Driving Safe Speed Estimation Based on Outside Environment Vision Analysis

Alexey Kashevnik
SPC RAS
Saint Petersburg, Russia
alexey.kashevnik@iias.spb.su

Ammar Ali
ITMO University
Saint Petersburg, Russia
ammarali32@itmo.ru

***Abstract*—Using predefined speed limits on roads had huge positive safety effects on decreasing the vehicle accident rate. On the other hand using these static limits was the 20th-century solution. With the recent evolution of machine learning, driver assistant systems, and autonomous driving the necessity for dynamic speed limits are raised. In this paper we propose a novel method to analyze the scene and adjust the speed limits according to the environment's dynamic changes taking into account static constraints. Our proposed system is a recommendation system to estimate safe speed limits. It will consider the static predefined speed limits as well as other values like (the number of vehicles on the scene, the relative distance to the closest vehicle, the road width and curvature, the weather state, and the day/night (illumination) states.**

I. Introduction

Estimating the suitable safe speed is an essential feature for autonomous vehicles to make it closer to production [1] and [2]. The importance of choosing a suitable speed according to the changes in the dynamic environment is not only because it will decrease the probability of accidents but also protect the pedestrians. Integrating the such system with a motion prediction system could overcome multiple obstacles that stand against the evolution of the autonomous vehicle. The benefits of a safe speed estimation system could extend to the driver assistant and monitoring systems to signal the driver if he is above the limit according to the dynamic changes in the environment where the predefined static limit is not so helpful. We present a methodology to estimate the dynamic safe speed limit for the driver according to the outside environment changes. Even with the evolution of machine learning techniques in computer vision, driver assistant systems, and autonomous driving almost all countries still use a static speed limit for each road. The static limits are not the 21st-century solution. In this paper we present our proposed system that adaptively changes the speed limit according to the changes in the outside environment using only a monocular camera taking into account the following factors:

- number of vehicles on the scene;
- the static speed limit;
- road width;
- the relative distance to each vehicle;
- the road curvature;
- weather state;
- day illumination state.

Due to the lack of research on the main topic (safe speed estimation) the related work will concentrate on the subtasks used to construct our proposed system (vehicle detection, road segmentation, monocular depth estimation, weather, and illumination classification).

The structure of the paper is as follows. In Section II we present the related work on the previous factors and the recent research on speed estimation. We present our proposed safe speed estimation approach in Section III. In section IV we showed our results for some system components. The conclusion summarizes the paper.

II. Related Work

In this section we present the latest researches on multiple fields that are important for our proposed system including (3D object detection, monocular depth estimation, road segmentation, weather classification, and day illumination state classification.

A. 3D Object Detection

Authors of the paper [3] proposed a monocular vision-based approach for 3D vehicle localization and tracking. They used a multi-frame optimization based on the camera motion and track which made the results competitive with the LIDAR-based approaches.

In the paper [4] authors proposed a multi-fusion approach to get multi-view representations of LIDAR from the point clouds and images to design an end-to-end 3D object detection and trajectory prediction.

Another approach was proposed in the paper [5]. This monocular approach is based on a pseudo stereo generation with three virtual view generations (image-level, feature level, and feature clone). The authors used a disparity-wise dynamic convolution to filter the features from the image adaptively to generate the virtual image features and fix the depth errors.

In the paper [6] authors proposed a graph neural network for vehicles and pedestrian landmarks extractions and pose estimation. To assign training weights to different parts of a pose they used graph centrality measure. The approach is the current state of the art for car pose estimation on the ApolloCar3D dataset.

Authors of the paper [7], proposed a 3D detection and tracking neural network based on the CenterNet detector with points tracking approach. The approach allows to accelerate the 3D object detection which is also known as tracking by detection.

B. Monocular Depth Estimation

In this research, [8] authors proposed to use a simple encoder (EfficientNet) - decoder (UNet) architecture with (mini-ViT) head to generate a bin of values for each pixel. Then, using the centers of these bins to generate the depth map achieving the state of the art results.

