International Journal of applied exercise physiology

ISSN: 2322-3537
Vol-11 Issue-02 July 2022

Vehicle Tracking and Speed Estimation from Traffic Videos

Mr. Potharaju Srinivas
Assistant Professor, Department of CSE,
Malla Reddy College of Engineering for Women.,
Maisammaguda., Medchal., TS, India

Abstract

Recent explosions in the processing power of commonplace computers have allowed widespread use of deep learning techniques for traffic data analysis.

security camera footage. Basic elements of traffic analysis include anomaly detection, traffic flow forecasting, vehicle re-identification, and vehicle tracking. Predicting traffic flows, sometimes known as estimating vehicle speeds, is one of the most actively studied applications in the field today. The ability to more accurately predict the need for public transportation will greatly benefit road design and reduce the likelihood of accidents. Our solution for the 2018 NVIDIA AI City Challenge aims to offer an effective method for predicting vehicle speed by combining state-of-the-art deep learning models with traditional computer vision methodologies. In this article, we provide our solution for Track 1 of the Challenge and several state-of-the-art methods for estimating vehicle speed, detecting vehicles, and tracking objects.

Introduction

The ever-increasing number of cars on the road has placed a heavy burden on road capacity and infrastructure, complicating traffic management and eventually leading to the need for further road expansion.

to issues such as traffic jams, car accidents, and polluted air. Our everyday lives are severely impacted by these issues. To mitigate their influence, a reliable and effective traffic management system is necessary. Everyday, a mountain of traffic information is compiled. Information about traffic flow, distribution, pattern, and collisions may be found in traffic data, which can then be utilized to address a wide range of traffic problems. Road segments may have their geometry and framework optimized based on traffic volume and distribution. Analyzing traffic accidents allows researchers to examine and evaluate the relationship between traffic volume and the incidence and severity of collisions.

Danger of collision. The information gathered is useful for more than only addressing traffic issues; it may also be used in research aimed at reducing pollution to the environment and the amount of fuel used. Also,

The average number of cars on the road for a certain time period, as well as the current condition of congestion, are only two examples of the statistical factors that may be evaluated to help with highway management [17].

NVIDIA has begun a series of competitions at the interface of AI and Smart City to help solve these problems. Object identification, location, and categorization were the main goals of the inaugural AI City Challenge, held in 2017. With the use of deep learning and computer vision techniques, the 2018 AI City Challenge hopes to analyze urban traffic recordings and find ways to enhance traffic flow and reduce the number of accidents that occur in cities. In particular, this session will be addressing the complex issues of urban

International journal of applied exercise physiology
ISSN: 2322-3537
Vol-11 Issue-02 July 2022

transportation. Specifically, there are three different paths to take in this competition: The goal of the first track, dubbed "Traffic Flow Analysis," is to create models that can determine the average speed of cars in films captured by fixed cameras at highways and crossings. Video abnormalities, such stopped cars or accidents, are the topic of Track2 (Anomaly Detection). The goal of Track3 (Multi-camera Vehicle Detection and Re-identification) is to identify and follow all cars seen in a group of movies taken at various city sites.

This article presents a formulation for the solution of Track1.

Vehicle recognition and tracking play a crucial role in our methodology.

Because of the lack of labeled data in this Challenge, training a vehicle identification model from scratch is challenging. Instead, we use the 3D Deformable model [16] for vehicle recognition by transfer learning and carry out inference on our dataset. Since the model by Bhandary et al. [2] achieved comparable performance to the 3D Deformable in the 2017 competition, we thought about using it instead. Vehicle identification performance is compared between the two models using mean Average Precision, and this is done by extracting all frames from one of the Track 1 movies and measuring the models' performance on the frames (mAP). In this experiment,

The results of the experiment reveal that the 3D Deformable model [16] is superior than the model of Bhandary et al. [2] by a margin of 74% mAP.

Our follower follows a detect-then-track strategy. In this way, the tracker's effectiveness is heavily reliant on the precision of its detection. When cars are spotted in a given frame, we use those attributes to help us identify them in the next frame. These characteristics' shifting positions within the frame provide crucial data for determining the speed of the moving object.

