Journal of Intelligent & Robotic Systems (2022) 105: 4
https://doi.org/10.1007/s10846-022-01606-3

REGULAR PAPER

ULODNet: A Unified Lane and Obstacle Detection Network Towards Drivable Area Understanding in Autonomous Navigation

Zhanpeng Zhang[1] · Jiahu Qin[1,2] · Shuai Wang[1] · Yu Kang[1,2,3] · Qingchen Liu[1]

Received: 20 August 2021 / Accepted: 24 February 2022 / Published online: 20 April 2022

Abstract
Drivable area understanding is an essential problem in the fields of robot autonomous navigation. Mobile robots or other autonomous vehicles need to perceive their surrounding environments such as obstacles, lanes and freespace to ensure safety. Many recent works have made great achievements benefiting from the breakthrough of deep learning. However, those methods resolve the challenge in a separated way which cause repeated utilization of resources in some occasions. Thus, we present a unified lane and obstacle detection network, ULODNet, which can detect the lanes and obstacles in a joint manner and further frame the drivable areas for mobile robots or other autonomous vehicles. To better coordinate the training of ULODNet, we also create a new dataset, CULane-ULOD Dataset, based on the widely used CULane Dataset. The new dataset contains both the lane labels and obstacle labels which the original dataset do not have. At last, to construct an integrated autonomous driving scheme, an area intersection paradigm is introduced to generate the driving commands by calculating the obstacle area proportion in the drivable regions. Moreover, the well-designed comparison experiments verify the efficiency and effectiveness of the new algorithm.

Keywords Autonomous navigation · Computer vision · Environment perception · Mobile robot

1 Introduction

Autonomous navigation, which plays a crucial role in the research fields of driverless vehicles, attracts more and more attention for the huge hidden potentials in industrial and civil applications. In recent years, thanks to the emergence of large-scale datasets [1, 2] and some well-performed convolution neural networks [3, 4], many autonomous navigation algorithms are proposed to help the autonomous vehicles drive more soundly. Generally speaking, autonomous navigation algorithm comprises perception module, control module, decision module, etc.. In this paper, we mainly focus on the discussion on perception module and propose a unified lane and obstacle detection network, ULODNet, for environment perception. It is noteworthy that we also design a tiny control module following ULODNet in addition to the perception module, which can provide the vehicles with available driving guidance suggestions. The overall architecture of the designed autonomous navigation scheme is illustrated in Fig. 1.

The perception module of autonomous navigation processes the driving environment information captured by the camera mounted in mobile vehicles, it can be divided into several sub-tasks such as obstacle detection, lane detection, freespace detection and so on. Most contemporary works [1, 5, 6] establish a single-functional network only concentrating on a separated sub-problem. However, in the real-world environment, integrating those single-functional

✉ Jiahu Qin
jhqin@ustc.edu.cn

Shuai Wang
wsustcid@mail.ustc.edu.cn

Yu Kang
kangduyu@ustc.edu.cn

Qingchen Liu
liuqingchen1989@gmail.com

Zhanpeng Zhang
zzpzkd@mail.ustc.edu.cn

[1] Department of Automation, University of Science and Technology of China, Hefei, 230027, China

[2] Institute of Artificial Intelligence, Hefei Comprehensive National Science Center, Hefei, 230088, China

[3] Institute of Advanced Technology, University of Science and Technology of China, Hefei, 230001, China

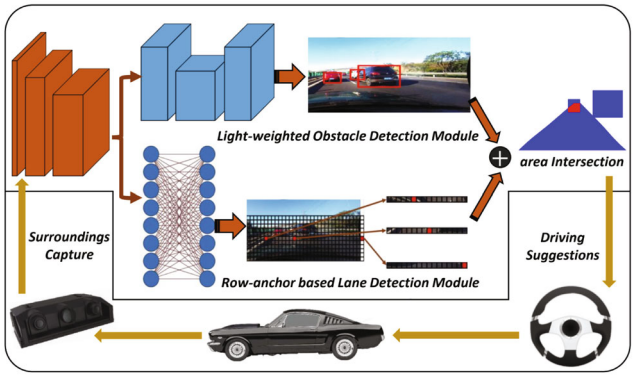

Fig. 1 Our designed autonomous navigation scheme: the camera captures surrounding images, which will be digested by perception module to get the lane and obstacle information, and be post-processed by decision module to output the driving commands

methods into a whole perception network consumes wasted resources because different sub-tasks have few interactions with each other. There are also some existing works [7] which can detect the drivable areas, lane lines and traffic objects jointly. But they adopt the semantic segmentation method to predict the specific classification of each pixel. This kind of method inevitably causes large computation resource consumption and affects the performance of the network. Thus, in this paper, we introduce a unified lane and obstacle detection network, ULODNet, to overcome the aforementioned shortcomings. In our work, we focus on the balance between speed and accuracy in both lane and obstacle detection branch. We also conduct some related experiments to show the efficiency and effectiveness of our proposed network.

