JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

All about Object Detection & Tracking

Devika Rani Roy, Shivani Gupta, Akshada Golhe & Krishna Khandelwal

Information Technology,

K.C. College of Engineering and Management Studies and Research.

Thane, India.

ABSTRACT: Using modern computer vision technology, this research sought to construct an object identification and tracking system. The project results in a working tracking system. It is a fusion of optical and current infrared technologies that can be used for unsupervised monitoring or semi-autonomous control. It is stable and usable as a stand-alone system, but it could also be readily incorporated into a wide variety of sizes.

KEYWORDS: Multiple Object Tracking, DeepSORT, YOLOv4, Target tracking, TensorFlow, Image Processing, Image Segmentation.

INTRODUCTION

During the last few years, there has been a rapid and successful expansion of computer vision research. Parts of this success have come from adopting and adapting machine learning methods, while others are from the development of new representations and models for specific computer vision problems or from the development of efficient solutions. One area that has attained great progress is object detection. The present works give a perspective on object detection research.

Object detection and tracking is an important challenging task within the area of Computer Vision that try to detect, recognize and track objects over a sequence of images called video. It helps to understand, and describe object behaviour instead of monitoring computers by human operators. It aims to locate moving objects in a video file or surveillance camera. Object tracking is the process of locating an object or multiple objects using a single camera, multiple cameras or a given video fileThe inventiononhigh-qualitylity the imaging sensor, quality of the image and resolution of the image are improved, and the exponential increment in computation power is required to be created of new good algorithm and its application using object tracking. In Object Detection and Tracking we have to detect the target object and track that object in consecutive frames of a video file.

Given a set of object classes, object detection consists in determining the location and scale of all object instances, if any, that are present in an image. Thus, the objective of an object detector is to find all object instances of one or more given object classes regardless of scale, location, pose, and concern for the camera, partial occlusions, and illumination conditions.

LITERATURE SURVEY

The research conducted so far for object detection and tracking objects in video surveillance systems. Tracking is the process of locating the interested object within a sequence of frames, from its first appearance to its last. The type of object and its description within the system depends on the application. During the time that it is present in the scene, it may be occluded by other objects of interest or fixed obstacles within the scene. A tracking system should be able to predict the position of any occluded objects.

In [1], the author suggests an algorithm to isolate the moving objects in video sequences and then presented a rule-based tracking algorithm. The preliminary experimental results demonstrate the effectiveness of the algorithm even in some complicated situations, such as new track, ceased track, track collision, etc. A tracking method without background extraction is discussed in [2]. Because while extracting background from a video frame if there are small moving things in that frame they form a blob in thresholding which create confusion in the case of tracking that blob as they aren't of any use that can be reduced here. The author introduces video tracking in computer vision, including design requirements and a review of techniques from simple window tracking to tracking complex, deformable objects by learning models of shape and dynamics [3].

SCOPE

Finding and identifying a variety of things on an image or during real-time monitoring is called object detection. The "variable" component of the difference is crucial. As the number of items recognized may vary from picture to image, object detection output is less consistent than issues like categorization. So, in real life, we need detailed knowledge of our surroundings.[1].We must comprehend how the items are shifting concerning the camera. Understanding how different items interact might also be beneficial. For instance, understanding how people interact with one another can enable self-driving cars properly forecast pedestrian behaviour. This forecast will ultimately assist the self-driving car in making wise decisions on a busy route.[2]

METHODOLOGY

In This section, we discuss the proposed methodology and complete workflow of our project in detail. The projects consist of three phases, i.e., detecting the object, tracking the object and counting the object in the zone. Each of these phases is explained below.

A. *Phase 1: Object Detection*

This phase is concerned with the detection of objects. For the detection of objects, we used YOLOv4 [2] with pre-trained weights. YOLOv4 is an extension of YOLOv3, with improved speed and accuracy. All of the YOLO models are built and used for the detection of objects. Object detection models are trained to classify the objects in an image. When the object classes are detected, they are enclosed in the bounding boxes, and they are classified. Object detection models are generally trained and evaluated on the COCO (Common Objects in Context) dataset [3], containing a wide range of 80 object classes.[4] The main reasons for using YOLOv4 in our project are:

- For our project, real-time object detection is essential. The advantage of real-time detection models is that they are easy to wield by all developers.
- YOLOv4 makes it a priority for real-time object detection. The authors' intended for computer vision engineers and developers to use their YOLOv4 framework in custom domains quite easily.

B. *Phase 2: Tracking of Objects*

For tracking objects, we used the Deep SORT algorithm. It is an extension of the Simple Real-Time Tracker (SORT) algorithm. The Deep SORT algorithm uses the Kalman filters, and it is a crucial component to estimate the tracks of existing objects in the current frame. It has states that contain eight variables; (u, u', v, v', a, a', h, h'), where:

- a – aspect ratio
- (u,v) – centres of the bounding boxes
- h – the height of the image
- other variables are respective to their velocities

The DeepSORT uses the standard and straightforward form of Kalman filters that use constant velocities and linear observations.