On the other hand other researchers tried to solve this problem using geometric information [9]. They propose to use a generative adversarial network to generate synthesis data for other views as well as to reduce the difference between the generated data and the reality they mapped the corresponding domain-specific information related to the primary task into shared information.

In the paper [10] authors used a multi-frame approach with geometric constraints to get the depth values using a time recurrence-based model and a spatial reprojection layer to maintain the Spatio-temporal consistency between the levels.

The authors of the paper [22] proposed a new loss function (virtual normal loss) for a simple encoder-decoder architecture. The idea beyond their loss function is to enforce the geometric constraints by aligning the spatial relation between the predicted depth maps and the ground truth.

C. Road Segmentation

Road segmentation is a typical segmentation task and a really important one for self-driving and driver assistant systems. In the paper [12] authors targeted embedded devices and their main research was to compare three well-known neural networks for segmentation (PSPNet, FCN, and SegNet) taking into account the trade-off between the running time and the intersection over union (IoU metric).

Other researchers suggested a new post-processing probability layer [13]. The main approach is based on building a traditional encoder-decoder architecture for road segmentation and a 2D array using the probability layer over the training set, then combining both using a neural network. It boosted the results in comparison to the traditional solutions on Kitti Dataset.

Another research that tackles the road segmentation problem on challenging domains [14] without depending on a huge training set. Authors proposed visual place recognition to find similar scenes and fuse predefined masks to the prediction using a Bayesian approach to achieving a new state-of-the-art solution in challenging datasets (snow, night conditions).

D. Weather and light Classification

In the paper [15] authors proposed architecture for weather classification based on images processing. The dataset they used presents 11 different weather states. Authors called their new architecture MeteCNN which is completely based on the inception architecture with a modified neck by using a skip connection between the base architecture and the head upscaling layer obtaining a 92% accuracy on the WEAPD testing set.

On the other hand authors of the paper [16] build a multi-class solution not only to classify the weather state but also for the light and road states. They contributed with an open-source dataset that is designed to tackle autonomous driving.

E. Safe Speed Estimation

Even though this research is new there are some similar researches that tackled this problem. In the paper [17] authors proposed a system to estimate the safe speed on Urban roads taking into account the traffic accident rate in road sections, the degree of roadside development, and the geometric characteristics of road sections with other optional variables to estimate the safe speed limits.

Another study [18] shows a problem with the current static speed limits by analyzing the relation between 85th percentile of the operating speed distributions (V_{85}) and road characteristics which showed that the current static speed limits should be adjusted.

Our research will not only based on static information to adjust the speed limits but also on the dynamic changes in the environment including (the number of vehicles, distance (depth maps), pedestrians) and static features like (road curvature, road width, and etc).

III. Safe Speed Estimation

Our proposed system takes into consideration multiple factors to estimate a vehicle safe. The factors include static ones (predefined speed limit, road width, road curvature, and etc.) as well as dynamic (number of vehicles, weather state, day time and etc.). We propose to define the safe speed estimation by the following formula (see Fig. 1):

$$safe\_speed = \frac{SL * D * RW * cos(\theta)}{MD * max(log(NV + 1), 0.4)}, \qquad (1)$$

where SL is the predefined speed limit, D is the distance to the closest vehicle on the scene, RW is the road width, θ is the angle of the road and NV is the number of vehicles. Also, we have two more factors that affect to the safe speed estimation in a constant way: (1) W the weather state and (2) I the daytime. MD is the max-depth value and it is a constant value depends on the depth estimation model in our system it will be 10. 0.4 is a tuning parameter to adjust the safety factor. The formula is described in details in the Section III-E. The proposed approach requires to get these factors as inputs to estimate the safe speed. We propose to use multiple neural networks to manage this data.

A. Vehicles Around

In the subsection we describe developed method for monocular debt estimation that allow us to calculate a number of vehicles around as well as calculate the distance to vehicles in front of our vehicle.