In the fields of lane detection, Some researches [1, 5] typically adopt semantic segmentation paradigms to predict the lane pixels from input images. However, these methods cost significant inference time and their success rates heavily rely on the segmentation labels. Some other researches [6] relieve the drawbacks by treating the lane detection as a key point multiple dimensional classification problem, but they lose some global features and degrade the detection accuracy. Thus, we provide a better trade-off choice between segmentation methods and classification methods in our lane detection branch. Instead of predicting the class of every pixels of a whole image, we select some crucial anchor pixel clusters predictions to draw the lanes. This kind of method can save computation resources and shorten the inference time. Besides, considering that more global information is vital in the lane prediction problem, we adopt the spatial convolution modules in our designed network. Diversified convolution directions are applied which is different from the traditional convolution layers.

As for the obstacle detection branch, there are many existing object detection works that handle the task soundly. [8, 9] use anchor-based region proposal mechanism. They first apply pre-defined anchors to every pixel and classify some potential proposals, then a regression network is introduced to predict the offsets of bounding boxes and output the final results. [4, 10–12] merge the two detection phases into a whole architecture, the input images are divided to several grids and the object coordinates are extracted in the corresponding grid. Specifically, we redesign the network architecture based on the main idea of [11] in our obstacle detection branch. Feature maps from different blocks of backbone have different receptive fields, thus we utilize an up-sampling layer to combine them to output more accurate obstacle coordinates. Besides, we also change the number of obstacle classes to better improve the performance in the new CULane-ULOD Dataset.

To better illustrate the effectiveness of the proposed ULODNet, a new dataset named CULane-ULOD Dataset based on CULane Dataset [1] is created after failing to find an existing dataset. Obstacle labels are added to complement the original dataset. To the best of our knowledge, this is the first dataset which contains both the lane labels and obstacle labels. We pick common objects in driving environment such as car, truck and pedestrian manually and mark them as an overall category ***avoidance***. Finally, for the completeness of an autonomous driving scheme, an area intersection algorithm is introduced acting as the extensive research part of ULODNet for the purposes of estimating the appropriate driving instructions. The driving suggestions are judged by calculating the proportion of detected obstacle areas in the detected drivable area regions of two adjacent lanes. The transfer experiments to TuSimple Dataset [13] verify the robustness of the proposed algorithm.

In general, the contributions of this paper are concluded as follows:

- A unified lane and obstacle detection network, ULODNet, is proposed to process the information in autonomous driving.
- Two sub-branches of ULODNet, lane detection branch and obstacle detection branch, are deeply studied. In lane detection branch, there is a 1.2% accuracy increase of our designed model compared to the baseline; In obstacle detection branch, the effective network that we design can reach SOTA accuracy in the novel CULane-ULOD Dataset.
- A new CULane-ULOD Dataset transformed from CULane Dataset is created to better train our ULODNet. It includes both the lane marking labels and obstacle message labels. We also measure some basic information of the new dataset for the convenience of the utilization in other researches.
- An extensive post-process algorithm is proposed to help issue the driving commands based on the obstacle

area proportion in drivable regions. Thus, an integrated autonomous driving scheme is put forward, consisting of ULODNet and the post-process algorithm.

The remainder of this paper is organized as follows: Section 2 looks back on the relevant works in recent years, Section 3 demonstrates the architecture details of ULODNet, Section 4 illustrates the dataset used in this paper and the experiments based on the dataset, Section 5 discusses the post-process area intersection algorithm of ULODNet and Section 6 concludes the paper.

2 Related Work

2.1 Object Detection

Overall, the object detection algorithms [14] have two main routes. One is the anchor-based algorithm route, since RCNN [8] introduces regional convolution neural network into object detection to replace the hand-crafted features, the following researches develop many effective tricks, Faster RCNN [9] designs the RPN network to extract a higher-qualified ROI (Region of Interest) within a limited time, which can improve the processing time of the network, FPN [15] focuses on improving the ability to detect small objects, a hierarchical network is introduced to output more feature maps with divergent receptive fields so that the network can deal with different sizes of objects. And Mask RCNN [16] extends the regional convolution neural networks to semantic segmentation field, which utilizes the architecture in FCN [17] to predict the semantic information for every pixel of the images. However, [4, 18] is not satisfied with the inference time and it merges the proposal extraction procedure and the refinement procedure into an integrated network, and the following researches [10–12, 19, 20] develop many more advanced functions based on this work. Some of the detection methods are already applied in the robot-related researches [21, 22].