Related Works

In order to construct a reliable vehicle speed estimate model, vehicle tracking is essential. Numerous strategies based on traditional computer vision and machine learning have been created.

Strategies for detecting and following objects. Traditional optical-flow algorithms and motion vector estimates were used by Kale et al. (2015) [11] to address the object tracking issues.

They suggested an optical-flow technique for detection and motion vector estimation for estimating velocity, otherwise known as a "track-by-detect" approach. The non-stationary environment was addressed with a reinforcement learning-based framework and a Kalman Filter, which was supplied by Geist et al. (2009) [8]. The study claims that the challenge of tracking video objects may be recast as one of anticipating the bounding box of the item of interest at each frame. A recurrent convolutional neural network model was created by Zhang et al. (2017) [22] and was taught using a Reinforcement Learning method. In his 2012 article, "A Simple and Detailed Explanation of Kalman Filters," Faragher [6] provided a clear and comprehensive overview of the method. This work presents the underlying assumption of Kalman Filters and provides the procedure for mathematically modeling a tracking issue. The tracking issue is nonlinear and non-Gaussian, and Brasnett et al. (2005) developed a method to solve it by combining features with a particle filtering technique.

The subject of vehicle speed detection has also seen a great deal of study, with many different methods being tested.

An strategy presented by Rad et al. (2010) [15] for predicting traffic speed from digital video collected with a stationary camera involves comparing the vehicle location between the current frame and the preceding frame. Geometric formulae were used to calibrate the cameras.

The method developed by Rad et al. has an average inaccuracy of 7 km/h for the detected vehicle speed and has the potential to be applied to other

ISSN: 2322-3537
Vol-11 Issue-02 July 2022

application areas. Employing real-time tracking methods, Ferrier et al. (1994) [7] obtained many metrics, including vehicle speed, by using the image's motion characteristics and information on the projection between the ground plane and the image plane.

Scene-specific adjustment of the dynamics has also been utilized to improve the tracker's ability to forecast the position of a target. Recently, digital techniques have been used by Yamazaki et al. (2008) [21] to simply separating the cars and shadows from two successive aerial photos, the speed of passing vehicles may be determined. To determine velocity, we pair together moving objects from these photos by distance, order, and size; then, by utilizing distance between related cars and time lag.

The coordinates in the picture domain have been mapped into the physical world in an effort to improve accuracy, as shown by Wu et al. (2009) [20]. A pair of QuickBird panchromatic and multi-spectral photos were employed for speed detection by Liu and Yamazaki [12]. It was utilized by Ger'at et al. [9] to estimate velocities using a hybrid of Kalman filters and opticalflow methods. The former helps with the issue of transient occlusions, while the latter ensures more precise speed delivery.

With the use of a distance-to-pixels mapping, Wang [19] demonstrated a method for detecting moving objects in a video. Features of moving cars were extracted using three-frame differencing and background differencing in this technique. Then, centroids of vehicles were used to determine their locations.

Dataset

In contrast to the 2017 AI City Challenge, which featured a sizable group annotation effort for the purpose of training supervised models to solve traffic-related issues, the 2018 AI City Challenge does not, this year's competition put greater emphasis on transfer learning techniques and did not include any annotations.

The accessible data was collected by fixed cameras at highways and city streets. Sample data from the dataset obtained at a highway interchange is shown in Figure 1. More information on the dataset may be found below.

Track 1 data includes 27 1920x1080p films that are one minute long and were captured at 30 frames per second (fps). The first two movies are taken on a highway, while the third and fourth videos are taken at intersections.

There are one hundred movies in the Track 2 dataset, and each one is around 15 minutes long and captured at 800x410 resolution and 30 frames per second.

There are fifteen movies, all shot in 1080p at 30 frames per second and originating from four distinct sites, included in the Track 3 collection. From half an hour to almost an hour and a half, that's how lengthy each video is.

Geometry Information & Speed Limit

It is possible to estimate the maximum speed limit for each road stretch from the metadata files that accompany each video. This is due to the cameras' perspective.

At positions 1 and 2, cameras only record traffic going in opposite directions, while at locations 3 and 4, they record vehicles going through a four-way junction.

