Archives of
Transport System
Telematics

Volume 13

Issue 1

February 2020

Lane Queues Length Estimation Using Single 2D Camera and Image Depth Estimation Algorithm

K. BOLEK[a], K. LIVER[b]
[a] POLISH-JAPANESE ACADEMY OF INFORMATION TECHNOLOGY, Koszykowa 86, Warszawa, Poland
[b] WASKO S.A., Berbeckiego 6, Gliwice, Poland
EMAIL: k.bolek@pjwstk.edu.pl

ABSTRACT

During last 10 years the technology connected to video streaming analytics and deep learning algorithms connected to Intelligent Transport System and later, initiative of worldwide concept called "Smart City" pushed in reality a telematics into direction of IoT and possible use of neural network to support and create an added value in the way the Traffic Management Systems are collecting raw date in a massive data streaming mode, to be able to manage in an optimized and wright way big cities transportation networks. Recently HD cameras started to be one of the more widely used detectors replacing inductive loops and radars. The observations during productive life time of this kind of sensors on the field, created other challenges, efficiency aspects and unfavourable cost structure as consequence of usage of this product (HD camera), on such a big scale. Cameras designed to detect the length of queues at intersections require configuration by the operator. This process is extremely arduous, tedious and time consuming , considering how many cameras are located in cities. In addition, the configuration has to be done by a trained person, and what's more, moving or replacing the camera involves its manual reconfiguration. The aim of this article is to present a prototype of an algorithm that uses RetinaNet neural network to detect bikes/motorbikes on the street, and using monodepth2 determines the length of the queue and autonomously determines the direction of cars on the road. All the work undertaken confirmed that the approach used is effective and additionally allows to limit the operator's work only to defining the focal length and size of the camera sensor.

KEYWORDS: queues length estimation, RetinaNet, monodepth2

1. Introduction

Every day, the cameras dedicated to return queues in a given lane are replaced with new ones, installed in new locations or require reconfiguration. This requires the operator to log in to the camera via the web service, to define the detection zones along with the determination of the lane width and the length of the queue expressed in meters (set by operator) or the conventional unit that the operator will define. This process takes time and has a direct impact on the effectiveness of the camera proper working.

The build system for automatically queue length estimation consists of several elements:

- lane detection algorithm
- the algorithm for vehicle detection in a given lane
- algorithm for estimate the distance between visible vehicles and camera
- algorithm for determining the queue length

In order to achieve the autonomy of the system, neural network algorithms have been used for object/vehicle detection, 2D image depth prediction algorithms to detect the angle at which the camera looks at the lane and additionally algorithm to precise queue length in metres.

Creating a system that will automatically allow the camera to be configured without the involvement of the operator will save the operator's time and significantly shorten the process associated with the commissioning of the new investment. It will also make the system resistant to changes resulting from changes in traffic organization (e.g. closing a given lane).

2. Automatically lane and direction detection algorithm

Many neural networks only recognize vehicles and objects within close proximity to cameras, making them unsuitable for city surveillance. In this project for object detection, RetinaNet [1] was used, which, thanks to FPN, correctly detects objects regardless of their scale. This approach correctly searches for objects that are far from the camera.

Frame extraction from the camera recording and car detection on each frame allows to determine the number of lanes in the camera view. The longer the recording period is analysed, the less likely it is to make a mistake in not qualifying a less frequented lane (e.g. left-handed turn) as a crossroads element. In our experiment it turned out to be sufficient to analyse only a two-minute recording (30 FPS/s) during rush hours.

Apart from determining the sheer number of lanes, the direction in which the found vehicles move was also examined. In order to investigate the direction of their movement, the vectors of point displacement in Cartesian space were determined. The points determining the central point of the found bounding box of the vehicle were connected with the nearest neighbour in the next frame.