For monocular depth estimation we designed a simple encoder (EfficientNetB0) decoder (Nested Unet) architecture

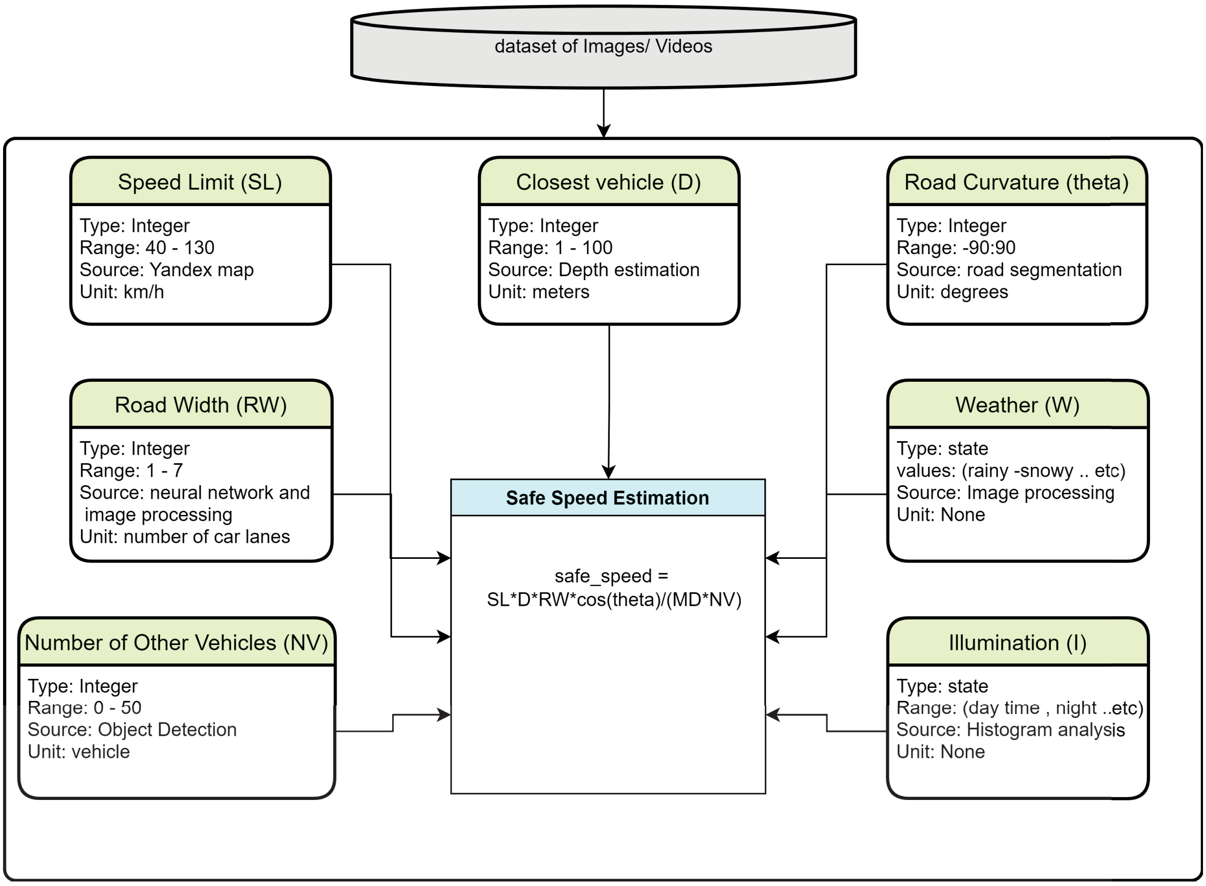

Fig. 1. Proposed Approach to Safe Speed Limit Estimation

shown in(2). We trained the model over our data using a pseudo labeling techniques to auto annotate based on the last 4 SOTA models for monocular depth estimation (BTS [19], LapDept [20], DPT Hybrid [21], and VNL [22]).

To enforce the model to a better convergence we use a hybrid loss function consisting of.

