Deep speed estimation from synthetic and monocular data*

João Barros[1] and Luciano Oliveira[2]

***Abstract*— Current state-of-the-art in speed measurement technologies includes magnetic inductive loop detectors, Doppler radar, infrared sensors, and laser sensors. Many of these systems rely on intrusive methods that require intricate installation and maintenance processes that hinder traffic while leading to high acquisition and maintenance costs. Speed measurement from monocular videos appears as an alternative in this context. However, most of these systems present as a drawback the requirement of camera calibration – a fundamental step to convert the vehicle speed from pixels per frame to some real-world unit of measurement (*e.g.* km/h). Considering that, we propose a speed measurement system based on monocular cameras with no need for calibration. Our proposed system was trained from a synthetic data set containing 12,290 instances of vehicle speeds. We extract the motion information of the vehicles that pass in a specific region of the image by using dense optical flow, using it as input to a regressor based on a customized VGG-16 network. The performance of our method was evaluated over the Luvizon's data set, which contains real-world scenarios with 7,766 vehicle speeds, ground-truthed by a high precision system based on properly calibrated and approved inductive loop detectors. Our proposed system was able to measure 85.4% of the speed instances within an error range of [-3, + 2] km/h, which is ideally defined by the regulatory authorities in several countries. Our proposed system does not rely on any distance measurements in the real world as input, eliminating the need for camera calibration.**

I. INTRODUCTION

Several countries use speed control systems to enforce road speed limits, preventing then traffic accidents. Vehicle speed can be used to determine possible accidents, hazardous areas in rush hour, as well as wrong or unsafe driver behavior [1], [2]. Therefore speed measurement becomes one of the main allies of traffic surveillance systems mainly for their management and enforcement.

Speed measuring systems can be divided into two main categories [3]: Active, which measures the effect of signals transmitted to a vehicle passing on a highway, and passive, which comprises video systems where there analyses of consecutive video frames take place in order to track a vehicle and thereby measure its speed. There are three devices that are commonly used in vehicle speed detection on the roads: Induction loop, laser, and radar. In active measuring systems, sensors that perform speed measurement are used to trigger a video camera to record vehicle plates that exceed the speed limit. Such sensors are usually expensive and require extensive calibration and maintenance, besides having complex installation requisites [4]. On the other hand video-based systems have become popular due to technological advances in cameras and computer devices [5]. Additionally the cost-effectiveness of these systems makes it an advantage over traditional systems. This way it is possible to make speed measurement simpler and cheaper by using information extracted from video frames.

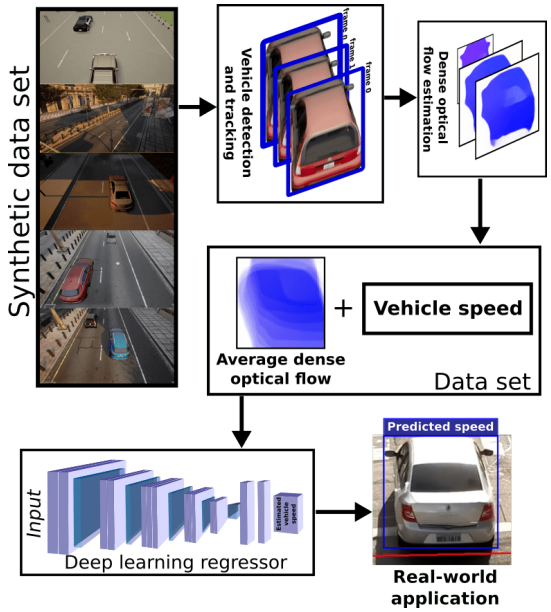


Fig. 1: Our pipeline for speed estimation through monocular camera. We created a synthetic data set and from it we extract the dense optical flow. The average of the dense optical flow is calculated and used as an input in a deep learning regressor. Our regressor is a VGG-16-based network that we have adapted for this purpose. This regressor is validated against a real-world data set [6].

