Scientific Research Publishing

Journal of Geographic Information System, 2021, 13, 395-409
https://www.scirp.org/journal/jgis
ISSN Online: 2151-1969
ISSN Print: 2151-1950

YOLOv2 Deep Learning Model and GIS Based Algorithms for Vehicle Tracking

Mohamed El Imame Malaainine[1], Hatim Lechgar[2], Hassane Rhinane[2]

[1]Hassania School of Public Works, Casablanca, Morocco
[2]FSAC-Hassan II University, Casablanca, Morocco
Email: mohamed.malaainine@gmail.com

How to cite this paper: Malaainine, M.E.I., Lechgar, H. and Rhinane, H. (2021) YOLOv2 Deep Learning Model and GIS Based Algorithms for Vehicle Tracking. *Journal of Geographic Information System*, **13**, 395-409.
https://doi.org/10.4236/jgis.2021.134022

Received: June 3, 2021
Accepted: July 13, 2021
Published: July 16, 2021

Abstract

The latest advances in Deep Learning based methods and computational capabilities provide new opportunities for vehicle tracking. In this study, YOLOv2 (You Only Look Once—version 2) is used as an open source Convolutional Neural Network (CNN), to process high-resolution satellite images, in order to generate the spatio-temporal GIS (Geographic Information System) tracks of moving vehicles. At first step, YOLOv2 is trained with a set of images of 1024 × 1024 resolution from the VEDAI database. The model showed satisfactory results, with an accuracy of 91%, and then at second step, is used to process aerial images extracted from aerial video. The output vehicle bounding boxes have been processed and fed into the GIS based LinkTheDots algorithm, allowing vehicles identification and spatio-temporal tracks generation in GIS format.

Keywords

Deep Learning, YOLOv2, Vehicle Tracking, GIS

1. Introduction

Vehicles tracking is an important subject with interesting applications. It has been extensively studied from different angles, using both classical methods of traditional object detection and GIS methods, based on GPS and real time communications tools.

As one of the most important tasks in computer vision, object detection is rapidly growing, thanks to the latest advances in deep learning based methods and computational power with clusters of graphics processing units (GPUs). This offers new opportunities for vehicle tracking, through the use of high-resolution satellite imagery and deep learning methods, based on Convolutional Neural

Networks (CNNs) [1]. In this paper, for vehicle tracking purposes, YOLOv2 model [2], a fast growing open source CNN, is train on VEDAI images, an open dataset of vehicles imagery. GIS functionalities and LinkTheDotes algorithm are used for spatio-temporal tracks creation, control and visualization.

The plan of the paper is as follows. This section presents a literature review of some studies on vehicle tracking and object detection, with the basic concepts of Deep learning, CNN and YOLO. Section 2 presents the general approach, preparation of input data, YOLOv2 training and LinkTheDots algorithm as well as used GIS features. Section 3 examines the results obtained and section 4 provides some conclusions.

1.1. Vehicle Tracking

Vehicles tracking became an important task with important applications in many fields such as urban traffic monitoring [3], intelligent transportation systems [4], ground surveillance [5], driving safety and security [6], advanced driving assistance systems [7], etc.

While classical methods of vehicle tracking are based on the combination of GPS, GSM, GPRS and internet technologies [8] [9] [10] [11], new methods based on imagery and AI are rapidly evolving [12]-[17]. the advantage of these new methods is their ability to process data at large scales, without the need to first install special equipment in tracked vehicles; They take advantage of accelerated advances in artificial intelligence, especially deep learning, and thus significantly reduce the cost of access to these analysis data for the largest number of interested researchers and businesses.

1.2. Object Detection and GIS

Object detection consists of detecting instances of a certain class (such as vehicles, humans, or trees) in digital images. It is a computer vision subject, that finds numerous applications, in several fields such as facial recognition [18], autonomous driving [19], and lately face mask detection amid COVID-19 pandemic [20]. The main objective of object detection is to develop computational systems that deliver a key information to computer vision applications which is: "What objects are where"? [21], which is also the basis of multiple GIS (Geographic Information Systems) applications. The two areas benefit and complement each other [22] [23] [24].

1) Object detection and image classification

The objective of image classification is to extract existing classes of visual objects, without necessarily specifying their location in the image. It answers the question "what object is in the image?".

On the other hand, object detection locates instances of classes on the image, with bounding boxes or bounding polygons [25] as shown in **Figure 1**.

2) Object identification

Object identification happens when the detected objects in the image are

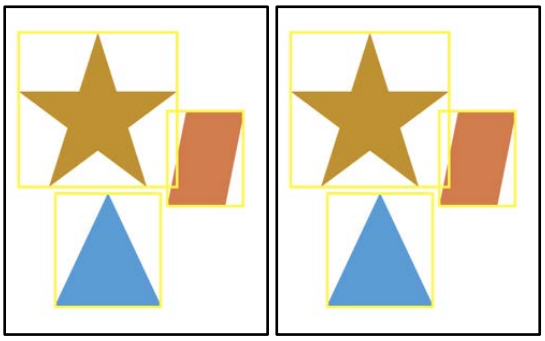

Figure 1. Bounding boxes (left) vs. bounding polygons (right).

assigned unique identification codes. It is used for real-time object tracking applications for instance [26].