1) For enforcing the depth values to be correct on the edges of the images we used Sobel Loss.
2) Using the logarithm $L1$ Loss between the predicted depth map and the ground truth to enforce the smoothing on the predicted depth values.
3) Cosine similarity between the output and the ground truth.
4) Use virtual normal loss to enforce the geometric constraints by constructing a point cloud from the depth maps and use triangulation for correction that will assure high-order geometric supervision in the 3D space.

The depth estimation model returns depth maps for the full input image but we need to compute the distance to other vehicles in the scene. Therefore, we used the YOLOv5 medium trained on our dataset [23] to detect the vehicles and then for each vehicle on the scene, we take the closest point as the distance between it and the camera.

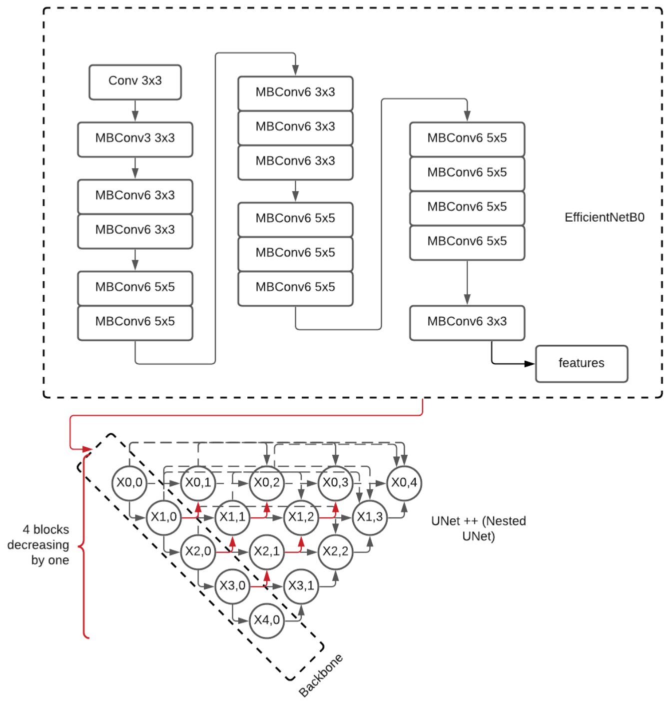

Fig. 2. Proposed Light-Weight Efficient Monocular Depth Estimation Model

In Fig. 3 for each input image we generate the corresponding depth map using our depth estimator. On the other hand using our detector we get the bounding boxes for each vehicle on the scene cropping the area of these bounding boxes from the depth map and taking the min value will return the information about the closest vehicle on the scene. Moreover, using the vehicle detector we get information about the number of vehicles on the scene.

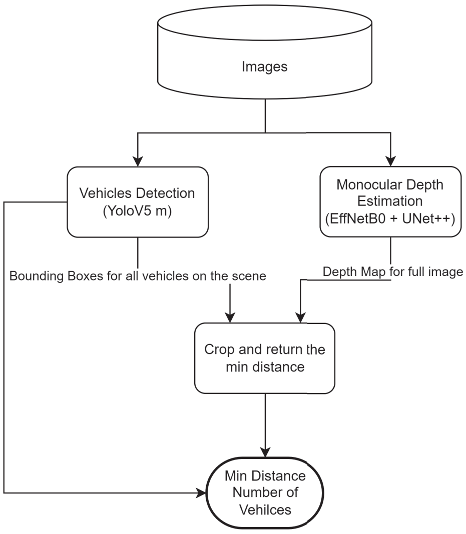

Fig. 3. An Algorithm for getting the Closest Vehicle and the Number of Vehicles on the Scene

B. Road Curvature and Width

We use the same architecture proposed in [24] to detect road curvature and width. It is EfficientNetB3 with bidirectional FPN and segmentation head only. We implement training on Berkeley DeepDrive Dataset [25] using the same hyper-parameters and training details. We used our dataset only for validation. So that the input is a simple RGB image and the output is a segmentation mask for the road with border lines for lanes detection.