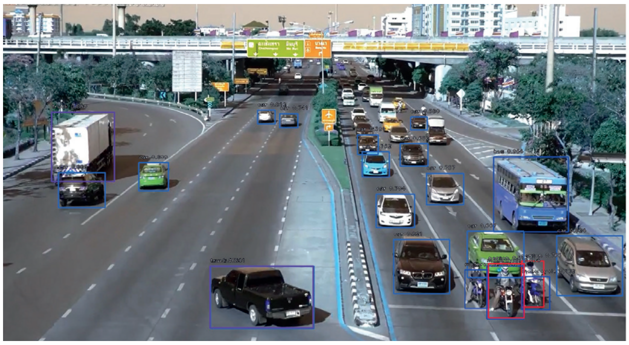

Fig. 1. Object detection for one camera frame using RetinaNet [own study]

Connection between these points makes it possible to determine the number of lanes as well as the direction of travel.

Such linking of straights led to the development of an intersection map without the intervention of the system operator.

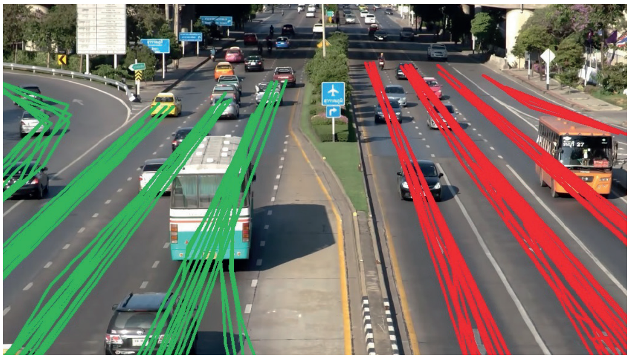

Fig. 2. Connections between the central point of bounding box found in each consecutive image frame determine the direction of travel and the number of lanes at an intersection [own study]

3. Algorithm for estimate the distance between visible vehicles and camera

The determination of the distance between vehicles at a junction and the cameras is done in several stages.

The first verification of the depth of the image seen by the camera is performed by the monodepth2 [2] depth prediction algorithm. The algorithm returns a map, which determines how the objects are located relative to each other in the image (without specifying the exact distance between them).

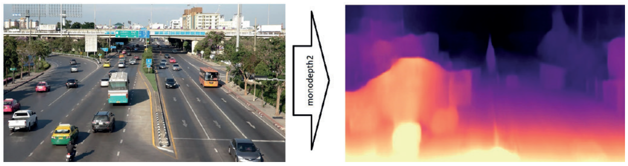

Fig. 3. A map of the depth of the image seen by the camera processed by monodepth2 [own study]

The algorithm, based on colors, marks the lines that determine on the preview image the distances of subsequent points seen in the camera view. The use of the monodeepth2 algorithm is necessary when determining the distance between points on the roadway when the camera is not set parallel to the axis of the roadway.

After determining the horizontal lines determining the relative distance between points/elements visible in camera view, the algorithm searches for specific object classes whose height is set rigidly. The algorithm searches:

- motorcycles (average height is about 105cm and length 230cm)
- bikes (average height is about 100cm and length 180cm)

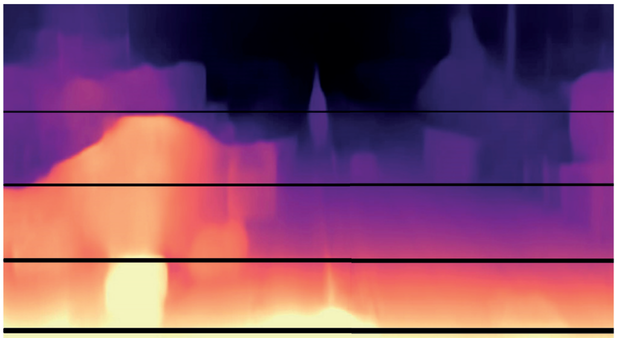

Fig. 4. Horizontal lines determined by defining points of the same distance (colour) at the longest intersection. Due to the inaccuracy of the algorithm, only the first line is determined in this way, and each other line is a line parallel to the determined one [own study]

The algorithm takes the focal length of 12mm by default (the operator has the possibility to define another one on start). Based on the above mentioned sizes and by knowing the real height of the object (d), the focal length of the camera (a) and the height of the object in the frame (c), the distance between the camera and the object (b) can be deduced.