1.3. Image Processing with Deep Learning

1) Deep Learning (DL)

Recent research works show that Deep Learning methods have arose as powerful Machine Learning methods for object recognition and detection [27] [28] [29] [30] [31]. Deep learning happens with complicated nonlinearity, when composing many nonlinear functions [32]. While traditional approaches of Artificial Intelligence and Machine Learning make it possible to learn hierarchical representations corresponding specifically to the analyzed data [33], we tend to believe that with Deep Learning Neural Networks, there is an incremental evolution of the representation of raw data into categories of abstractions as the system is fed with data [34] [35]. Thus, with its boosted capacity to adjust billions of parameters thanks to massive parallelism computing capabilities, Deep Learning algorithms success in AI application such as image and video processing stands phenomenal [36].

2) Convolutional Neural Networks (CNN)

When dealing with images, unlike the traditional approaches, Deep Learning models learn the features immediately from the raw pixels, developing local receptive fields from lower layers to upper layers. For instance, lower layers recognize simple features like lines and corners, while higher layers extract complex features representing real life objects such as vehicles. The successes of DL in image processing are testified by the challenging ImageNet classification task across thousands of classes [30] [37] by using a kind of deep neural network called a Convolutional Neural Networks (CNN) [38].

The structure of CNNs was initially based on the animal visual cortex organization [39]. After a slow start in the early 1990s due to computing capacity limits [40] [41], CNNs experienced a huge boom with the rapid development of these capabilities with, among others, cloud computing.

CNNs are made up of several layers similar to feed-forward neural networks. The outputs and inputs of the layers are given as a set of image matrices. CNNs can be constructed by different combinations of convolutional layers (where

convolution operation is done on specified filters), pooling layers, and fully connected layers (generally, before the output) with nonlinear activation functions. A typical CNN architecture is shown in **Figure 2** [38].

3) Single Shot CNN: YOLO

You Only Look Once (YOLO) is a Convolutional Neural Network object detection system, that handles object detection as one regression problem, from image pixels to bounding boxes with their class probabilities. Its performance is much better than other traditional methods of object detection, since it trains directly on full images.

YOLO is formed of 27 CNN layers, with 24 convolutional layers, two fully connected layers, and a final detection layer [2] (**Figure 3**).

YOLO divides the input images into an N by N grid cell, then during the processing, predicts for each one of them several bounding boxes to predict the object to be detected. Thus, a loss function has to be calculated. YOLO calculates first, for each bounding box, the Intersection over Union (IoU); It uses then sum-squared error to calculate error loss between the predicted results and real objects. The final loss being the sum of the three loss functions: 1) classification loss: related to class probability, 2) localization loss: related to the bounding box position and size and 3) confidence loss measuring the probability of objects in the box [42].

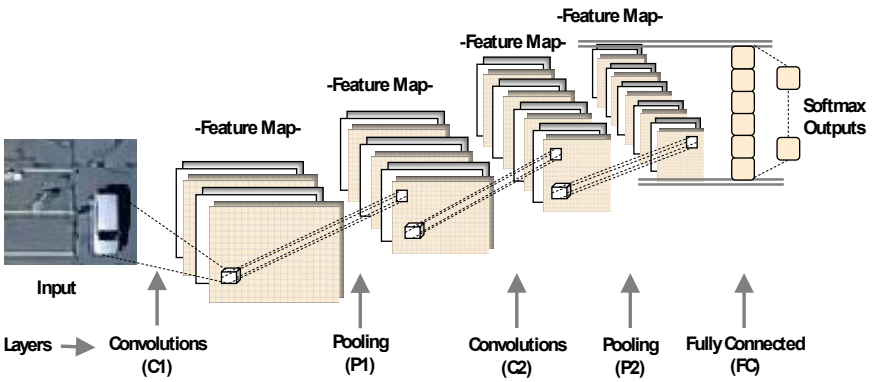

Figure 2. Convolutional neural networks architecture [38].