For speed measurement, it is necessary to obtain the time values, wherein a vehicle travels a route, and the traveled distance on the road plane. The time measurement is the trivial part to be computed. However, to get the distance, one needs to know the actual distance between two points on the road. For that, calibration is demanded. Calibration is the main step in the speed measurement process. Generally, three steps are required to obtain a vehicle speed from video: (i) Vehicle detection where each vehicle passing in the scene is located, (ii) vehicle tracking where the detected vehicle is tracked until it leaves the scene, and (ii) camera calibration to

*This work was supported by the Bahia State Research Founding Agency (FAPESB), grant No. APP0015/2016. Luciano Oliveira has a research scholarship from Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), grant 307550/2018-4. João Barros has a scholarship supported by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) [Finance code 001].

[1]João Barros and [2]Luciano Oliveira are with the Intelligent Vision Research Lab (Ivisionlab), Universidade Federal da Bahia http://ivisionlab.ufba.br/people.

find the parameters needed to transform the vehicle trajectory shift, in pixels, to some real-world unit of measurement. Calibration methods for this context can be manual or automatic. Manual calibration usually requires distance measurements on the road plane (there is the assumption that road can be approximated by a plane) [7], [8].

Automatic calibration usually takes into account the mean vehicle dimensions [9] [10]. The major limitation of current works is that, for each capture environment, camera calibration is required to calculate speed in terms of actual measurement values (*e.g.* km/h) rather than pixels per frame. Then, for each environment in which the estimation will be carried out, it is necessary to obtain some distance measurement on the road to be used as an input for calibration. Even automatic calibrations need some real measurement, which in this case is the assumption that all vehicles have roughly the same length. The vehicles' average dimension is used as a real world reference to obtain a scale factor to compute the real vehicle speed.

Contrary to the current state-of-the-art methodology, here we use only synthetic data to build our speed measurement system, without any real-world measurement, eliminating the use of calibration or any manual input for this purpose. An illustration of our proposed system pipeline is depicted in Fig. 1. We explore a deep learning regressor over our synthetic data set, while able to estimate the speed of vehicles in the real world with relative precision, only with the use of visual motion features and dense optical flow. For that, we used a Faster R-CNN [11] combined with a DeepSORT tracker [12]. FlowNet2 [13] is used to extract the dense optical flow that along with the vehicle speed ground truth feed a VGG-16 [14] to train a regressor for the final vehicle speed. We validated our system in the Luzivon's data set [6], which consists of approximately 5 hours of 7,766 real-world speed annotation, captured in different weather and recording conditions, and we achieved competitive results by using only synthetic data. In [6], Luvizon et al. found an average error of -0.5 km/h, with a standard deviation of 1.36 km/h, with 96% of measurements in an acceptable error range, considered from -3 to $+2$ km/h. We, without using any calibration method or measurement in the real world, achieved an average error of -0.77 km/h, with a standard deviation of 2.66 km/h and 85.4% of the measurements being in the ideal error range. Camera calibration for each region where there will be speed measurement is not necessary in our system, avoiding stopping traffic, or other disturbances necessary to perform this procedure. Our speed regressor is built entirely using synthetic data, which also avoids any intervention in the road infrastructure.

II. RELATED WORK

A speed measurement system from a monocular camera requires an extensive data set for its development. However there are few vehicle speed data sets available for research purposes. These data sets found in the literature are those of [15] and [6].

A. Data sets in the literature

There are only two data sets on the theme of vehicle speed measurement in the literature available academically. The first is [6] whose data set we chose to validate our system. Luvizon et al. [6] gathered a data set with approximately five hours of video captured by a fixed overhead camera with frame resolution of 1920×1080 pixels and 30.15 FPS. The ground truth speeds were obtained from a high precision speed meter based on inductive loop detectors, properly calibrated and approved by the Brazilian national metrology agency (Inmetro). 7,766 vehicles were annotated as valid since they have both a visible license plate and an assigned speed. There are videos in different weather and recording conditions: High quality, frames affected by natural or artificial noise, frames affected by severe lighting conditions, motion blur, and rain.