International journal of applied exercise physiology

ISSN: 2322-3537
Vol-11 Issue-02 July 2022

Figure 1. A sample of images captured at traffic intersections and highway.

Table 1. Geometry & speed limit data for track 1.

Loc.	Latitude	Longitude	Direction	Speed
1	37.316788	-121.950242	E → W	65 MPG
2	37.330574	-122.014273	NW → SE	65 MPG
3	37.326776	-121.965343	NW → SE	45 MPG
3	37.326776	-121.965343	NE → SW	35 MPG
4	37.323140	-121.950852	N → S	35 MPG
4	37.323140	-121.950852	E → W	35 MPG

Figure 2. VGG image annotation tool.

four distinct trajectories Track 1 video locations and speed limits in one direction are summarized in Table 1. The direction in which we are heading has the comparable to the speed limit shown on the road signs.

Annotation tool

In order to compare vehicle detection methods, we manufactured some ground-truth data. The VGG Image Annotator(VIA) [5] was the tool we utilized to annotate the data. To that end, we provide this instrument has sophisticated features like copy/pasting a bounding box and is browser-based. VGG is a straightforward method for labeling challenge data due to the consecutive nature of the frames we annotated and the low degree of variation in vehicle location between frames. A screenshot of the VGG software is seen in Figure 2.

Methodology

Below, we detail the procedures we've developed to identify moving objects in traffic footage and calculate their speeds. Using an initial phase of detection followed by object tracking, our technique any vehicle detection algorithm's object detections as input.

Next, we'll talk about how we plan to make it happen excellent detections, distinguishing vehicle tracks, and velocity estimation.

Vehicle Detection

Sadly, we were unable to train a vehicle identification model since the 2018 AI City Challenge dataset did not include any ground-truth detection or tracking annotations customized for this data set model. We instead use trans- fer learning, capitalizing on state-of-the-art deep learning models that have already been trained for this purpose.

Videos in Tracks 1 and 3 of the dataset were utilized in the item identification, location, and classification problem at the 2017 AI City Challenge [14]. Since then, we've settled on using the top two models from that competition: the 3D Deformable model by Tang et al. [16] and the model that Bhandary et al. [2] developed in our own lab for the competition. For each frame, the output of both models is a set of bounding-boxes, the class of the item they are most confident in detecting, and a confidence score. Car, SUV, small truck, medium truck, big truck, pedestrian, bus,

International journal of applied exercise physiology

ISSN: 2322-3537
Vol-11 Issue-02 July 2022

van, group of people, bicycle, motorbike, traffic light - green, traffic signal - yellow, and traffic signal - red were only few of the categories that were searched for in the 2017 challenge. We filter the detector output in the following ways to increase the value of the detections we feed into our algorithm:

If a detection's confidence is below a certain level, discard it.

• Discard any detections that don't pertain to vehicles, and preserve just those that identify cars, SUVs, trucks, buses, vans, bicycles, and motorcycles.

• Remove from consideration bounding boxes inside the same frame that don't overlap by at least a certain threshold, as determined by the Intersection-over-Union (IOU) score. When the IOU of the detection with an already chosen bounding box is more than, the detection is filtered as it moves left to right across the frame from top to bottom.

Vehicle Tracking System

The localization findings from the vehicle detection techniques outlined in Section 4.1 provide the basis for our vehicle tracking algorithm, which is then further improved by the use of optical-flow based characteristics to give reliable vehicle trajectories.

Detection-Based Tracking

Since the original film was recorded in 1080p and 30 fps, very little pixel shifting occurs between frames into the next. Therefore, we provide a practical technique for defining initial object tracks using just the overlap of observed object bounding boxes across frames. Specifically,

Each item in a frame is given the ID of the previously unseen object with the greatest IOU score in the prior h frames, provided that this score is greater than some threshold i. This is done in descending order of detection confidence.

However, when the detector fails to locate an item for more than h frames, the tracking-by-detection method is prone to a high rate of ID changes, which is not ideal for our environment.

Speed estimate via bounding boxes is also erroneous since detectors often produce loose localisation bounding boxes around the identified objects. Some instances of incorrect detection are shown in Figure 3, such as a tree and a building being misidentified as cars and large gaps between the bounding-box and the detected item.