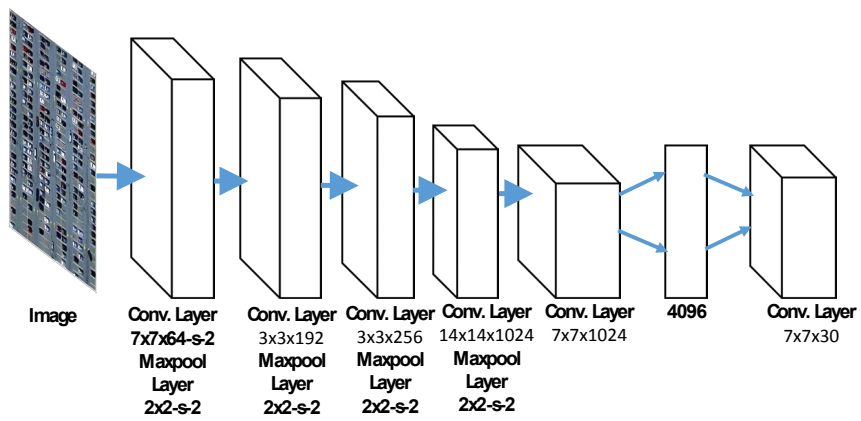

Figure 3. YOLO architecture [2] [42].

2. Methodology

In order to generate vehicles temporal paths in GIS format from aerial video, a three steps process is adopted:

- To solve the problem of handling continuous aerial video stream, which represents a big technical challenge [43], the video stream is converted into a series of images, with a suitable resolution for the trained YOLOv2 algorithm.
- Each individual image is then processed with YOLOv2 algorithm trained beforehand.
- With LinkTheDots algorithm, the detected vehicles are then tracked throughout the output series of images, generating a specific GIS dated path for each vehicle.

Figure 4 shows the general process, and **Figure 5** presents the process of YOLOv2 algorithm training (LinkTheDots algorithm process is detailed later in this section).

2.1. Input Data: From Areal Video to a Series of Images

From an aerial video of a busy parking lot [44], the series of frames was extracted. **Figure 6** presents one of the extracted images.

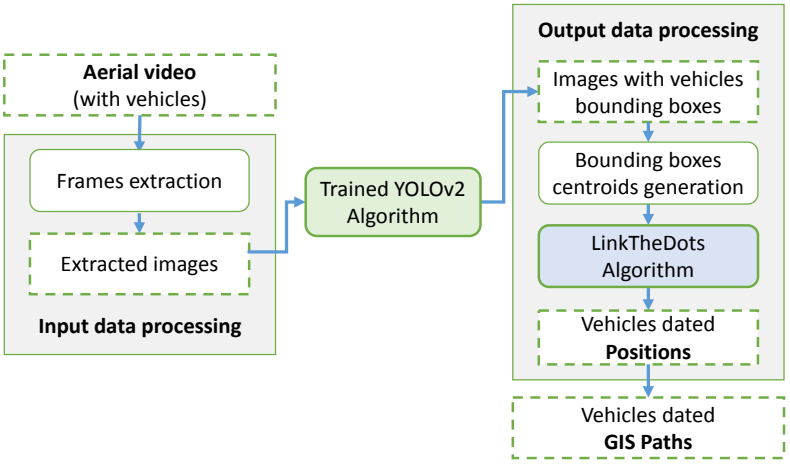

Figure 4. Method's general process.

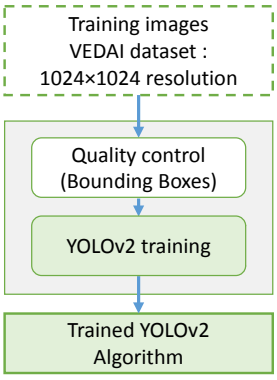

Figure 5. YOLOv2 algorithm training.

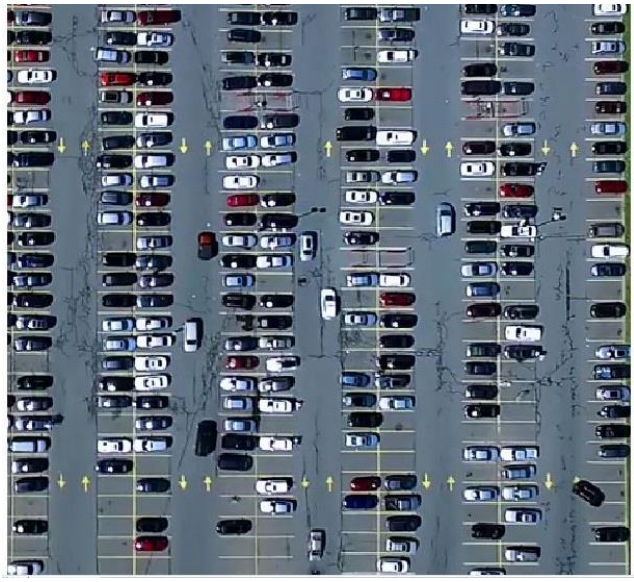

Figure 6. A frame from the series of extracted images from the areal video [44].

The metadata of each frame contains the detailed date of the image, which is inherited by all detected vehicles on the frame.

At this stage of the study, the set of images are ready to be processed one by one, with the trained YOLOv2 algorithm for vehicles detection.

