 13–18 June 2010; pp. 2544–2550.
7. Bewley, A.; Ge, Z.; Ott, L.; Ramos, F.; Upcroft, B. Simple online and realtime tracking. In Proceedings of the 2016 International Conference on Image Processing (ICIP), Phoenix, AZ, USA, 25–28 September 2016; pp. 3464–3468.
8. Henriques, J.F.; Caseiro, R.; Martins, P.; Batista, J. High-speed tracking with kernelized correlation filters. *IEEE Trans. Pattern Anal. Mach. Intell.* **2014**, *37*, 583–596. [CrossRef] [PubMed]
9. Henriques, J.F.; Caseiro, R.; Martins, P.; Batista, J. Exploiting the circulant structure of tracking-by-detection with kernels. In Proceedings of the 2012 European Conference on Computer Vision (ECCV), Florence, Italy, 7–13 October 2012; pp. 702–715.