Let's walk through the working of the DeepSORT algorithm:

- When every new frame comes, the position of every track is estimated based on its previous locations.

Only the spatial information is used for track estimation.

- Then we use the appearance feature vector to describe all the features of a given image.
- This vector obtains the appearance information of detections by extracting features and tracking images from previous frames.
- This vector is a convolution neural network trained on a large re-identification dataset.
- It extracts features such that the features with different identities are at a considerable distance apart, and the features with my identity are close together.
- The new detection results are associated with each coming frame's existing tracks using the appearance feature vector and the tracks' predicted position.
- Detections with low confidence value than the threshold are filtered out. New detections have to pass this threshold to become candidates for data association.

C. *Phase 3: Zonal Counting*

A zone is made using by setting the coordinates in the frame. We used the OpenCV library for visualizing the zone into the frame. We count the number of objects with the unique ID for calculating the number of objects in the zone. Then a limit is set for the zone. Initially, the zone is green in colour, and if the number of objects exceeds the limit, then the alarm is raised, and the colour of the zone changes to red.

Tracking works in such a way that when it detects a particular object it assigns a specific id number to that particular object and afterwards this unique id becomes the only source for tracking that particular object. By assigning unique IDs, it makes it easier for the tracker to keep track of objects throughout the video or imaging sequence.[5]The assigned id for the different objects can be seen in Fig. 2.

During the tracking Process, there may occur a scenario when a particular object is occluded by another object which may or may not be the point of study and can make the object in the study disappear from view entirely or partially. By applying the Kalman filters, this problem of occlusion is resolved. One example of occlusion occurrence and continuous tracking is shown in Fig. 3.

One of the critical applications of this object detection and tracking system in a particular zone is to prevent the spread of coronavirus by putting this system in the covid-19 hotspot areas and creating multiple zones, and raising the alarm when the number of people in the zone becomes greater than the threshold value. Fig. 4 and 5 show that when the number of people in the zone is less than 5 (threshold value), the zone becomes green, and when the number of people is greater than or equal to 5, then it becomes red.

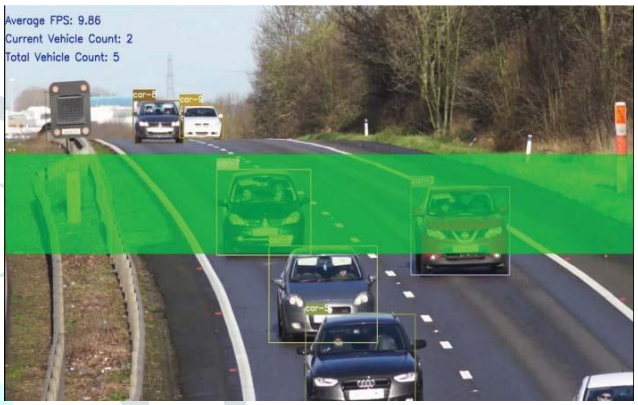

Fig. 1. Detected objects assigned with the unique ID

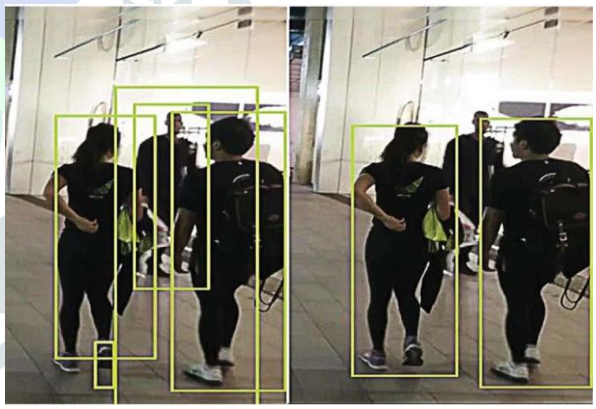

Fig. 2. Occlusions

Fig.3. The zone becomes red when the number of objects is greater than or equal to 5.

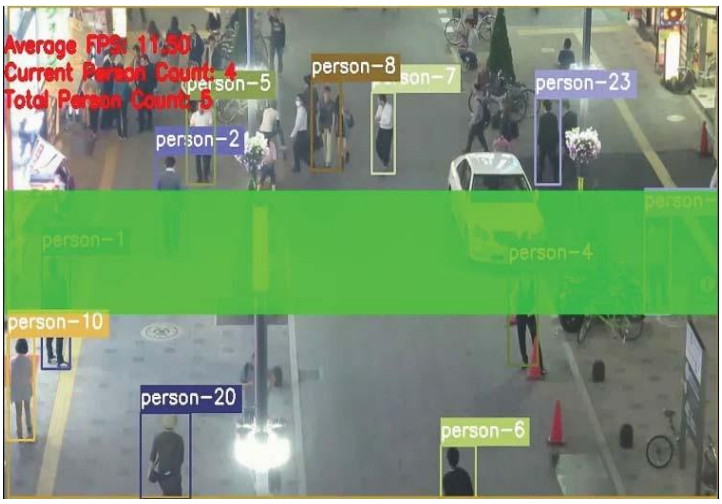

Fig. 4. The zone becomes green when the number of objects is less than 5.