Tracking-by-Flow

Computing the optical flow for a sparse collection of observed object characteristics, particularly Shi-Tomasi corners [18], allows us to improve upon the basic detection-only based tracking.

Bouguet's iterative Lucas-Kanade approach using pyramids [3, 13]. Cor- ners, which are used for flow estimates, are tiny uv areas in the picture with considerable change in intensity in all directions. Instead of calculating the matrix's eigenvalues 1 and 2, like the Harris corner detector [10] does to discover possible corners, the Shi-Tomasi corner detector uses a more sophisticated method.

$$M = \sum_{x,y} w(x,y) \begin{bmatrix} \mathbf{I}_x\mathbf{I}_x & \mathbf{I}_x\mathbf{I}_y \\ \mathbf{I}_x\mathbf{I}_y & \mathbf{I}_y\mathbf{I}_y \end{bmatrix},$$

where the x and y derivatives of the frame are denoted by Ix and Iy, and w is a function that quantifies the relative importance of the derivative windows to the whole. While

In Harris's mind, a window might have a corner if

$$R_1 = \lambda_1\lambda_2 - k(\lambda_1 + \lambda_2)^2$$

windows where R2 = min(1, 2) was over a certain threshold were more likely to have unique corner characteristics, as shown by Shi and Tomasi. In

International journal of applied exercise physiology
ISSN: 2322-3537
Vol-11 Issue-02 July 2022

Using our technique, we don't filter out frames before selecting corner points, and we then restrict those points to just being inside the region of at least one bounding box supplied by our vehicle detector.

The Lucas-Kanade technique relies on the assumptions that the intensity of a selected point does not vary from one frame to the next and that the point and its neighbors move in the same way in each frame.

It calculates the distance traveled by factoring in the positional shift of the target point and its surrounding features between the two points of interest frames. Our technique remembers the detected point positions in the most recent t frames for each corner point associated with a tracked object. The little snippets of music that were uncovered serve as a signal for calculating the speed of the cars represented by the four corners. Tracklet samples from each of the four challenge camera positions may be seen in Figure 4.

Speed Estimation

Several robust assumptions underpin our strategy, which is data-driven in its approach to calculating vehicle speeds.

To begin with, the traffic camera shouldn't move while filming, which should be taken into consideration for the 2018 AI City Competition. Second, we think at least one car in the clip is traveling at the legal maximum speed for the roads shown. Our system predicts vehicle speeds as a function of local motion and smax, the maximum speed at which any vehicle is estimated to travel in the clip. That is, we use the greatest historical corner point movement inside the tracklets that are linked to the vehicle as a measure of its local mobility, i.e.

$$\Delta m = \underset{i}{\mathrm{perc}_p}\left(\max_{j=2}^{|T_i|}(\,\|T_i(j) - T_i(j-1)\|_2\,)\right),$$

where Ti is the ith tracklet found for the car, Ti(j) is the jth point in the tracklet's history, and |Ti| is the tracklet's size. Perc's Function results in the pth percentile of all tracklet changes being calculated. Some outliers may be filtered out by considering the distribution of tracklet estimated local motions, which can aid with corner point identification in some tracklets that is inaccurate.

This frame has an inconsistent local movement to vehicle movement ratio. Since each camera captures the roadway from a somewhat different perspective, a 34 projection matrix P must be estimated using a set of vanishing landmark points and previous knowledge of camera settings to standardize item motions [4]. Since we cannot determine the camera parameters, we instead estimate the projection by training a set of functions over horizontal tiles of the input video. It seems to reason that cameras should face the horizon. Therefore, cars going across the screen from top to bottom (or vice versa) will seem to slow down as they approach the horizon, even though their actual speed remains constant. On the other hand, the ratio of local movement to speed remains generally constant for vehicles moving across the frame from left to right.

The modification of the local movement to vehicle speed relationship will be insignificant for tiles small enough. Thus, we think of a model that predicts the speed, called a predicted speed (PS) model.