The other one is the anchor-free algorithm route, which is first introduced by CornerNet [23]. They convert the task of object detection into paired key points (the upper-left and lower-right corner points) detection, afterwards, the embedding branch predicts the location information for all pixels which can be compared to match the detected corner points for a specific object. Although some other methods like [24–27] propose many advanced detection tricks based on [23], the anchor-free method is still not mature enough for super-fast object detection problems compared to anchor-based algorithms.

Thus, for the specific application scenarios of ULODNet, the YOLO networks [11] are chosen as the baseline of obstacle detection network considering the outstanding processing time compared to Faster RCNN [9] and the remarkable accuracy advance compared to anchor-free algorithms [23, 25, 26].

2.2 Lane Detection

As for road lanes, there are about three main forms of expression.

(i) Polynomial expression: Some methods perceive the lane detection problem as a polynomial regression task, where the lanes can be represented by polynomials, [28–30] summarize conventional specialized hand-craft feature detectors to extract the road lines across the image and further group them into the polynomials. Some CNN-based methods such as PolyLaneNet [31] also attempt to output polynomial parameters straight from the neural networks, but the polynomial expression still struggles with the high bias towards straight lanes.

(ii) Segmentation expression: In 2018, LaneNet [5] considers instance segmentation expression [17] to replace the pure polynomial expression which picks the lane pixels directly from the background pixels, since segmentation methods depend significantly on end-to-end neural networks, the accuracy of lane detection has made enormous progress. [5] separates lane detection into two sub-branches, one is segmentation branch utilizing binary segmentation network to distinguish whether a specific pixel is a lane pixel or not, the other is embedding branch which helps disentangle the detected lane pixels belong to which lane, a tiny H-Net is also proposed to help fit the curve for each lane. At the same time, [1] proposes SCNN modules that utilize a specifically designed scheme for long thin structures to replace the conventional layer-by-layer convolution expecting to gain more global features of the image. And [32] focuses more on the convolution stride, they propose RESA (REcurrent Feature-Shift Aggregator) network with different convolution directions (horizontal and vertical) and different strides to gather global features.

(iii) Classification expression: Nevertheless, all the above-mentioned networks require large amounts of memory resources and the inference time is relatively slow, which hinders the applicability in some real-time required cases. Fortunately, the latest work UFLD [6] introduces the row-anchor based classification method, which is based on a grid level division, to represent the road lanes. They detect the most probable cell which contains lane markings for each row rather than detect all lane pixels, this kind of expression can accelerate the inference process significantly

without sacrificing accuracy. For this reason, the lane detection branch of ULODNet selects UFLD [6] as the baseline to help reduce the inference time.

3 Methodology

The proposed ULODNet network architecture is shown in Fig. 2. As it demonstrates, the network can be conceptually separated into two complementary branches, the one is obstacle detection network located at the top of the figure to help the driverless vehicles dodge the obstacles, the other one is the lane detection network at the bottom of the figure, which is designed to recognize the road lanes.

3.1 Obstacle Detection Branch

In this sub-section, the obstacle detection sub-network will be illustrated. Object detection related research has attracted more and more attention in recent years and many up-to-date tricks have proved to be effective. To make full use of the progress in this field, we design the obstacle detection network based on the existing mechanisms such as [4, 10, 11].

Generally, obstacle detection branch can be formulated as a regression task. The camera mounted in the autonomous platform provides RGB image $\Phi \in R^{3\times H\times W}$ to the branch. The bounding box coordinates (4 channels for x, y, w, h) and existence probabilities (1 channel for probability) are expected to regressed straight from the input image Φ.

Specially, as depicted in Fig. 3, the extraction network of the branch, referred as the backbone, is composed of 5 blocks to generate feature maps in different dimensions, where we can get features Φ_3, Φ_4 and Φ_5 from the input image Φ. In the following, feature map Φ_5 is processed by a ConvSet block, designed as Conv(3×3)-Conv(1×1)-Conv(3×3)-Conv(1×1)-Conv(3×3)-Conv(1×1), and a nearest upsampling layer to make sure the same size with Φ_4, then the new feature map of concatenated Φ_4 and Φ_5 do the same ConvSet process, upsampling process and concatenating process with Φ_3 to generate the final feature map. Then the detection decoding network outputs the regression predictions. At last, we calculate the offsets of the fixed anchors as the final networks output rather than the true coordinates considering the latter are more position-variant. Besides, we apply Non-maximum Suppression algorithm to remove the redundant predictions for the same obstacle and the loss function is formulated following [11].