RESULT ANALYSIS

We used YOLOv4 for object detection; therefore, the image classifier's accuracy is the same as that of YOLOv4. YOLOv4 shows a 10 percent and 12 percent increase in AP and FPS, respectively, compared to YOLOv3. Fig. 4, compares the YOLOv4 with YOLOv3.

YOLO detectors are best for object detection, which provides higher accuracy with higher fps. Table 1, compares the YOLO models' performance with another state of our models on the real-time COCO benchmark dataset [1].

We used the MOT17 dataset [5] for evaluating our model. This is a vast dataset containing **17757 frames, 2355 tracks and 564228 bounding boxes**. The dataset is formulated by extending the version of MOT16 [1] and MOT15 datasets [1], which has various image sequences including sequences with moving cameras shot from a bus, pedestrian scenes at night, people walking in a large square etc.

PERFORMANCE ON THE COCO BENCHMARK DATASET

Model	Map	FPS
CenterNet HarDNet-85	43.6	45
SSD512-HarDNet85	35.1	39
TTFNet	35.1	54.4
YOLOv3-418	31.0	34.0
YOLOv3-608	33.0	20.0
YOLOv4-512	43	83
YOLOv4-608	43.5	62

Evaluating Multi-Object Tracking causes some problems as:

- Datasets are challenging to establish.
- Multiple Object Tracking (MOT) algorithms are dependent on the detector, and the researchers do not always agree on the protocol to follow.
- Whether to evaluate the tracker with the same detector for all or to evaluate the whole system (detector plus tracker) together?
- Although some performance measures such as MOTA (Multiple Object Tracking Accuracy) are commonly used, some researchers still do not adhere to this metric.

We have used MOTA (Multiple Object Tracking Accuracy) for evaluating our model. The MOTA is the most used summary metric for MOT. It is defined as:

$$MOTA = 1 - \frac{\sum_t FN_t + FP_t + IDS_t}{\sum_t GT_t}$$

where,

- FP_t = number of false positives (ghost trajectories)
- IDS_t = number of identities switches at time t
- FN_t = number of false negatives (missed targets)
- GT_t = ground truth

Note: A target is considered missed if the IoU with the ground truth is less than the given threshold.

We evaluated our model with the MOT17 dataset on Jetson Xavier NX using py-mot metrics. When using public detections from the MOT17 dataset, the MOTA scores are close to state-of-the-art trackers. Tracking speed can reach up to 31 FPS (frames per second) depending on the object density. On a desktop CPU/GPU, FPS should be even higher. Even though the tracker runs much faster, it is still highly accurate. Lightweight detectors such as YOLOv4-tiny can potentially be used to obtain higher fps up to 44(fps).

CONCLUSION

Object detection is a key ability for most computer and robot vision systems. Although great progress has been observed in the last years, and some existing techniques are now part of many consumer electronics (e.g., face detection for auto-focus in smartphones) or have been integrated into assistant driving technologies, we are still far from achieving human-level performance, in particular in terms of open-world learning. It should be noted that object detection has not been used much in many areas where it could be of great help. As mobile robots, and in general autonomous machines, are starting to be more widely deployed (e.g., quad-copters, drones and soon service robots), the need for object detection systems is gaining more importance. Finally, we need to consider that we will need object detection systems for nano-robots or for robots that will explore areas that have not been seen by humans, such as depth parts of the sea or other planets, and the detection systems will have to learn to new object classes as they are encountered. In such cases, a real-time open-world learning ability will be critical.

REFERENCES:

1. Ribaric S, Adrinek G, Segvic S. Real-time Active visual tracking system[C].Proc. Of the 12th IEEE Mediterranean, Electrotechnical Conf.,2004,5:231-234.
2. Wren CR,Azarbayejani A,Darrell T,Pentland AP.Pfinder: Real-time Tracking Of the Human Body[J].IEEE Trans. on Pattern Analysis And Machine Intelligence,1997,19(7):780-785.
3. HaritaogluI, Harwood D, Davis. W4:Real-time Surveillance of people and their activities[J]. IEEE Trans. on Pattern Analysis And Ma-Chine Intelligence,2000,22(8):809-830.
4. Beauchemin, S. S. and J. L. Barron (1995). "The computation of optical flow." ACM computing surveys (CSUR) 27(3): 433-466.
5. Zhu Minghan, Luo Dayong. Moving Objects Detection and Tracking Based on Two Consecutive Frames Subtraction Background Model[J].Computer Measurement & Control, 2006, (08): 1004-1009.