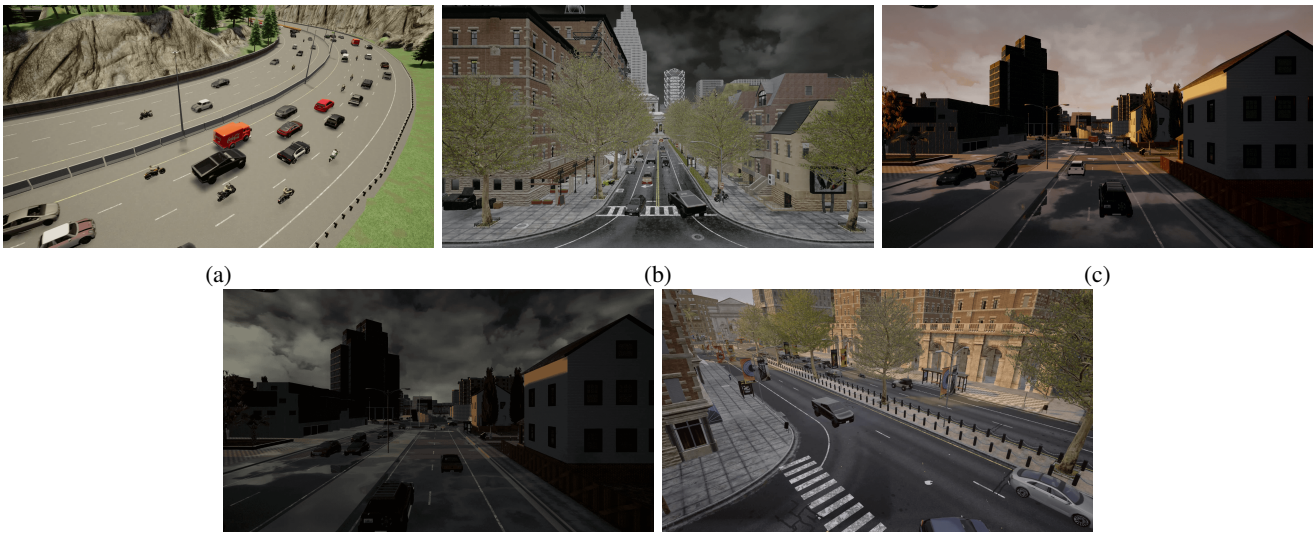
Sochor et al. [15] built a data set containing 20,865 valid instances of vehicles along with the corresponding reference speed values from three points of camera view. The reference speed values were obtained by a pair of LiDAR, positioned at a known distance between them, placed on the side of the road, perpendicular to the direction of traffic flow, and synchronized by a GPS. The average vehicle speed is calculated from the distance over the timestamps recorded at the time the vehicle enters the first and second laser. This data set present a particular limitation: If there are two vehicles near the first LiDAR, only the first vehicle identified by this LiDAR will be tracked, ignoring the second vehicle. Then, speed measurement is applied to a single vehicle. Speed measurement is not possible when there are multiple vehicles on the road. Luvizon et al. [6] apply speed measurement to multiple vehicles passing on the road at the same time. Because of these factors, we chose to use the Luzivon's data set in the validation process.

B. Vehicle detection and tracking

The purpose of detection is to locate vehicles passing on the highway. Detection is usually based on the location of vehicle license plates or the vehicle itself. Mask R-CNN [16], Faster R-CNN [11], and YOLO [17] are those classifiers commonly used for vehicle detection in speed estimation [18], [19], [20]. Other works use handcrafted methods [21], [22] to perform the detection, such as background subtraction [7]. Generally, DeepSORT [12], IoU [23], or KLT Tracker [24] is used to accomplish the tracking.

C. Camera calibration and vehicle speed estimation

The goal of camera calibration in the speed measurement application is to measure the distance, in meters or another unit of measurement, between two arbitrary points on a road. Much of the work in the literature uses some real-world measurement, be it road (calibration with manual input) or average vehicle size (automatic calibration), to obtain the estimated speed in a real-world unit of measurement. They usually perform vanishing point detection [25], [26], [27], [20] or perspective rectification using known points on the road, as is the case with [6], [18].



(a) (b) (c)

(d) (e)

Fig. 2: Samples of different scenarios obtained in the CARLA simulator in different weather conditions: In (a), an expressway scenario with clear noon weather; in (b), an urban perimeter scenario with hard rain noon weather; in (c), another urban perimeter scenario with wet sunset weather; in (d), an urban perimeter with wet cloudy sunset weather; and urban perimeter with sunset weather, in (e).