Discussion: The principle of the network architecture designing is motivated by an interesting conjecture which has been verified in the following experiment: FPN Network [15] is originally designed for a more accurate detection task for the object appeared in the image which could be extremely large or extremely small, but the obstacles in driving environment have the regular sizes in most cases. For this reason, we choose feature map Φ_3 from the middle block of the backbone as the main basis of prediction since it has the appropriate receptive fields. And the ablation experiment details will be discussed in Section 4.3.

3.2 Lane Detection Branch

For the lane detection sub-network, we transform the lane detection as a row-anchor based classification problem, avoiding the huge computation resource consumption in the methodology [1, 5] which applies the semantic segmentation paradigms.

In contrary to obstacle detection branch, the lane detection branch makes the best of row anchors. An anchor is represented by the coordinates (x_i, y_i), thus we define X collections and Y collections to be the anchor coordinate collections. First, we pick an equally spaced y-coordinates collection $Y = \{y_i\}_{i=1}^{N}$, where N is the total number of the anchor and y_i represents the coordinate of a row. Since Y is fixed, the x-coordinates $X = \{x_j\}_{y=y_i}$, where each x_j corresponded to the respective $y_i \in Y$, become the only key value to recognize the lane and the lane detection problem can be formulated as the multi-dimension classification problem on the row anchors. Thus, for a specific lane, a classification module is trained to classify the X anchor pixels whether it belongs to the lane.

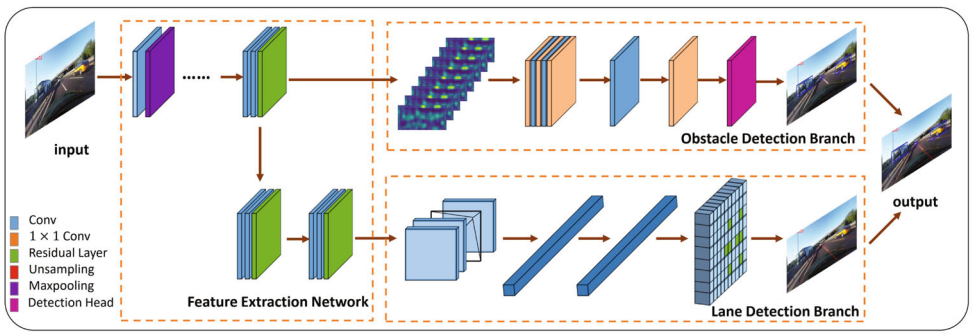

Fig. 2 ULODNet architecture: the backbone is the shared feature map extraction network of original images; the upper branch is the obstacle detection branch which can predict the obstacle coordinates; the bottom branch is the lane detection branch that conducts grid classifications based on row-anchors

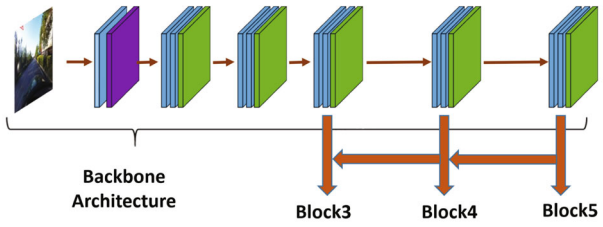

Fig. 3 Multi-Scale detection mechanism: The receptive field of block 5 is appropriate for large-size obstacles, block 4 is appropriate for middle-size obstacles and block 3 is appropriate for small-size obstacle

Overall, the lane detection branch consists of four components: (i) backbone: as stated in Section 3.1, the lane detection branch utilize same feature extraction network, but only $\Phi_5 \in R^{C^f \times H^f \times W^f}$ from the last block serves as the feature passing to the following phases. (ii) spatial convolution modules: to better help lane detection branch integrate more globally, we append the spatial convolution modules after the backbone network, which utilizes four different convolution directions (Up to Down, Down to Up, Left to Right, Right to Left) replacing the traditional convolution layer. (iii) Pooling Layer: a 1×1 convolution layer is adapted acting as the pooling layer to reduce the feature channel to 8 dimensions. (iv) Detection Decoder: the feature vector is finally processed by detection head block to output the lane localization, which is designed as FullyConnected-ReLu-FullyConnected architecture. At last, the lane detection branch uses the same loss function as [6].