From the predictions we consider the number of the detected lanes as the road width because it indicates the number of vehicles that could fit on the road horizontally.

To calculate the road curvature we take the nearest middle point of the lane that the vehicle is driving into and the farthest middle point and calculate the angle between the line that connects these points with the vertical axis and consider this angle as an approximation to the road curvature.

In Fig. 4 the white points represent the centers of the lane at the closest and farthest levels. The green dot line represents the vertical axis and the red dots line connects the centers, the θ is the angle between the green and red dot lines considered as the road curvature.

To calculate the angle between these two lines we need to find the slope m of the red line and subtract 90 degrees.

$$m = \frac{y_2 - y_1}{x_2 - x_1}, \tag{2}$$

where (x_1, y_1) and (x_2, y_2) are any two points from the red line. On the other hand, it is shown that the number of lanes is two in this image so we consider the road width is two.

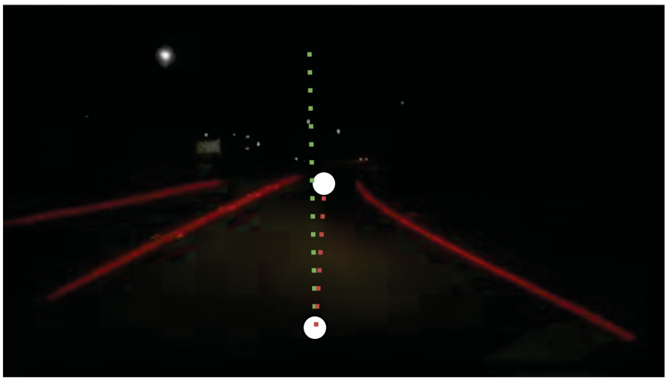

Fig. 4. Road Curvature and Width Estimation

C. Weather and Day/Night Classification

We used a pre-trained model that is trained on a cleaned dataset from ACDC and Berkeley DeepDrive datasets [26]. The strategy used to build the system is as follows.

1) For day/night classification a basic random forest model trained on average RGB and HSV values to classify the daytime.
2) For rain/snow/clear state a simple convolution neural network is used.
3) For foggy weather classification another simple convolution neural network is trained.

Our system will use this information to modify the estimated safe speed (see Algorithm 1).

Algorithm 1 Weather and day/night effect on the safe speed

0: If The day time is night then

$$safe\_speed = 95\% * safe\_speed \tag{3}$$

0: if the weather is rainy:

$$safe\_speed = 90\% * safe\_speed \tag{4}$$

0: if the weather is snowy:

$$safe\_speed = 80\% * safe\_speed \tag{5}$$

0: if the weather is foggy:

$$safe\_speed = 80\% * safe\_speed \tag{6}$$

D. Speed Limit

For each video we have captured during the last few years in our dataset we have GPS information (longitude, latitude). After extracting this information from our database use Yandex API to get information about the speed limit at each point of the path.

E. Safe Speed Estimation Formula

As mentioned above our system will estimate the safe speed using the formula 1 with other constant regulations according to the weather and the day time.

$$safe\_speed = \frac{SL * D * RW * cos(\theta)}{10 * max(log(NV + 1), 0.4)}.W.I, \quad (7)$$

where W is (1, 0.9, 0.8, 0.8) according to these weather states (clear, rainy, snowy, foggy) in order. I is (1, 0.9) if the daytime is (day, night). Notice that divided by 10 is for the distance value $\frac{D}{10}$ to normalize the depth values between 0 and 1.

It is important to notice that the proposed safe speed estimation could be higher than the speed limit itself. In the case for example we have a 3 width road (3 lanes) and there were no other vehicles, the road is straight so $cos(\theta) = 1$ and $\frac{D}{10} = 1$, as well the weather is clear in a day time then:

$$safe\_speed = 3 * SL$$

, which is not so good because even if all the conditions are perfect the government defined speed limit could not be wrong that match therefore we made an upper bound of the estimated speed to be equal to the predefined static speed limit as follows:

$$safe\_speed = Min(safe\_speed, speed\_limit) \quad (8)$$

IV. Results

In this section we present the output of each component of our system and the output of the full recommendation system for some random images from our dataset.