In order to increase the accuracy of the algorithm's operation, the distortion caused by the detection of a given object from above is also taken into account. By eliminating the inaccuracy of the bounding boxes of the detected objects, the accuracy was increased through an uninterrupted, iterative process of specifying the distance with each detection (every few seconds, searches for more motorbikes/bikes to increase the accuracy of the results). The first 3 iterations of the algorithm are shown below:

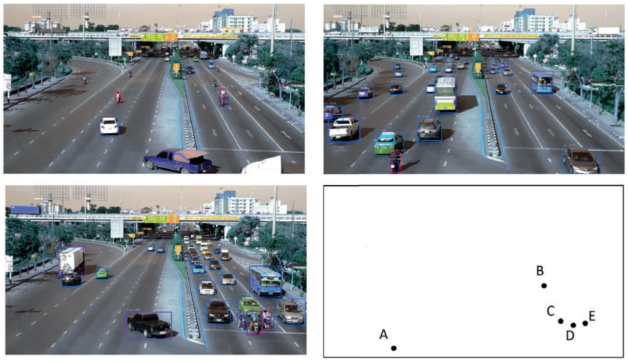

Fig. 7. Grouping the results of motorcycle detection performed on first three frames [own study]

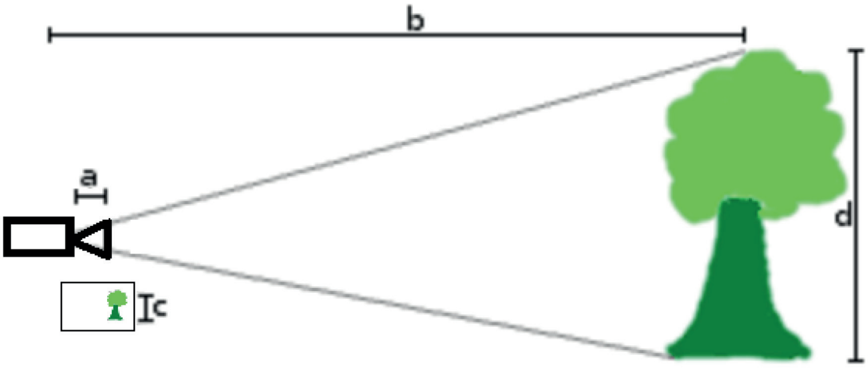

Fig. 5. A simple diagram showing how the distance is calculated using the similarity of the triangles [own study]

$$\textit{Distance to object (mm)} = \frac{\textit{f (mm) x real height (mm) x image height (pixels)}}{\textit{object height (pixels) x sensor height (mm)}}$$

Sensor size:

Detected point information :

A - {“class”: “motorcycle”,“coordinates”: [427, 1000, 483, 1078]}

B - {“class”: “motorcycle”,“coordinates”: [1373, 593, 1434, 655]}

C - {“class”: “motorcycle”,“coordinates”: [1477, 820, 1541, 926]}

D - {“class”: “motorcycle”,“coordinates”: [1538, 809, 1639, 955]}

E - {“class”: “motorcycle”,“coordinates”: [1634, 839, 1710, 930]}

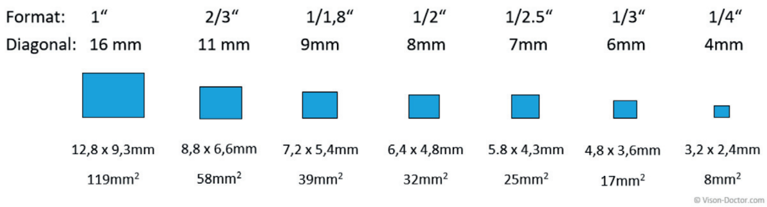

Fig. 6. Selected sensor sizes necessary for distance estimation [own study]