Discussion: Since lane is thin and long, the accuracy of lane detection relies more heavily on global information than the normal objects. It can help by merging the context from other lanes, which is crucial in some cases such as occlusion or no visible lane. For this reason, the spatial convolution modules are attached after backbone. As illustrated in Fig. 4, the mechanism behind the spatial convolution modules is that it tears the feature map into many slices along different directions and correspondingly defines diversified convolution directions taking over the traditional Front to Back direction convolution. We also conduct some ablation studies in Section 4.4 discussing the gains brought by the spatial convolution modules.

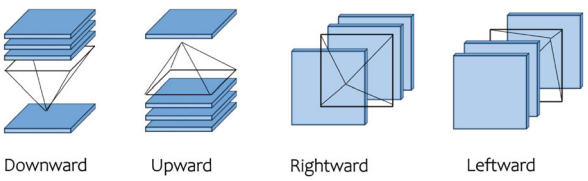

Fig. 4 Four different convolution directions in SCNN: Downward (Up to Down), Upward (Down to Up), Rightward (Left to Right) and Leftward (Right to Left)

4 Experiments

In this Section, Section 4.1 first takes a look at the newly created dataset CULane-ULOD Dataset; Section 4.2 discusses the uneven effects on the results made by the different backbones; then the training procedure of the obstacle detection sub-network is displayed in Section 4.3; and at last, Section 4.4 summarizes the training details of the lane detection sub-network.

4.1 Dataset

To better train and promote the proposed network, a new benchmark named CULane-ULOD Dataset transferred from CULane Lane Detection Dataset [1] is created to serve as the following experiments platform. The original dataset consists of more than 133 million frames, including the normal road scenario and 8 advanced road scenarios (crowded, night, no line, shadow, arrow, dazzling light, curve, cross-road). In order to better adapt to the autonomous vehicle working environments, CULane-ULOD Dataset have made several amendments compared to the original dataset.

First, the original dataset only contains the annotations of road lines since it is made to complete the single challenge of detecting road lines. But this paper aims to detect lane and obstacle jointly to help vehicles drive autonomously in an unfamiliar environment. So 80 classes of common obstacles (e.g., car, truck, pedestrian) are picked from CULane Dataset to make up the lack of obstacle annotations imitating COCO Dataset [2]. But our labeling methods have two main differences with COCO: (i) COCO Dataset assigns a concrete category to every object because it is designed for a precise object detection task. However, some labels such as food and animal seldom exist in the driving environment, so we remove the redundant classes. (ii) From the view of drivers, they do not pay much attention on recognize the exact classes of obstacles, only a label ***avoidance*** can help them react to different traffic conditions. Thus, the final label has only one overall class: ***avoidance***. We first utilize a trained object detection network to detect the obstacles on the road and then check all the images manually. Altogether 253 thousand obstacles are marked manually to serve as the prior ***avoidance*** label, the label utilizes 5 parameters to indicate the obstacle information: a bool type parameter p for the probability of an obstacle and four float type parameters $[x, y, w, h]$ pointing out the coordinates of bounding boxes, which are all within a restriction of $[0, 1]$.

Second, In case of the utilization of image segmentation information, the original dataset also contains the additional segmentation annotations for each frame to help predict more accurate road lane information. However, this paper aims to construct a high performance-price convolution network, so the relevant segmentation annotations are ignored to help save computer memory.

At last, to help the future researches, we also measure some basic obstacle property distributions to help better understand the CULane-ULOD Dataset. Fig. 5 illustrates how the metrics are defined and Table 1 lists the results.

4.2 Backbone

Aiming at adapting to the unified lane and obstacle detection challenge better and saving more computation resources, the backbone in ULODNet is designed to be shared by the obstacle detection branch and lane detection branch. Hence, the backbone is trained alone before the training of these two detection sub-branches. We look forward to finding a light but accurate backbone network to extract the global feature of the input which can be transferred to the following detection tasks and finally ResNet series networks [3], DarkNet network [4, 10, 11] and VGG series networks [33] are picked up to act as the backbone alternatives. In the real experiment, we use the Pytorch official open-source weights applied to the backbone modules. Table 2 lists some crucial properties of the different backbone choices, including model size, GMACs, and classification accuracy. In Section 5, the extensive experiments are based on the ResNet-18 considering the highest cost-performance.

4.3 Obstacle Detection Branch Training

Training Parameters. In the training process, the total loss in the obstacle detection layer consists of MSE (Mean Squared Error) Loss applied on the coordinate predictions and BCE (Binary Cross Entropy) Loss applied on the confidence predictions. Adam algorithm with default settings ($learning\ rate = 10^{-3},\ \beta = (0.9, 0.999)$) acts as the optimizer. The batch size is set to 8 and the training epoch is set to 25 because the experiments show some clues of over-fitting after 15 epochs.