A. Depth estimation and vehicle detection

We trained the models on our dataset the ground truth was generated using psuedo labeling. Fig. 5 shows an example of our detector and depth estimator.

Table I presents a comparison between our depth estimation model and Adabins model (state of the art on NYU dataset).

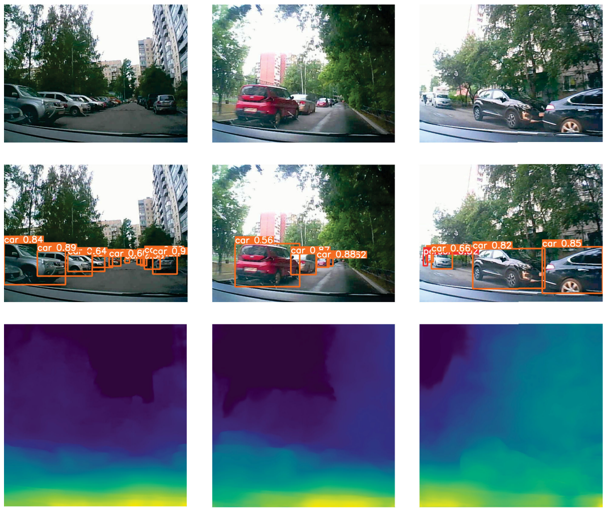

Fig. 5. First row is the original image, second row is the output of Yolov5m6 (vehicle detector), third row is the output of the depth estimator (EffB0 with Nested UNet)

Our Yolov5 medium detector achieves 82.3% mAP@.5:.95 for car class and an average mAP for all classes of 77.6 (pedestrians, cars, trucks and motorcycles).

B. Road width and curvature

The road width is defined as the number of lanes on the road regardless if these routes are occupied by other vehicles or not as shown in Fig.6 (road width of both is three).

In the right down image the segmentation mask is not presented on the left lane because it is occupied by another vehicle.

In Fig. 7 we show how to approximately estimate the road angle which equals 72 degrees (calculated automatically III-B) on this image. For future enhancement it is better to calculate the angle between the line which connects the centers of the nearest and farthest points of the mask with the motion direction axis instead of the vertical axis.

C. Weather and Day/Night Classification

The day/night classification accuracy on our data is 97% using the pre-trained random forest model. The weather classification is 89% using the pre-trained CNN provided on the repo.

D. Safe speed estimation results

Let us take image shown in Fig.7. For example a number of detected vehicles is 3, speed limit is 40 km/h, weather is clear, and day time.

TABLE I
EVALUATION METRICS FOR EACH METHOD OVER OUR DATASET

Method	Encoder	Decoder	RMSE	ABS_REL	log_10	Delta1	Delta2	Delta3
AdaBins	EfficientNet-b5	UNet + miniViT	0.5136	0.1176	0.0477	0.8638	0.9658	0.9884
Ours (large)	EfficientNet-b0	UNet++	0.5960	0.1173	0.04885	0.8559	0.9597	0.9841

Fig. 6. First row is the original images, second row is the output of the road segmentation head and the borders (it is clear that both images include three lanes)

The road width is 1 and the road angle is $\theta = 72$. The relative distance to the closes vehicle is 8.2.

$$safe\_speed = \frac{20 * 7 * 1 * cos(72)}{10 * max(log_{10}(4), 0.4)} * 1 * 1 \quad (9)$$

$$safe\_speed = 7km/h \quad (10)$$

As you can see in the image there is a cross to change direction with multiple vehicles on the corner with a road with one lane (hardly fits two cars). Therefore, it is safe to drive at a lower speed than the specified driving limit. Our system proposed that the safe run to turn is (7 Km/h) instead of the static speed limit (20 Km/h). Let us take the left image shown on Fig.6

$$safe\_speed = \frac{60 * 8 * 1 * cos(0)}{10 * max(log_{10}(2), 0.4)} * 1 * 1 \quad (11)$$

$$safe\_speed = min(120, 60) = 60km/h \quad (12)$$

As shown in the figure the road is almost empty and the distance to the closest vehicle is good enough. As well the road is in one direction with three lanes. Therefore our system proposed that the safe speed is the max allowed speed which is equal to the predefined static limit ($60Km/h$). Moreover the left image shown on Fig. 5.