Calculated distance:
A = (6mm * 1,05m * 2160px) / (78px * 5,4mm) = 32m
B = 40,65m
C = 23,77m
D = 17,26m
E = 23,12m

After more than 1500 motorbike/bike detection algorithm stop working and save results in matrix of averaged results for points in a given area (the camera image has been divided into areas of 80x80px). The matrix is saving to .csv format and has size 48x27. Based on the horizontal lines defining the image depths determined by the monodepth2, the values are smoothed horizontally between the individual areas in a given line, followed by vertical smoothing. Both smooths are done by calculating a weighted average with the rejection of extreme high and low values. Below, the effect of such smoothing is shown graphically by applying the results to the camera image. The real distance was determined by using the Google Earth tool.

Fig. 8. Comparison of the real distance with the result of the algorithm [own study]

The approximation result shows that the algorithm approximation smaller distance than real, but the distances between the successive lines are kept within the permissible error margin.

4. Algorithm for determining the queue length

To estimate the queue length in a given lane, the camera uploads video frames every few seconds to the server running RetinaNet neural network. Once processed, the detection is merged with the lines that determine the distance between the vehicles and the stop line. Determination of the stopping line of vehicles is based on the detection of the place where the vehicles are not moving, while meeting the condition that no other vehicle has been found in front of them. Repeated detection of the stopping point makes it possible to determine the line in front of which the vehicles are stopping with big accuracy. This method is unreliable in case of traffic jams and when the camera doesn't cover the beginning of the lane in its view. In these situations the algorithm will not work properly. In order to detect such a deviation, the algorithm verifies whether the vehicles are detected during a stop in the immediate vicinity of the lower edge of the camera view, and then passes the information to the server that the queue length cannot be determined in this lane.

When a stopped vehicle is detected, a new queue length is calculated for that lane. Thanks to this approach, the algorithm cyclically verifies the occupancy of the lane and differently transmits information about the traffic condition to the server.

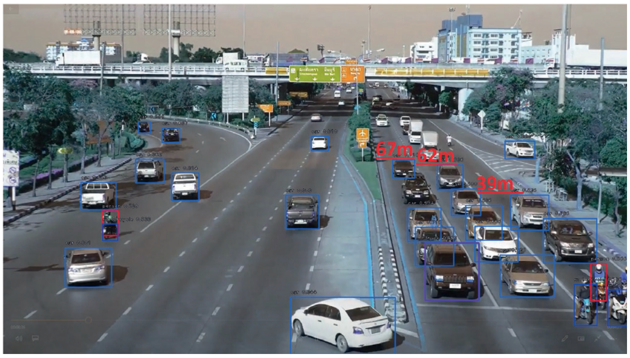

Fig. 9. Detection of the last vehicle stopped in a given lane determines the point in the image that determines the end of the queue, which is then mapped to the distance in meters determined by the algorithm [own study]

5. Further steps and conclusions

All the work and verification of the efficiency of operation was performed by using the recordings from the camera monitoring system located at the intersection of coordinates 13.790939, 100.681019.

The whole work revealed that the use of RetinaNet and vehicle detection approach on the road to determine the length of queue on a given lane is possible and gives satisfactory results.

The aim of all the works was to minimize the time of the operator's involvement in the configuration of the camera for counting the length of queues at intersections. This goal was achieved. The operator can specify the accuracy by giving only 2 information (focal length and size of the camera sensor). The rest of the activities are performed automatically, and correct determination of queues length is possible after about 1500 bike/motorcycle detection on the road.

The next step will be first of all to compare the effectiveness of other neural networks for detection of vehicles seen from city surveillance cameras and to implement the algorithm to the traffic management system in Wroclaw.

Bibliography

[1] LI Y., FENGBO R.: Light-weight retinanet for object detection. arXiv preprint arXiv:1905.10011, 2019
[2] GODARD C., ET AL.: Digging into self-supervised monocular depth estimation. Proceedings of the IEEE international conference on computer vision, 2019