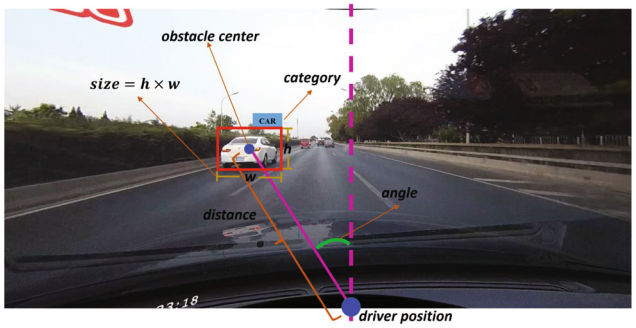

Fig. 5 The graphical interpretation of the obstacle measuring metrics of category, size, distance and angle

Metrics. We use the mAP (mean average precision) metric for the obstacle detection branch. First, we calculate the IoUs (Intersection-over-Union) between predictions and ground-truths. Second, we classify the predictions as true positive (TP) if the IoU is larger than 0.5, or false positive (FP). we also rate the ground-trues with no predictions as false negative (FN). Third, we collect all the predictions made for avoidances in all the images and rank them in descending order according to the predicted confidence score. Then, we plot the P-R Curve, the set of x element is the recall parameter, where $Recall = \frac{TP}{TP+FN}$ and the set of y element is the precision parameter, where $Precision = \frac{TP}{TP+FP}$, we calculate Average Precision (AP) as follows: $AP = \int_0^1 p(r)\,dr$.Since CULane-ULOD Dataset has only one class, $mAP = AP$. Results are shown in Table 3.

Ablation Study. In Section 3.1, we discuss the architecture of the obstacle detection branch. The backbone extraction network consists of 5 convolution blocks which output the feature maps with variant sizes. The feature maps from different blocks have different impacts on the detection of obstacles in multiple scales. Table 4 displays the comparison of the effects made by block3, block4 and block5. We draw some conclusions from it: (i) Feature maps from block3 give the best experiment results and achieve the best precision compared to the other two blocks. This is not surprising as the extremely small or large obstacles seldom exist in the driving environments, which usually have relatively regular sizes. (ii) The inference time have no significant effects caused by the different block combinations.

Results. There are diversified backbone extraction network combinations trained for the obstacle branch, as can be seen in Table 3. Different backbone extraction modules have similar influences on the detection precision. DarkNet-53 [11] reaches the highest of 0.837 and VGG-19 [33] gets the lowest of 0.788.

4.4 Lane Detection Branch Training

Training Parameters. In the training process, Structural Loss following the research [6] is utilized to serve as the loss function and the SGD algorithm with the learning rate of 0.1, momentum of 0.9 and weight decay of 10^{-4} is utilized to serve as the optimizer. Besides, the batch size is set to 8 and the training epoch is set to 30 because the evaluation metrics do not improve any more in the validation set after about 20 consecutive epochs.

Table 1 Distributions of CULane-ULOD Dataset Metrics

Category	*person*	***vehicle***[1]	*others*		
	7.1%	**90.1%**[2]	2.8%		
Size ($kpixels$)	[**0**, **10**]	(10, 15]	(15, 20]	(20, 25]	(25, MAX]
	51.7%	8.2%	5.1%	4.3%	30.7%
Distance ($pixels$)	[0, 200]	(**200**, **400**]	(400, 600]	(600, 800]	(800, MAX]
	3.1%	**61.2%**	23%	12.4%	0.3%
Angle	[−20°, 20°]	± (20°, 45°]	± (**45°**, **75°**]	± (75°, 90°]	
	33%	29.5%	**35.6%**	1.9%	

[1]Class *vehicle* includes *car*, *truck* and *bus*

[2]The bold entries mean the highest proportion

Table 2 Backbone Choices Comparison

Model	Top-1 Acc(%)	Top-5 Acc(%)	Size (MB)	Parameters[1] $\times 10^7$	GMACs[2] @(416 × 416)[3]	GMACs @(288 × 800)[4]
DarkNet-53	77.2	93.8	155	4.06	24.62	32.77
VGG-16	71.59	90.38	528	1.47	53.045	70.622
VGG-19	72.38	90.88	548	2.0	67.408	89.744
ResNet-18	69.76	89.08	**45**[5]	**1.12**	**6.28**	**8.37**
ResNet-34	73.3	91.42	83	2.13	12.68	16.88
ResNet-50	76.15	92.87	98	2.35	14.21	18.92
ResNet-101	77.37	93.56	170.4	4.25	27.07	36.04