2.2. YOLOv2 Algorithm Training

1) Training data

YOLO and CNN algorithms in general, when applied on imagery data, can be trained with data from anywhere and applied with the same degree of certainty elsewhere [45]. For this reason, in the absence of local data sources of areal imagery, VEDAI (Vehicle Detection in Aerial Imagery) data source [45] is used. In addition to its open access and the important number of offered images (more than 10,000), VEDAI database offers labels for each vehicle, ready to use for recognition algorithms trainings **Figure 7**.

The YOLOv2 model was trained and tested with a set of images of 1024 × 1024 resolution. Overall, a dataset of 1200 images were used; 70% of them as training data and 30% for tests.

2) Training platform

YOLO algorithm training, like all deep learning models, requires considerable computing capacity [32]. Therefore, the used platform was in the cloud with the configuration specified in **Table 1**. One of the most important aspects of this configuration is the high performance GPU (Graphics Processing Unit), as it has an efficient parallel architecture for model learning. Combined with clusters or cloud computing, it considerably reduces network training time.

Darknet [46] was used as a training framework; it is an open source Neural Network framework written in C and CUDA that supports CPU and GPU computation.

Figure 7. VEDAI dataset image.

Table 1. YOLOv2 training environment specifications.

Environment	Amazon AWS
Instance type	p2.xlarge
System	Windows Server 2016-X64
Processor	4 CPU Intel Xeon E5-2686 V4 2.30Ghz
RAM	61Go
GPU	1 GPU NVIDIA TESLA K80 with 12Go in memory
HDD	100Go

2.3. LinkTheDots Algorithm

In order to track the same vehicle throughout successive frames, LinkTheDots algorithm was developed. Its main task is to link the centroid of a vehicles bounding box on a certain frame, to the centroid of the same vehicle's bounding box on the next frame. This would indicate that, between the two frames instants, this particular vehicle has moved from the first point to the second.

After all the frames are processed with the trained YOLOv2 algorithm and all bounding boxes are generated, all vehicles' centroids are created with GIS tools. LinkTheDots algorithm processes then all of these resulting frames, starting with the first, where all points should be identified by a vehicle's ID. From there, starting with the second frame, the algorithm must check if the associated vehicle has already been identified in the previous frame in order to obtain its ID, otherwise, a new vehicle's ID must be attributed. **Figure 8** shows the detailed process of LinkTheDots algorithm.

LinkTheDots identifies the position of the vehicle position in the previous frame by performing a geographic search, within a distance of Δmax, beyond which, no vehicle would ever be able—supposedly—to move between two frames time, given the assumed parameters such as maximum vehicle speed. Therefore Δmax is considered as an algorithm adjustment parameter.

3. Results and Discussion

3.1. YOLOv2 Algorithm Training Results

Here below, in **Table 2**, the main parameters of a YOLOv2 training:

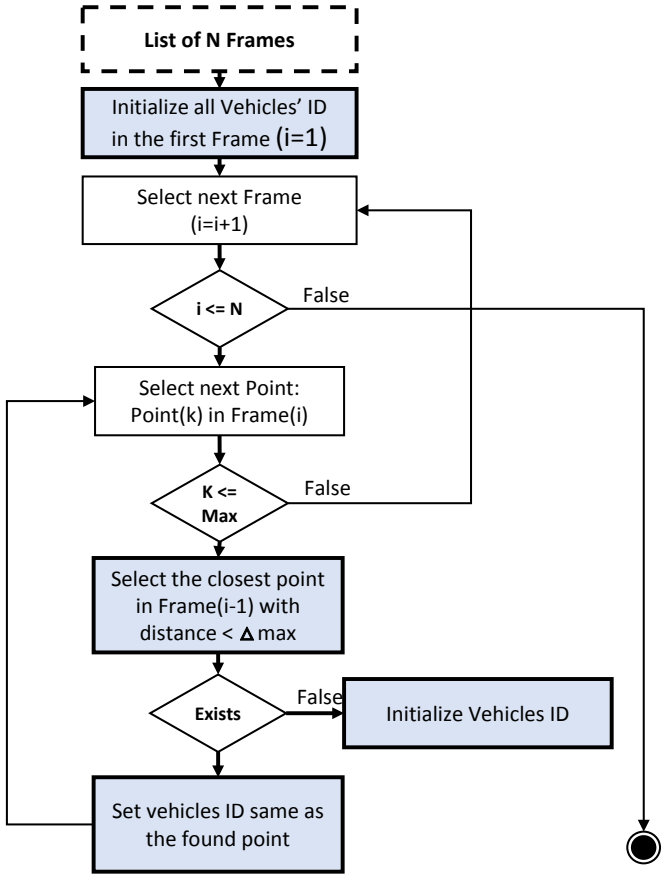

Figure 8. LinkTheDots algorithm process.

Table 2. YOLOv2 training output parameters.