Figure 3. Detection error. (*Best viewed in color*)

International Journal of applied exercise physiology

ISSN: 2322-3537
Vol-11 Issue-02 July 2022

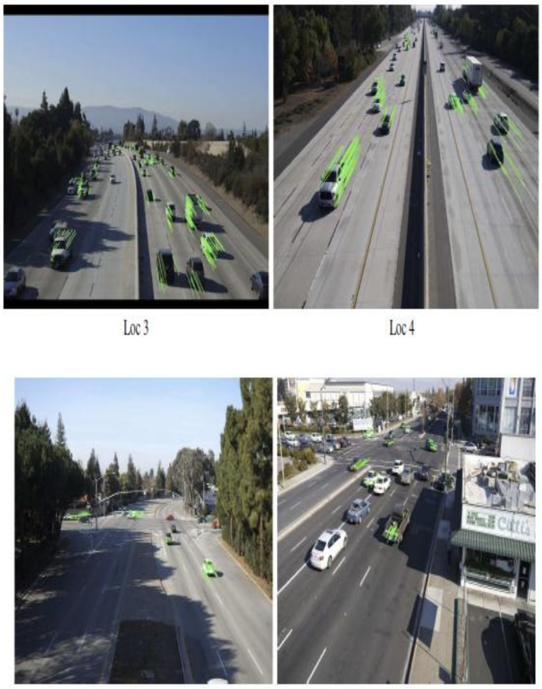

Figure 4. Tracklets obtained through optical-flow estimation. (*Best viewed in color*)

of a vehicle, in each tile, as

$$s = \frac{\Delta m}{\max\limits_{T} \Delta m} \times s_{\max},$$

where max T m is the greatest possible regional motion in any snippet of a vehicle's path across the tile, as seen from inside the present vehicle's time frame t. We further remove extremes by averaging the highest predicted vehicle speed for a window of time that is at most h frames long. Finally, we restrict the projected speed to be between zero and the maximum speed.

Especially in highway traffic with little to no congestion, as shown in the Loc 1 and 2 films of the challenge, many vehicles travel at consistent speeds across the frame. As a result, we think it's important to Following filtering based on confidence and track overlap, a second constant speed (CS) model is used, which gives every vehicle in the movie the input smax speed.

Experiments

We manually annotated the first 250 frames of Video 1 at Location 1 to serve as our training set before applying either of the two detection models we explored in Section 4.1.

Table 2. Detection performance.

Mddel	mAP_0	mAP_1
Bhandary [2]	0.34	0.34
3D Deformable Model [16]	0.28	0.74

the real deal, having both models recognize automobiles in the video and then contrasting their mean Average Precision (mAP). We eliminated models in which we had little confidence by table 2: mAP 0 and mAP 1 represent detections with confidence ratings of 0.0 (no filtering) and 0.1, respectively. Given an appropriate confidence level, the findings seem to favor the 3D Deformable model above that of Bhandary et al. We submitted 3D Deformable models to the Challenge with a minimal confidence score between 0.01% and 0.050%.

The NVIDIA Titan X general-purpose graphical processing unit (GPGPU) and 16 gigabytes of random-access memory were used to run all tests. Using the maximum speed limitations shown in Table 1, we conducted tests at smax=5, 10, and 15 miles per hour at each site. We implemented an IOU threshold of 0.9 for duplicate record screening. We used overlapping detections with a minimum IOU of 0.7, tracked up to h = 10 corner points per tracklet, and assessed local motion using the top p = 80% of tracklet segments in our tracklet identification process.

S1 = DR(1NRMSE), where DR is the vehicle detection rate for the set of ground-truth vehicles and NRMSE is the RMSE score across all detections of the ground-truth vehicles, normalized via minmax normalization with respect to the RMSE scores of all best solutions of other teams submitting solutions to Track 1 of the 2018 AI City

ISSN: 2322-3537
Vol-11 Issue-02 July 2022

Challenge. NVIDIA staff members drove ground-truth cars past the camera vantage sites, measuring instantaneous speeds using a GPS device. You can read more about the datasets available for the Challenge and the criteria used to judge them at [1].

Results

We apply our tracking and speed estimation algorithms to the Track 1 footage from the 2018 NVIDIA AI City Challenge and examine the resulting data. We first detail the outcome of the Challenge and then discuss the ways in which our model might be improved.