[1]the parameters only include the basic convolution layers of the backbone

[2]GMACs means billion Multiply-Add operations like $k \times x + b$ acted on the tensor

[3]416 × 416 is the input image size trained by obstacle detection branch

[4]288 × 800 is the input image size trained by lane detection branch

[5]The bold entries mean the best performance compared to others

Table 3 Training Results of ULODNet

Backbone	DarkNet-53	VGG-16	VGG-19	ResNet-18	ResNet-34	ResNet-50	ResNet-101
OD mAP	0.837	0.816	0.788	0.819	0.799	0.828	0.836
LD Acc	–	0.706	0.709	0.686	0.697	0.696	0.6997

Table 4 Multi-Scale Detection Decoder Ablation Study

Model[1]	mAP	Runtime(ms)[2]
block 3	**0.8533**[3]	110.41
block 4	0.7719	111.04
block 5	0.5865	112.43
block 3 + block 4	0.8446	110.38
block 3 + block 5	0.8223	110.09
block 4 + block 5	0.7339	111.62
block 3 + block 4 + block 5	0.8192	113.99

[1]The backbone of the experiments is ResNet18 [3]

[2]All the runtime tests are conducted on NVIDIA GeForce 940MX

[3]The bold entry means the highest mAP

Table 5 Spatial CNN Module Ablation Study

Model[1]	Accuracy	Runtime(ms)[2]
Baseline(No SCNN Module)	0.68636	44.67
Downward+Upward	0.68639	46.89
Rightward+Leftward	0.6915	49.19
Downward+Rightward	0.6882	49.48
Upward+Leftward	0.6873	47.72
Downward+Upward+ Rightward+Leftward	**0.6976**[3]	51.73

[1]The backbone of the experiments is ResNet18 [3]

[2]All the runtime tests are conducted on NVIDIA GeForce 940MX

[3]The bold entry means the highest accuracy

Metrics. For the lane detection branch, the metric to test whether a lane is correctly marked is F_1. Each lane is considered as a 30-pixels-width line and it will be recognized as true positive (TP) only when the IoU (Intersection-over-Union) between ground-truth and prediction is greater than 0.5. Otherwise, the lane will be rated as false positive (FP) when there is no existing ground-truths matching the prediction or false negative (FN) when there is no predictions matching a specific ground-truth. Then the F_1 measure is defined as follows: $F_1 = \frac{2\times Precision\times Recall}{Precision+Recall}$, where $Precision = \frac{TP}{TP+FP}$ and $Recall = \frac{TP}{TP+FN}$.

Ablation Study. As stated in Section 3.2, we append some spatial convolution modules of diversified convolution directions between the backbone and pooling layer expecting to collect more global information. Table 5 displays the effects made by the appended spatial convolution modules and we have a few observations from it: (i) Adding spatial convolution modules can improve the detection accuracy and that is a proof of the effectiveness of spatial convolution modules. (ii) However, the spatial convolution modules can degrade the inference time to some extent.

Results. We train different versions of lane detection branches with variant backbone extraction networks, which is listed in Table 3. VGG-based backbones reach the highest accuracy and in the test environment of Section 5, four different spatial convolution modules are all appended.

5 Extensive Research

To construct a complete autonomous vehicle driving scheme, some extensive researches of ULODNet are conducted. As demonstrated in Sections 3 and 4, ULODNet predicts the road lane key points and obstacle coordinates jointly, but it does not integrate the information or order any driving instructions. Vehicles still remain perplexed when they receive the output information from ULODNet. Hence, An area intersection paradigm is designed to help process it comprehensively.

Accordingly to the actual driving situations, the freespace region which is defined as the region where vehicles drive freely is the decisive factor for the instruction issuance. In this paper, we measure the obstacle occupied region as the obstacle interference. Excluding the obstacle interference, we can calculate the freespace region and at last the driving guidance commands (e.g., turn left, turn right, accelerate or stop immediately) are published on the basis of the obstacle interference proportion. The algorithm diagram is shown in Fig. 6.

The obstacle detection results from ULODNet are the obstacle coordinates $[x, y, w, h, conf]$ and the lane detection results are the anchor location collections $X = [x_1, x_2, ..., x_N]$. First, we draw the lane pixel predictions and obstacle coordinates on the input image as shown in Fig. 6(a). This type of additional information is not intuitive enough to be understood by the self-driving system. Thus, the boundary coordinates of a specific detected obstacle are calculated and pixels within the boundaries will be recognized as obstacle inference for the decoding procedure shown in Fig. 6(b). As for the lane detection results, we frame two adjacent lane prediction area as the lane regions shown in Fig. 6(c). Normally, there are four road lanes extracted from the ULODNet and the lane region can be therefore separated into 3 sub-regions. Finally, we utilize the polygon intersection algorithm to restrict the freespace region which are defined as the lane regions excluding the obstacle inferences as shown in Fig. 6(d).