There are some methods that do not use calibration to recover the scene's scale. Dong et al. [4] use Sochor's data set [15] to measure vehicle speed using an RGB image sequence plus Optical Flow as input to a 3DConvNets [28], and Kampelmühler et al. [29] use a combination of optical flow, RGB and depth to feed an MLP as a regressor to estimate vehicle speed from a camera installed in an autonomous car.

Dong et al. [4] obtain the average speed from the sequence of vehicles in the scene, making it difficult to identify the individual speed of the vehicle when there are multiple vehicles. Kampelmühler et al. [29] make the estimation for autonomous cars, where a camera is attached to the vehicle in order to make the captures. Our work measures the individual speed of vehicles, aiming to measure this from monocular video surveillance cameras. In addition, we do not use any calibration method or any real-world measure to obtain the scale factor in the speed estimation process.

III. VEHICLE SPEED ESTIMATION

A. *Synthetic data set generation*

To build our speed estimator, we created a synthetic data set comprised of 12,290 instances of vehicles together with their corresponding speeds. We use the CARLA simulator [30] to create the synthetic data set. CARLA is an open-source autonomous driving simulator, focused on a series of tasks involving autonomous driving problems. In this simulator, it is possible to obtain vehicle speed data along with its route. The use of the simulator when generating synthetic data sets allows the construction of several scenarios that can reflect the real world, by generating varied samples with variations in weather and light conditions, which can be difficult and costly to be captured in the real world. CARLA offers 8 city maps (scenarios) that can be customized with 18 types of vehicle models, including bicycles, motorcycles, cars and trucks. We used three scenarios to make the capture, two of which are of urban perimeters, in which vehicles travel at low and medium speeds, and one of expressway in which vehicles travel at a faster speed. We apply five different weather conditions to obtain the data: Clear noon, sunset, hard rain noon, wet cloudy sunset, and wet sunset. We

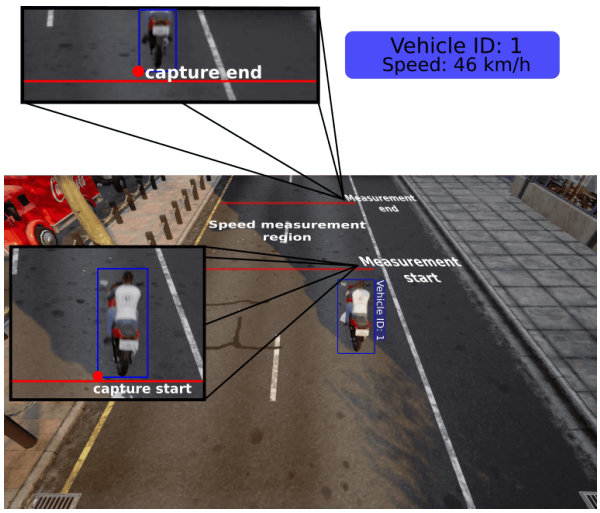


Fig. 3: Region where vehicles are captured, optical flow is estimated and average speed is obtained. The optical flow and speed data will compose the synthetic data set. The milestones for the beginning and end of capture are determined from the moment the lower left point of the vehicle's bounding box (red dot) passes the lines that demarcate the measurement region.

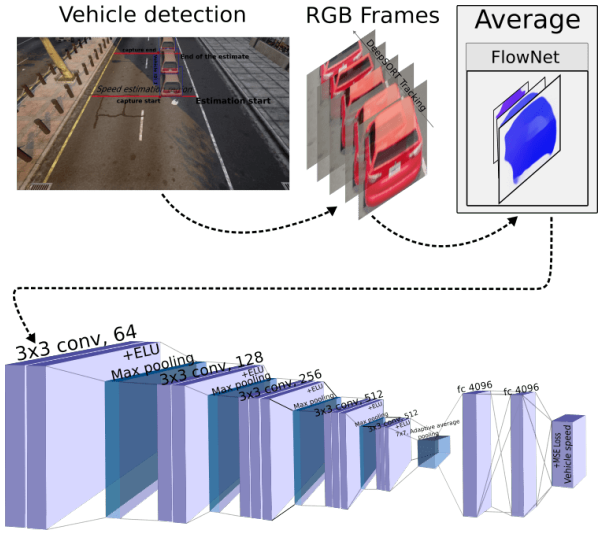


Fig. 4: General speed estimation pipeline: First, the bounding boxes captured on the Faster R-CNN, are used as input to DeepSORT. From the tracking result, we estimate the dense optical flow using FlowNet2 [13]. An estimation of the average dense optical flow is passed as input to a modified VGG-16. In this network, we add an adaptive average pooling before the three Fully-Connected (FC) layers. The first two FC layers contain 4,096 neurons, while the last one has a single neuron, which corresponds to the estimated speed. Instead of RELU, we use ELU, as an activation function, in the convolutional layers.