Submission of a Challenge

Our highest-scoring Challenge entry received a DR of 1.0 and an RMSE of 12.1094 for a total S1 score of 0.6547, placing us in second place overall but a whole 0.0017 points behind the first-place team.

We found that although our model was excellent at detecting targets, the speed estimate part of our strategy was inaccurate enough to compete with the best teams, team48 and team79, which got S1 scores of 1.0 and 0.9162, respectively.

Given that team48 has a perfect score of 1.0, got the lowest RMSE score while simultaneously having the highest detection rate score (1.0).

Model Analysis

The best model we tested assumed a constant speed between 70 and 70 kilometers per hour at sites 1 through 4, and 50 and 30 kilometers per hour at places 5 through 6. Unfortunately, the accuracy of our speed prediction models dropped as compared to the CS model, both in terms of detection rate and RMSE. Both the PS (a) and CS (b) models were run with the identical settings, and their anticipated speeds for two consecutive frames from Location 1 are shown in Figure 5. Tracklet detection, which is based on optical flows, is used by the PS model to estimate vehicle speeds, and a detection is only reported if the speed estimate is accurate. Due to this discrepancy, the PS model does not accurately reflect the number of cars identified by the CS model.

To see where the PS model falls short, we picked 10 tracks at random from each site, ranging in length from 45 to 60 frames, and plotted them in Figure 7. Certain paths exhibit the smooth transitions typical of typical traffic, while others exhibit rapid increases in speed, which may be an indication that the corner feature detector is picking the wrong corner in some instances.

By visualizing the distributions of speed ranges (difference between maximum and lowest speed) in tracks of films over all four sites, we are able to further demonstrate the inherent uncertainty in the PS model's speed predictions. Most vehicles are only visible for a few seconds as they pass through the frame, so we can assume that their speed is fairly consistent and their fluctuation is minimal. Each quadrant of Figure 6 displays a uniform random sample from the speed range distribution in the supplied movie by employing a line for each video at a specific place.

Our model displays extreme variability, with over 20% of cars reporting more than 15 miles/hour of fluctuation, although some variability is to be anticipated owing to typical traffic. Location 3 fares much worse since the wind or bridge vibrations constantly shifted the camera, reducing the clarity of the captured films.

Conclusion & Future Work

Using a detect-then-track paradigm and an optical-flow-based, data-driven speed calculation, we present here a model for monitoring automobiles in traffic recordings method, and outlined our answers to the first set of challenges in this year's NVIDIA AI City Challenge. Although our model fared well, we were unable to keep up with the other Challenge entrants due to its high degree of unpredictability. We did not have enough time to evaluate our approach to existing detectthen-track algorithms, but we will do so in future work. Ad-

(a) Predicted Speed Model

(b) Constant Speed Model

Figure 5. Speed estimates of the predicted (a) and constant (b) speed models. *(Best viewed in color)*

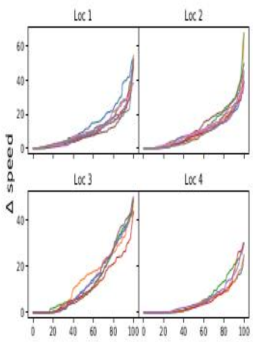

Figure 6. Speed range distribution.

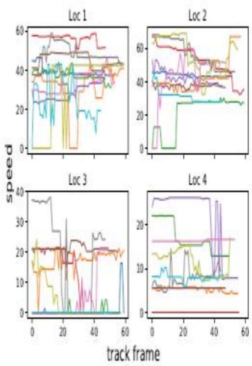

Figure 7. Speed of random tracks.

Furthermore, we want to look into smoothing approaches for the projected vehicle speeds, which may enhance the performance of the model.

References

[1] Data and Evaluation nvidia ai city challenge. https: //www.aicitychallenge.org/?page_id=9. Accessed: 2018-04-01.

[2] N. Bhandary, C. MacKay, A. Richards, J. Tong, and D. C.Anastasiu. Robust classification of city roadway objects fortraffic related applications. 2017.

[3] J.-Y. Bouguet. Pyramidal implementation of the lucaskanade feature tracker description of the algorithm, 2000.