$$safe\_speed = \frac{40 * 9 * 1 * cos(0)}{10 * max(log_{10}(10), 0.4)} * 1 * 1 \quad (13)$$

$$safe\_speed = 36km/h \quad (14)$$

Fig. 7. Upper image is the original (input image), the bottom Image is the output of the segmentation head of HybridNets

In the image, we can see that the road is empty. On the other hand, it has one lane with multiple parking vehicles. So, our system proposed that the safe speed is $36Km/h$ instead of $40Km/h$.

V. Conclusion

We proposed an approach to estimate the safe speed according to the dynamic situation on the road. The proposed system will take into account the speed limit specified by the government and adjust its value according to the number of vehicles, the distance to the closest vehicle, the road curvature and width, the weather state, and the time of the day on a particular scene. Our system recommends a max speed equal to the specified limit by the government in perfect conditions. Otherwise, it suggests being slower and safer. It shows promising results on a small amount of data with manual testing and evaluation. Unfortunately, the lack of research tackling this problem makes it hard to evaluate properly. Proposed approach is based on SOTA models to estimate the safe speed. Even though it still needs to be expanded and to take more features into account including pedestrians and motion prediction for the moving objects on the scene.

ACKNOWLEDGMENT

This work was supported by the Russian Science Foundation (project 18-71-10065). We also thank to Yandex LLC company that provide us access to speed limit information in St. Petersburg city area.

REFERENCES

[1] S. Li, M. Huang, M. Guo, and M. Yu, ”Evaluation Model of Autonomous Vehicles’ Speed Suitability Based on Overtaking Frequency”, Sensors 2021, 21, 371. https://doi.org/10.3390/s21020371

[2] J. Daniel , W. Stewart , S. Mao. (2021). ”What is the appropriate speed for an autonomous vehicle? Designing a Pedestrian Aware Contextual Speed Controller.” https://doi.org/10.48550/arXiv.2104.06147

[3] H. Zhang, H. Ji, A. Zheng, J. -N. Hwang and R. -H. Hwang, ”Monocular 3D Localization of Vehicles in Road Scenes,” 2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), 2021, pp. 2855-2864, doi: 10.1109/ICCVW54120.2021.00320.

[4] S. Fadadu et al., ”Multi-View Fusion of Sensor Data for Improved Perception and Prediction in Autonomous Driving,” 2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), 2022, pp. 3292-3300, doi: 10.1109/WACV51458.2022.00335.

[5] C. Yi-Nan , D. Hang , D. Yong. (2022). Pseudo-Stereo for Monocular 3D Object Detection in Autonomous Driving. (Accepted to CVPR 2022) arXiv:2203.02112

[6] D. Zauss, S. Kreiss and A. Alahi, ”Keypoint Communities,” 2021 IEEE/CVF International Conference on Computer Vision (ICCV), 2021, pp. 11037-11046, doi: 10.1109/ICCV48922.2021.01087

[7] X. Zhou, V. Koltun, and P. uhl. Tracking objects as points. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, Computer Vision – ECCV 2020, pages 474–490, Cham, 2020. Springer International Publishing.

[8] S. Farooq Bhat, I. Alhashim and P. Wonka, ”AdaBins: Depth Estimation Using Adaptive Bins,” 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 4008-4017, doi: 10.1109/CVPR46437.2021.00400.

[9] P. N. V. R. Koutilya, H. Zhou and D. Jacobs, ”SharinGAN: Combining Synthetic and Real Data for Unsupervised Geometry Estimation,” 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 13971-13980, doi: 10.1109/CVPR42600.2020.01399.