generate 32 videos ranging from 10 to 40 minutes, totaling approximately 9 hours and 30 minutes of video. Figure 2 illustrates these scenarios and weather conditions.

We use the same principle as [6] and an inductive loop detector to generate the data set from the CARLA simulator. In the simulator scenario, we demarcate a speed estimation region and place a camera aimed at the region. Thus, there is a line defining when the vehicle enters and another line, when the vehicle exits. The average vehicle speed between these two lines is recorded for each vehicle that passes through that region. The input video is captured by a single fixed overhead camera, positioned so as to view the rear of the vehicle, in a single location for each synthetic scenario. This speed is obtained from the simulator. We use Faster R-CNN[11], with ResNet101[31] and Feature Pyramid Network [32] as the backbone, and DeepSORT [12] as the baseline methods for vehicle detection and multi-object tracking, respectively. The tracking system allows to identify which vehicle bounding boxes from Faster R-CNN are part of a particular vehicle. From the moment the lower left point of the vehicle's bounding box passes the first line that marks the beginning of the measurement, the estimation of the dense optical flow is performed. This estimation will last until the lower left point of the vehicle's bounding box passes the last line that marks the end of the measurement. Fig. 3 illustrates this procedure. We extract the optical flow information using a state-of-the-art neural network architecture, FlowNet2 [13]. After obtaining the motion data, we average the optical flow tracked within the speed measurement region. The result of this average will be used as an input to our deep learning regressor.

B. Vehicle detection and tracking

We retrained Faster R-CNN from the COCO data set [33] by adding synthetic samples for training. As we started from the tracking by detection paradigm, by using the DeepSORT tracker, we also added samples from another data set in which we captured several vehicles at a long distance from the camera in order to strengthen the accuracy of our detector. We captured this data set in the city of Salvador, Bahia. Car, bus, motorcycle and truck vehicles are included in our annotations. Vehicle bounding boxes, obtained from Faster R-CNN, are used as input for DeepSORT, which combines Kalman filter with frame-by-frame data association. It also uses appearance feature vector to retrieve the information of the tracked object. We opted to use the customization of Shishira [34], which uses the Siamese network [35] to extract the appearance feature vector from the objects being tracked, instead of the original implementation, which uses CNN. Siamese networks are known to work well in resource matching problems. This network was trained from the NVIDIA AI city Challenge data set [36].

C. Features extraction and deep learning regressor

Initially, we used other features besides the optical flow. We use the RGB images and the depth we extract from them. So, we did some tests varying these features (RGB, depth and optical flow) as input and also training our regressor from the three features extracted. We obtained the best result using the dense optical flow as an input. We extract the dense optical flow using FlowNet2 [13]. FlowNet treats speed estimation as a supervising problem where a neural network is trained with two sequentially adjacent input images, which have the ground truth of the optical flow as a supervision signal. In our work, we used the FlowNet2 architecture pre-trained on the synthetic Flying Chairs [37] data set. For each tracking obtained with DeepSORT within the measurement region, we calculate the average of the optical flow obtained and pass it as input to a deep learning based regressor. We retain the image dimension of the first tracking bounding box. The remaining tracking bounding boxes are enlarged, starting from the center, until reaching the dimension of the first tracking image. This is so to preserve the context of the scenario, its background, avoiding for the need of resizing the image when calculating the average at the end of the tracking, as well as possible losses.

Before reaching our regressor, we carried out some tests with ResNet [31] variations: ResNet-18, ResNet-34, and ResNet-50. We realized that as we increased the complexity of the network, our training goes to overfitting. Therefore, we chose to adapt a simple VGG-16 convolutional network [14] for our purpose. We use the Exponential Linear Unit (ELU) [38] as an activation function instead of the original RELU [39], in the convolutional layers of the network. ELU

TABLE I: Comparison of measurement results obtained by our method and the Luzivon et al. method [6]. First and second columns are the standard deviation and average error of the estimated speeds, respectively. "Lower", "Higher" and "Ideal" represent speed errors below, above and within acceptable limits, respectively.