[4] B. Caprile and V. Torre. Using vanishing points for cameracalibration. *Int. J. Comput. Vision*, 4(2):127–140, May 1990.

[5] A. Dutta, A. Gupta, and A. Zissermann. VGG image anno-tator (VIA). http://www.robots.ox.ac.uk/ gg/software/via/,2016. Accessed: January 10, 2018.

[6] R. Faragher. Understanding the basis of the kalman filter via a simple and intuitive derivation [lecture notes]. *IEEE Signal Processing Magazine*, 29(5):128–132, Sept 2012.

[7] N. J. Ferrier, S. Rowe, and A. Blake. Real-time traffic monitoring. In *Proceedings of 1994 IEEE Workshop on Applica- tions of Computer Vision*, pages 81–88, 1994.

[8] M. Geist, O. Pietquin, and G. Fricout. Tracking in reinforcement learning. In *International Conference on Neural Infor- mation Processing*, pages 502–511. Springer, 2009.

[9] J. Ger´at, D. Sopiak, M. Oravec, and J. Pavlovicov´a. Vehicle speed detection from camera stream using image processing methods. In *ELMAR, 2017 International Symposium*, pages 201–204. IEEE, 2017.

[10] C. Harris and M. Stephens. A combined corner and edge detector. In *Proceedings of the 4th Alvey Vision Conference*, pages 147–151, 1988.

[11] K. Kale, S. Pawar, and P. Dhulekar. Moving object tracking using optical flow and motion vector estimation. In *Reliability, Infocom Technologies and Optimization (ICRITO)(Trends and Future Directions), 2015 4th Interna- tional Conference on*, pages 1–6. IEEE, 2015.

[12] W. Liu and F. Yamazaki. Speed detection of moving vehicles from one scene of quickbird

images. In *Urban Remote Sensing Event, 2009 Joint*, pages 1–6. IEEE, 2009.

[13] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision (darpa). In *Proceedings of the 1981 DARPA Image Understanding Workshop*, pages 121–130, April 1981.

[14] M. Naphade, D. C. Anastasiu, A. Sharma, V. Jagrlamudi,\H. Jeon, K. Liu, M. C. Chang, S. Lyu, and Z. Gao. The nvidia ai city challenge. In *2017 IEEE SmartWorld Conference*, SmartWorld'17, Piscataway, NJ, USA, 2017. IEEE.

[15] A. G. Rad, A. Dehghani, and M. R. Karim. Vehicle speed detection in video image sequences using cvs method. *In-ternational Journal of Physical Sciences*, 5(17):2555–2563,2010.

[16] Z. Tang, G. Wang, T. Liu, Y. Lee, A. Jahn, X. Liu, X. He,and J. Hwang. Multiple-kernel based vehicle tracking using3d deformable model and camera self-calibration. *CoRR*,abs/1708.06831, 2017.

[17] P. H. Tobing et al. Application of system monitoring and analysis of vehicle traffic on toll road. In *Telecommunication Systems Services and Applications (TSSA), 2014 8th Inter- national Conference on*, pages 1–5. IEEE, 2014.

[18] C. Tomasi and J. Shi. Good features to track. In *Proc. IEEE Conf. on Comp. Vision and Patt. Recog.*, pages 593– 600, 1994.

[19] J.-x. Wang. Research of vehicle speed detection algorithm in video surveillance. In *Audio, Language and Image Pro- cessing (ICALIP), 2016 International Conference on*, pages 349–352. IEEE, 2016.

[20] J. Wu, Z. Liu, J. Li, C. Gu, M. Si, and F. Tan. An algorithm for automatic vehicle speed detection using video camera. In *Computer Science & Education, 2009. ICCSE'09. 4th Inter- national Conference on*, pages 193–196. IEEE, 2009.

[21] F. Yamazaki, W. Liu, and T. T. Vu. Vehicle extraction and speed detection from digital aerial images. In *Geoscience and Remote Sensing Symposium, 2008. IGARSS 2008. IEEE International*, volume 3, pages III–1334. IEEE, 2008.

[22] D. Zhang, H. Maei, X. Wang, and Y.-F. Wang. Deep reinforcement learning for visual object tracking in videos. *arXiv preprint arXiv:1701.08936*, 2017.