Parameter	Indication	Target
Avg IOU	Average "Intersection over Union" IOU = Area of Overlap/Area of Union The two areas are: "the predicted bounding box" and "the ground truth of target object"	100%
Avg recall	Average "Recall" = Recall/Count The ratio of the number of detected objects to the total number of objects to be detected	100%
Count	The total number of objects to be detected in the current set (number of originals)	-
Number of iterations	The number of iterations	-
Average loss	The average loss error	As low as possible
Total time	The total time spent processing this batch	-

In **Table 3**, the results of the YOLOv2 training are presented: images resolution, dataset size, beginning of convergence, number of iterations, average loss and training duration.

The evolution of the average loss during the iterations of the learning process is presented in **Figure 9** and test results illustration is presented in **Figure 10**.

Table 3. Parameters and overall results of YOLOv2 training.

Training images resolution	Number of images	Beginning of convergence	Number of iterations	Average loss	Training duration
1024 × 1024	1200	802 Iteration	22,000	0.05	7 days

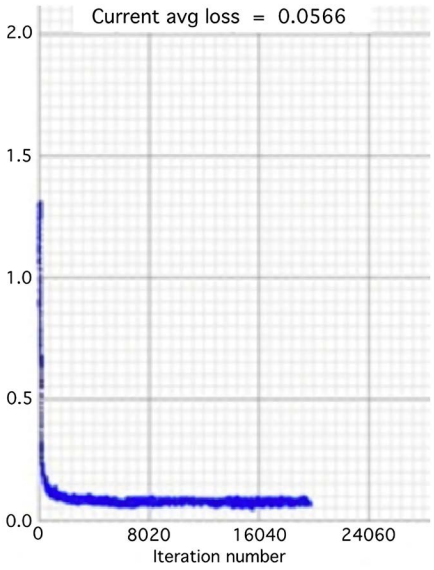

Figure 9. Evolution of the average loss according to the number of iterations.

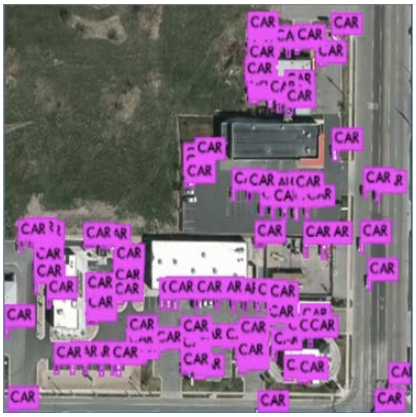

Figure 10. YOLOv2 test results illustration.

The model detected 91% of test vehicles. These results show that the trained model can identify vehicles with satisfactory accuracy that meets the intended application requirements for spatio-temporal tracking. With a larger set of training images, this accuracy can be significantly improved.

3.2. Vehicles Tracking Results

The results of the trained YOLOv2 algorithm and the processing of the output data (**Figure 4**), are 1) the table of positions of moving vehicles, produced by LinkTheDots algorithm, an extract of which is presented in **Table 4**; And 2) vehicles' positions throughout the input areal video time, shown in **Figure 11**.

Table 4. Excerpt from LinkTheDots space-time table of moving vehicles.

Point ID	Frame	Vehicle ID	TimeStamp (ddmmyyHHMMSS)
1503	7	8	08052018133553
1489	6	8	08052018133551
1202	5	8	08052018133549
835	4	8	08052018133547
652	3	8	08052018133545
354	2	8	08052018133543
193	1	8	08052018133541

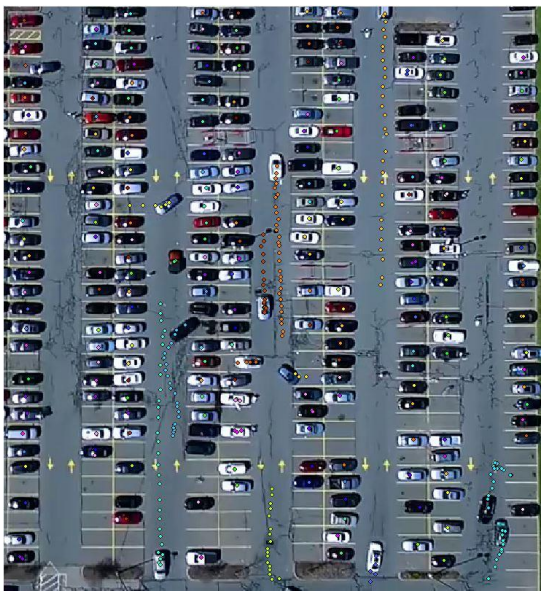

Figure 11. Generated centroids throughout time.

Using GIS tools to convert collections of points to lines, these points were converted into circuits, sorted by vehicles' ID numbers. Thus, the spatio-temporal tracks of moving vehicles in the areal video were obtained (**Figure 12**).