Methods	Standard deviation	Average speed error	Lower	Ideal	Higher
Luzivon	1.36 km/h	-0.5 km/h	1.1%	96.0%	2.8%
Ours	2.64 km/h	-0.78 km/h	9.0%	85.4%	5.6%

is a function that tends to converge to zero quickly, while reaching high precise results. At the end of the convolutional layers, we added an adaptive average pooling before fully-connected (FC) layers. This allows our regressor to receive the optical flow average of vehicles tracked in different dimensions, without having to resize them, avoiding losses. We use two FC layers with 4,096 neurons, and a layer with a single neuron, which is the estimated vehicle speed. Our network is trained in MSE between network output and vehicle speeds targets for 2,000 epochs long. The Adam optimizer [40] is used for training. Figure 4 illustrates the training pipeline together with the network structure used.

IV. EXPERIMENTAL ANALYSIS

We use the Luzivon's data set [6] to validate our estimator. Luzivon et al. [6] estimate the speed of vehicles by tracking their license plates using the KLT Tracker. To find the speed in km/h, from pixel images, Luzivon uses perspective rectification from a measurement performed on the road. The author reached an average error of -0.5 km/h and standard deviation of 1.36 km/h. The maximum nominal error value found for speed, across the whole data set, is -4.68 km/h and $+6.00$ km/h. Our system uses a regressor that has been fully trained by gathering synthetic data. We managed to achieve a relative precision in our estimates, reaching an average error of -0.78 km/h, standard deviation of 2.64 km/h, with a maximum nominal errors of -29.81 km/h and $+22.27$ km/h. 96% of the speeds estimated by the Luzivon et al. system [6] are within an error range of -3 to $+2$ km/h. This error range is considered ideal by traffic regulators in several countries. Without any measurement on the road and with a synthetic data set, our proposed system reached 85.4% of vehicles in this range. Table I summarizes this comparative results.

Luzivon et al. [6] found difficulties in measuring the speed of motorcycles to generate the ground truth. 4.5% of the total number of vehicles in its data set are motorcycles. The ground truth measurement system used, inductive loop detector, was able to measure the speed of only 43% of all motorcycles in the data set. This is also a challenge in our work. We obtained an average error in the speed measurement of motorcycles of -8.36 km/h, with only 18.64% of the estimated speeds within the ideal error range. The rest of the measurements of the other categories of vehicles, truck, bus and car, obtained rates ranging from 81.6% to 88.2% of estimated speeds within the ideal error range. The lowest rate is for truck and the highest is for bus.

TABLE II: Speed measurements by our method for different types of detected vehicles: Car, bus, motorcycle and truck, in the Luvizon's data set. Standard deviation and average error of the estimated speeds are in the first and second columns. The last three columns show the estimated error percentage within the "Lower", "Ideal" and "Higher" ranges.

Vehicle type	Standard deviation	Average speed error	Lower	Ideal	Higher
Car	2.3 km/h	-0.64 km/h	7.7%	86.7%	5.6%
Bus	1.88 km/h	-1.77 km/h	11.8%	88.2 %	0 %
Motor-cycle	5.35 km/h	-8.36 km/h	78.8%	18.6 %	2.5%
Truck	-1.1 km/h	3.98 km/h	10.7%	81.6%	7.8%

Details of the estimates by vehicle category are shown in Table II.

We also tested the performance of our system from training with real data, applying k-fold cross-validation. k-fold cross-validation divides the data set into k portions and uses 1 portion as test data and the remaining k-1 as training data. We used three k-Fold cross-validation for k = 3, k = 5 and k = 10 for validation on the Luzivon's data set [6], with 50 training epochs. The results for k = 3 is an average error of -0.6 km/h, -0.53 km/h for k = 5, and 0.66 km/h for k = 10, values very closer to the error found in the Luzivon's proposed method [6] (0.5 km/h), indicating that our method is able to generalize also from real data.