[15] Xiao, Haixia et al. “Classification of Weather Phenomenon From Images by Using Deep Convolutional Neural Network.” Earth and Space Science 8 (2021)

[10] Fonder, M., Ernst, D., Van Droogenbroeck, M. (2021). M4Depth: A motion-based approach for monocular depth estimation on video sequences. ArXiv, abs/2105.09847.

[11] W. Yin, Y. Liu, C. Shen and Y. Yan, ”Enforcing Geometric Constraints of Virtual Normal for Depth Prediction,” 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 5683-5692, doi: 10.1109/ICCV.2019.00578.

[12] T. Pham, ”Semantic Road Segmentation using Deep Learning,” 2020 Applying New Technology in Green Buildings (ATiGB), 2021, pp. 45-48, doi: 10.1109/ATiGB50996.2021.9423307.

[13] C. Weibin, ”Road Segmentation based on Deep Learning with Post-Processing Probability Layer”, IOP Conference Series: Materials Science and Engineering, 2020,DOI 10.1088/1757-899x/719/1/012076

[14] C. Malone, S. Garg, M. Xu, T. Peynot and M. Milford, ”Improving Road Segmentation in Challenging Domains Using Similar Place Priors,” in IEEE Robotics and Automation Letters, vol. 7, no. 2, pp. 3555-3562, April 2022, doi: 10.1109/LRA.2022.3146894.

[16] M. M. Dhananjaya, V. R. Kumar and S. Yogamani, ”Weather and Light Level Classification for Autonomous Driving: Dataset, Baseline and Active Learning,” 2021 IEEE International Intelligent Transportation Systems Conference (ITSC), 2021, pp. 2816-2821, doi: 10.1109/ITSC48978.2021.9564689.

[17] Kim, H.; Jung, D. Estimation of Optimal Speed Limits for Urban Roads Using Traffic Information Big Data. Appl. Sci. 2021, 11, 5710. https://doi.org/10.3390/app11125710

[18] V. Martinelli, R. Ventura, M. Bonera, B. Barabino, G. Maternini, Estimating operating speed for county road segments – Evidence from Italy, International Journal of Transportation Science and Technology, 2022,ISSN 2046-0430,https://doi.org/10.1016/j.ijtst.2022.05.007.

[19] Lee, J. Han et al. “From Big to Small: Multi-Scale Local Planar Guidance for Monocular Depth Estimation.” ArXiv abs/1907.10326 (2019): n. pag.

[20] M. Song, S. Lim and W. Kim, ”Monocular Depth Estimation Using Laplacian Pyramid-Based Depth Residuals,” in IEEE Transactions on Circuits and Systems for Video Technology, vol. 31, no. 11, pp. 4381-4393, Nov. 2021, doi: 10.1109/TCSVT.2021.3049869.

[21] R. Ranftl, A. Bochkovskiy and V. Koltun, ”Vision Transformers for Dense Prediction,” 2021 IEEE/CVF International Conference on Computer Vision (ICCV), 2021, pp. 12159-12168, doi: 10.1109/ICCV48922.2021.01196.

[22] W. Yin, Y. Liu, C. Shen and Y. Yan, ”Enforcing Geometric Constraints of Virtual Normal for Depth Prediction,” 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 5683-5692, doi: 10.1109/ICCV.2019.00578.

[23] Othman W., Kashevnik A., Ali A., Shilov N. DriverMVT: In-Cabin Dataset for Driver Monitoring Including Video and Vehicle Telemetry Information. Data, MDPI AG, Basel, Switzerland. 2022. Vol. 7(5). P. 1–13.

[24] Vu, Dat et al. “HybridNets: End-to-End Perception Network.” ArXiv abs/2203.09035 (2022): n. pag.

[25] https://bair.berkeley.edu/blog/2018/05/30/bdd/

[26] https://github.com/AlexandreSajus/Weather-Classification