3.3. The LinkTheDots Algorithm Limits

LinkTheDots algorithm is based on the assumption that the nearest bounding box centroid in the following image is related to the same vehicle. The algorithm parameter Δmax must be then set to a value that avoids confusion between two different vehicles on two successive frames.

Let:

Δ: The vehicle's travelled distance between two frames

W_{vehicle}: The vehicle's width

Then, to avoid confusion between vehicles, we must have:

$$\Delta < W_{\text{vehicle}} \quad (1)$$

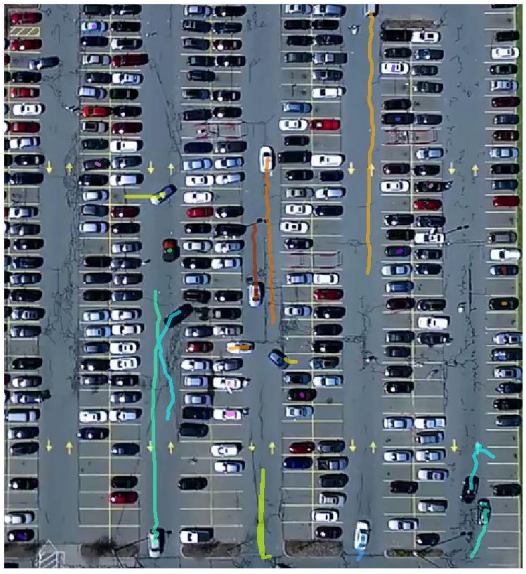

Figure 12. Veihicles GIS spatio-temporal tracks.

This means that Δmax must be less than the minimum vehicle's width.

Let:

V_{vehicle}: The vehicle's velocity

V_{camera}: The velocity of the camera

F_r: the number of frames per second (frames' rate)

Then:

$$\Delta = \frac{V_{\text{vehicle}} - V_{\text{camera}}}{F_r} \tag{2}$$

From (1) and (2):

$$\frac{V_{\text{vehicle}} - V_{\text{camera}}}{F_r} < W_{\text{vehicle}} \tag{3}$$

This implies that, in the case of a static camera ($V_{\text{camera}} = 0$), for an average vehicle width of 2 meters and a camera frame rate of 15 frames per second, the maximum velocity up to which a vehicle can be tracked is 30 m/s (108 km/h).

Another implication would be that if it is intended to track a vehicle with a velocity of 150 km/h—still with a static camera—the used camera should have a rate of 21 frames per second or better.

4. Conclusion

In this work, YOLOv2 model was trained for the detection of vehicles on aerial images. The trained model was coupled with LinkTheDots algorithm for GIS spatio-temporal tracking. The limits and the conditions of validity of the proposed algorithm were discussed according to the frames' rate in the raw aerial video and the speed of the tracked vehicles. The accuracy of the trained model which was found around 91% can be significantly improved, with a larger set of training images.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Yoon, Y., Jeon, H.G., Yoo, D., Lee, J.Y. and So Kweon, I. (2015) Learning a Deep Convolutional Network for Light-Field Image Super-Resolution. *Proceedings of the IEEE International Conference on Computer Vision Workshops*, Santiago, 7-13 December 2015, 24-32. https://doi.org/10.1109/ICCVW.2015.17

[2] Redmon, J., Divvala, S., Girshick, R. and Farhadi, A. (2016) You Only Look Once: Unified, Real-Time Object Detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, 27-30 June 2016, 779-788. https://doi.org/10.1109/CVPR.2016.91

[3] Cao, X., Jiang, X., Li, X. and Yan, P. (2016) Correlation-Based Tracking of Multiple Targets with Hierarchical Layered Structure. *IEEE Transactions on Cybernetics*, **48**, 90-102. https://doi.org/10.1109/TCYB.2016.2625320

[4] Chen, L. and Englund, C. (2015) Cooperative Intersection Management: A Survey. *IEEE Transactions on Intelligent Transportation Systems*, **17**, 570-586. https://doi.org/10.1109/TITS.2015.2471812

[5] Ulmke, M. and Koch, W. (2006) Road-Map Assisted Ground Moving Target Tracking. *IEEE Transactions on Aerospace and Electronic Systems*, **42**, 1264-1274. https://doi.org/10.1109/TAES.2006.314571

[6] Ibrahim, V.M. and Victor, A.A. (2012) Microcontroller Based Anti-Theft Security System Using GSM Networks with Text Message as Feedback. *International Journal of Engineering Research and Development*, **2**, 18-22.