V. CONCLUSIONS

This paper introduced a method for measuring vehicle speed through monocular cameras, built solely from synthetic data, and without making use of any calibration procedure. We achieved competitive results in comparison with state-of-the-art [6], which performs camera calibration and makes use of a real-world data set. Our approach can be generalized for any other scenario, since we can generate customized synthetic data sets, with other specific camera viewpoints. As future work, we intend to increase the number of synthetic data samples, including more scenarios and weather conditions, conduct training using samples from other dense optical flow extraction methods and train with other variations of deep learning. Yey the goal is to train samples by speed and type of vehicles in order to carrying out specific training for the types of vehicles whose estimation error is high, such as motorcycles. Finally, we intend to capture samples from real world scenarios and validate them using our method.

REFERENCES

[1] M. Mehrubeoglu and L. McLauchlan, "Determination of vehicle speed in traffic video," in *Real-Time Image and Video Processing 2009*, N. Kehtarnavaz and M. F. Carlsohn, Eds., vol. 7244, pp. 233 – 244, 2009. [Online]. Available: https://doi.org/10.1117/12.805932

[2] H. Weigel, H. Cramer, G. Wanielik, A. Polychronopoulos, and A. Saroldi, "Accurate road geometry estimation for a safe speed application," in *IEEE Intelligent Vehicles Symposium*, pp. 516–521, 2006.

[3] S. Javadi, M. Dahl, and M. I. Pettersson, "Vehicle speed measurement model for video-based systems," *Computers and Electrical Engineering*, vol. 76, pp. 238–248, 2019. [Online]. Available: https://doi.org/10.1016/j.compeleceng.2019.04.001

[4] H. Dong, M. Wen, and Z. Yang, "Vehicle speed estimation based on 3d convnets and non-local blocks," *Future Internet*, vol. 11, no. 6, p. 123, 2019. [Online]. Available: https://doi.org/10.3390/fi11060123

[5] E. A. Bernal, W. Wu, O. Bulan, and R. P. Loce, "Monocular vision-based vehicular speed estimation from compressed video streams," *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, no. Itsc, pp. 1155–1160, 2013.

[6] D. C. Luvizon, B. T. Nassu, and R. Minetto, "A Video-Based System for Vehicle Speed Measurement in Urban Roadways," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 6, pp. 1393–1404, 2017.

[7] S. Doğan, M. S. Temiz, and S. Külür, "Real time speed estimation of moving vehicles from side view images from an uncalibrated video camera," *Sensors*, vol. 10, no. 5, pp. 4805–4824, 2010.

[8] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2*, ser. IJCAI'81, p. 674–679, 1981.

[9] M. Dubská, J. Sochor, and A. Herout, "Automatic camera calibration for traffic understanding," *BMVC 2014 - Proceedings of the British Machine Vision Conference 2014*, 2014.

[10] V. K. Madasu and M. Hanmandlu, "Estimation of vehicle speed by motion tracking on image sequences," in *2010 IEEE Intelligent Vehicles Symposium*, pp. 185–190, June 2010.

[11] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *CoRR*, vol. abs/1506.01497, 2015. [Online]. Available: http://arxiv.org/abs/1506.01497

[12] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *2017 IEEE international conference on image processing (ICIP)*, pp. 3645–3649, 2017.

[13] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "Flownet 2.0: Evolution of optical flow estimation with deep networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul 2017. [Online]. Available: http://lmb.informatik.uni-freiburg.de//Publications/2017/IMKDB17

[14] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015. [Online]. Available: http://arxiv.org/abs/1409.1556

[15] J. Sochor, R. Juranek, J. Spanhel, L. Marsik, A. Siroky, A. Herout, and P. Zemcik, "Comprehensive Data Set for Automatic Single Camera Visual Speed Measurement," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–11, 2018.

[16] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick, "Mask R-CNN," *CoRR*, vol. abs/1703.06870, 2017. [Online]. Available: http://arxiv.org/abs/1703.06870

[17] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," *CoRR*, vol. abs/1506.02640, 2015. [Online]. Available: http://arxiv.org/abs/1506.02640

[18] T. Huang, "Traffic Speed Estimation from Surveillance Video Data Institute for Transportation , Iowa State University," *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 161–165, 2018.