[7] Hasberg, C., Hensel, S. and Stiller, C. (2011) Simultaneous Localization and Mapping for Path-Constrained Motion. *IEEE Transactions on Intelligent Transportation Systems*, **13**, 541-552. https://doi.org/10.1109/TITS.2011.2177522

[8] Almomani, I.M., Alkhalil, N.Y., Ahmad, E.M. and Jodeh, R.M. (2011) Ubiquitous GPS Vehicle Tracking and Management System. 2011 *IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies* (*AEECT*), Amman, 6-8 December 2011, 1-6. https://doi.org/10.1109/AEECT.2011.6132526

[9] Maurya, K., Singh, M. and Jain, N. (2012) Real Time Vehicle Tracking System Using GSM and GPS Technology—An Anti-Theft Tracking System. *International Journal of Electronics and Computer Science Engineering*, **1**, 1103.

[10] Lee, S., Tewolde, G. and Kwon, J. (2014) Design and Implementation of Vehicle Tracking System Using GPS/GSM/GPRS Technology and Smartphone Application. 2014 *IEEE World Forum on Internet of Things* (*WF-IoT*), Seoul, 6-8 March 2014, 353-358. https://doi.org/10.1109/WF-IoT.2014.6803187

[11] Pham, H.D., Drieberg, M. and Nguyen, C.C. (2013) Development of Vehicle Tracking System Using GPS and GSM Modem. 2013 *IEEE Conference on Open Systems* (*ICOS*), Kuching, 2-4 December 2013, 89-94. https://doi.org/10.1109/ICOS.2013.6735054

[12] Tang, Z., Naphade, M., Liu, M.Y., Yang, X., Birchfield, S., Wang, S., Hwang, J.N., *et al.* (2019) Cityflow: A City-Scale Benchmark for Multi-Target Multi-Camera Vehicle Tracking and Re-Identification. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Long Beach, 15-20 June 2019, 8797-8806. https://doi.org/10.1109/CVPR.2019.00900

[13] Tang, Z., Wang, G., Xiao, H., Zheng, A. and Hwang, J.N. (2018) Single-Camera and Inter-Camera Vehicle Tracking and 3D Speed Estimation Based on Fusion of Visual and Semantic Features. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, Salt Lake City, 18-23 June 2018, 108-115. https://doi.org/10.1109/CVPRW.2018.00022

[14] Chen, Y., Jing, L., Vahdani, E., Zhang, L., He, M. and Tian, Y. (2019) Multi-Camera Vehicle Tracking and Re-Identification on AI City Challenge 2019. *CVPR Workshops*, Vol. 2, 324-332.

[15] Hua, S., Kapoor, M. and Anastasiu, D.C. (2018) Vehicle Tracking and Speed Estimation from Traffic Videos. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, Salt Lake City, 18-23 June 2018, 153-160. https://doi.org/10.1109/CVPRW.2018.00028

[16] Peri, N., Khorramshahi, P., Rambhatla, S.S., Shenoy, V., Rawat, S., Chen, J.C. and Chellappa, R. (2020) Towards Real-Time Systems for Vehicle Re-Identification, Multi-Camera Tracking, and Anomaly Detection. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, Seattle, 14-19 June 2020, 622-623. https://doi.org/10.1109/CVPRW50498.2020.00319

[17] Li, P., Li, G., Yan, Z., Li, Y., Lu, M., Xu, P., Chuxing, D., *et al.* (2019) Spatio-Temporal Consistency and Hierarchical Matching for Multi-Target Multi-Camera Vehicle Tracking. *CVPR Workshops*, Long Beach, 16-20 June 2019, 222-230.

[18] Yang, D., Alsadoon, A., Prasad, P.C., Singh, A.K. and Elchouemi, A. (2018) An Emotion Recognition Model Based on Facial Recognition in Virtual Learning Environment. *Procedia Computer Science*, **125**, 2-10. https://doi.org/10.1016/j.procs.2017.12.003

[19] Grigorescu, S., Trasnea, B., Cocias, T. and Macesanu, G. (2020) A Survey of Deep Learning Techniques for Autonomous Driving. *Journal of Field Robotics*, **37**, 362-386. https://doi.org/10.1002/rob.21918

[20] Loey, M., Manogaran, G., Taha, M.H.N. and Khalifa, N.E.M. (2021) Fighting against COVID-19: A Novel Deep Learning Model Based on YOLO-v2 with ResNet-50 for Medical Face Mask Detection. *Sustainable Cities and Society*, **65**, Article ID: 102600. https://doi.org/10.1016/j.scs.2020.102600

[21] Zou, Z., Shi, Z., Guo, Y. and Ye, J. (2019) Object Detection in 20 Years: A Survey.

[22] Ardeshir, S., Zamir, A.R., Torroella, A. and Shah, M. (2014) GIS-Assisted Object Detection and Geospatial Localization. In: *European Conference on Computer Vision*, Springer, Cham, 602-617. https://doi.org/10.1007/978-3-319-10599-4_39

[23] Campbell, A., Both, A. and Sun, Q.C. (2019) Detecting and Mapping Traffic Signs from Google Street View Images Using Deep Learning and GIS. *Computers, Environment and Urban Systems*, **77**, Article ID: 101350. https://doi.org/10.1016/j.compenvurbsys.2019.101350

[24] Cheng, G. and Han, J. (2016) A Survey on Object Detection in Optical Remote Sensing Images. *ISPRS Journal of Photogrammetry and Remote Sensing*, **117**, 11-28. https://doi.org/10.1016/j.isprsjprs.2016.03.014

[25] Hurtik, P., Molek, V., Hula, J., Vajgl, M., Vlasanek, P. and Nejezchleba, T. (2020) Poly-YOLO: Higher Speed, More Precise Detection and Instance Segmentation for YOLOv3.