[19] M. C. Chang, Y. Wei, N. Song, and S. Lyu, "Video analytics in smart transportation for the AIC'18 challenge," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, vol. 2018-June, pp. 61–68, 2018.

[20] P. Giannakeris, V. Kaltsa, K. Avgerinakis, A. Briassouli, S. Vrochidis, and I. Kompatsiaris, "Speed estimation and abnormality detection from surveillance cameras," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, vol. 2018-June, pp. 93–99, 2018.

[21] P. Najman and P. Zemčík, "Vehicle speed measurement using stereo camera pair," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–9, 2020.

[22] M. G. Moazzam, M. R. Haque, and M. S. Uddin, "Image-based vehicle speed estimation," *Journal of Computer and Communications*, vol. 7, no. 6, pp. 1–5, 2019.

[23] E. Bochinski, V. Eiselein, and T. Sikora, "High-speed tracking-by-detection without using image information," in *14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pp. 1–6, 2017.

[24] C. Tomasi and T. Kanade, "Detection and tracking of point," Technical Report CMU-CS-91-132, Carnegie, Mellon University, Tech. Rep., 1991.

[25] J. Sochor, R. Juránek, and A. Herout, "Traffic surveillance camera calibration by 3d model bounding box alignment for accurate vehicle speed measurement," *Computer Vision and Image Understanding*, vol. 161, p. 87–98, Aug 2017. [Online]. Available: http://dx.doi.org/10.1016/j.cviu.2017.05.015

[26] V. Kocur and M. Ftáčnik, "Detection of 3d bounding boxes of vehicles using perspective transformation for accurate speed measurement," *Machine Vision and Applications*, vol. 31, no. 7-8, Sep 2020. [Online]. Available: http://dx.doi.org/10.1007/s00138-020-01117-x

[27] Z. Tang, G. Wang, H. Xiao, A. Zheng, and J. N. Hwang, "Single-camera and inter-camera vehicle tracking and 3D speed estimation based on fusion of visual and semantic features," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, vol. 2018-June, pp. 108–115, 2018.

[28] D. Tran, J. Ray, Z. Shou, S. Chang, and M. Paluri, "Convnet architecture search for spatiotemporal feature learning," *CoRR*, vol. abs/1708.05038, 2017. [Online]. Available: http://arxiv.org/abs/1708.05038

[29] M. Kampelmühler, M. G. Müller, and C. Feichtenhofer, "Camera-based vehicle velocity estimation from monocular video," *CoRR*, vol. abs/1802.07094, 2018. [Online]. Available: http://arxiv.org/abs/1802.07094

[30] A. Dosovitskiy, G. Ros, F. Codevilla, A. M. López, and V. Koltun, "CARLA: an open urban driving simulator," *CoRR*, vol. abs/1711.03938, 2017. [Online]. Available: http://arxiv.org/abs/1711.03938

[31] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pp. 770–778, 2016. [Online]. Available: https://doi.org/10.1109/CVPR.2016.90

[32] T. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie, "Feature pyramid networks for object detection," *CoRR*, vol. abs/1612.03144, 2016. [Online]. Available: http://arxiv.org/abs/1612.03144

[33] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds., pp. 740–755, 2014.

[34] S. R. Maiya, "Deepsort: Deep learning to track custom objects in a video," Apr 2020. [Online]. Available: https://nanonets.com/blog/object-tracking-deepsort/

[35] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in *ICML deep learning workshop*, vol. 2, 2015.

[36] M. Naphade, S. Wang, D. C. Anastasiu, Z. Tang, M. Chang, X. Yang, L. Zheng, A. Sharma, R. Chellappa, and P. Chakraborty, "The 4th ai city challenge," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 2665–2674, 2020.

[37] A. Dosovitskiy, P. Fischer, E. Ilg, P. Häusser, C. Hazirbas, V. Golkov, P. v. d. Smagt, D. Cremers, and T. Brox, "Flownet: Learning optical flow with convolutional networks," in *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 2758–2766, 2015.

[38] D. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (elus)," in *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2016. [Online]. Available: http://arxiv.org/abs/1511.07289

[39] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ser. ICML'10, p. 807–814, 2010.

[40] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015. [Online]. Available: http://arxiv.org/abs/1412.6980