[26] Bathija, A. and Sharma, G. (2019) Visual Object Detection and Tracking Using Yolo and Sort. *International Journal of Engineering Research Technology*, **8**, 705-708.

[27] Deng, L. and Yu, D. (2014) Deep Learning: Methods and Applications. *Foundations*

and Trends in Signal Processing, **7**, 197-387. https://doi.org/10.1561/2000000039

[28] Szegedy, C., Toshev, A. and Erhan, D. (2013) Deep Neural Networks for Object Detection.

[29] Krizhevsky, A., Sutskever, I. and Hinton, G.E. (2012) Imagenet Classification with Deep Convolutional Neural Networks. *Advances in Neural Information Processing Systems*, **25**, 1097-1105.

[30] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., *et al.* (2015) Imagenet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, **115**, 211-252. https://doi.org/10.1007/s11263-015-0816-y

[31] Yan, K., Wang, Y., Liang, D., Huang, T. and Tian, Y. (2016) CNN vs. Sift for Image Retrieval: Alternative or Complementary? *Proceedings of the* 24*th ACM International Conference on Multimedia*, Amsterdam, 15-19 October 2016, 407-411. https://doi.org/10.1145/2964284.2967252

[32] Fan, J., Ma, C. and Zhong, Y. (2019) A Selective Overview of Deep learning.

[33] Ilin, R., Watson, T. and Kozma, R. (2017) Abstraction Hierarchy in Deep Learning Neural Networks. 2017 *International Joint Conference on Neural Networks* (*IJCNN*), Anchorage, 14-19 May 2017, 768-774. https://doi.org/10.1109/IJCNN.2017.7965929

[34] Schmidhuber, J. (2015) Deep Learning in Neural Networks: An Overview. *Neural Networks*, **61**, 85-117. https://doi.org/10.1016/j.neunet.2014.09.003

[35] Gülçehre, Ç. and Bengio, Y. (2016) Knowledge Matters: Importance of Prior Information for Optimization. *The Journal of Machine Learning Research*, **17**, 226-257.

[36] LeCun, Y., Bengio, Y. and Hinton, G. (2015) Deep Learning. *Nature*, **521**, 436-444. https://doi.org/10.1038/nature14539

[37] Deng, J., Berg, A., Satheesh, S., Su, H., Khosla, A. and Li, F.F. (2012) Large Scale Visual Recognition Challenge. https://image-net.org/challenges/LSVRC/2012/

[38] Uçar, A., Demir, Y. and Güzeliş, C. (2017) Object Recognition and Detection with Deep Learning for Autonomous Driving Applications. *Simulation*, **93**, 759-769. https://doi.org/10.1177/0037549717709932

[39] Hubel, D.H. and Wiesel, T.N. (1968) Receptive Fields and Functional Architecture of Monkey Striate Cortex. *The Journal of Physiology*, **195**, 215-243. https://doi.org/10.1113/jphysiol.1968.sp008455

[40] LeCun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W. and Jackel, L.D. (1989) Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Computation*, **1**, 541-551. https://doi.org/10.1162/neco.1989.1.4.541

[41] LeCun, Y., Boser, B.E., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W.E. and Jackel, L.D. (1990) Handwritten Digit Recognition with a Back-Propagation Network. *International Conference on Neural Information Processing Systems*, Vol. 2, 396-404.

[42] Chen, R.C. (2019) Automatic License Plate Recognition via Sliding-Window Darknet-YOLO Deep Learning. *Image and Vision Computing*, **87**, 47-56. https://doi.org/10.1016/j.imavis.2019.04.007

[43] Bhanu, B., Ravishankar, C.V., Roy-Chowdhury, A.K., Aghajan, H. and Terzopoulos, D. (2011) Distributed Video Sensor Networks. Springer Science & Business Media, Berlin. https://doi.org/10.1007/978-0-85729-127-1

[44] Berrigan, T. (2017) Busy Parking Lot—Aerial Time-Lapse.

[45] Bengio, Y. (2009) Learning Deep Architectures for AI. Now Publishers Inc., Delft. https://doi.org/10.1561/9781601982957

[46] Redmon, J. and Farhadi, A. (2017) YOLO9000: Better, Faster, Stronger. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, 21-26 July 2017, 7263-7271. https://doi.org/10.1109/CVPR.2017.690