As for the decision-making portion, the driver is always located at the middle of the image bottom for the relative

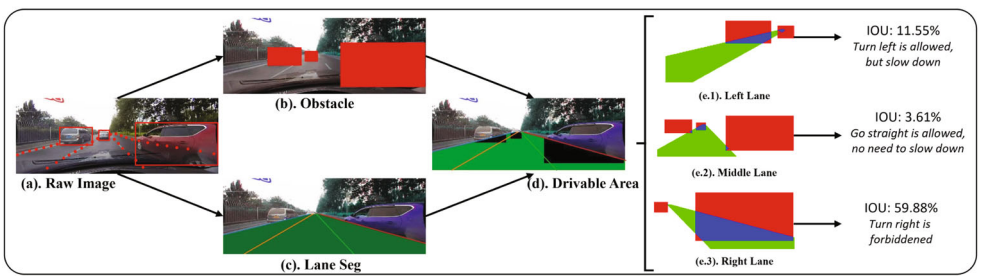

Fig. 6 The extensive research processing flow: driving suggestions are judged according to the comparison results between the intersection area and preset thresholds

Fig. 7 The experiment results on different scenarios of the CULane-ULOD test set, from left-top to bottom-right: normal, crowd, dazzling light, shadow, curve, arrow, noline, cross and night

coordinates of the driver and camera remain fixed in the data collection process and it is constrained that the driver must not cross two lane regions at a time. Empirically, the closer distance between the obstacle and the driver is, the larger area of the obstacle is. Consequently, the proportion of obstacle inference region in freespace become larger. Thus, different freespace regions are judged whether drivable on the basis of the proportion of the obstacle inferences in freespace region area as shown in Fig. 6(e) and the following formula.

$$traffic\ condition = \begin{cases} free, & proportion \leq 8\% \\ slower, & 8\% < proportion \leq 25\% \\ congested, & proportion > 25\% \end{cases}$$

Besides, ULODNet and the extensive paradigm are tested on the test set of the modified CULane-ULOD Dataset. The test set includes normal and 8 challenging categories, which correspond to the 9 different driving scenes in the real world. The formulation runs on each category separately and the results show that the algorithm performs well in most cases except dazzling light scenario. Considering that the dazzling light scenario disturbs the light so significantly that even human drivers have tremendous difficulties in recognizing road lanes and the extreme scenario seldom exists in the vehicle working environments, ULODNet helps vehicles accomplish the autonomous driving tasks to a large extent. The details of the test outcomes are shown in Fig. 7.

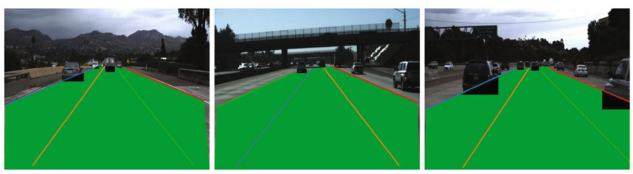

Fig. 8 The experiment samples on TuSimple test set in highway environment

At last, to further test the robustness of the well-designed ULODNet, we transfer to the TuSimple Dataset which is another popular dataset in the self-driving research fields designed by TuSimple Technology Company. The dataset contains the major typical scenes in a highway environment. 3626 frames are collected in the train set and 2782 frames are collected in the test set. ULODNet views each image as an input and outputs the alternative driving suggestions to help vehicles realize autonomous driving. Given the prediction results shown in Fig. 8, the model can give the appropriate instructions in the highway occasions.

6 Conclusions

In this paper, we propose ULODNet to detect the lane and obstacle jointly in the autonomous driving environments. ULODNet consists of obstacle detection branch and lane detection branch. We design both the sub-branches benefiting from the achievements in convolution neural network research fields and they reach a high detection accuracy as well as a fast inference time. To help better train the model, we transfer CULane Dataset to create a new dataset, CULane-ULOD Dataset. Furthermore, we also propose an area intersection algorithm to construct an integrated autonomous driving scheme. It merges the information from ULODNet to help issue the driving commands to the driver. At last, the conducted experiments on CULane-ULOD Dataset and Tusimple Dataset illustrate the effectiveness and robustness of the algorithm.

Acknowledgements This work was supported in part by the National Natural Science Foundation of China under Grant 61922076, Grant 61873252, and Grant 61725304.

Author Contributions Zhanpeng Zhang: Coding and writing; Jiahu Qin: Writing and review; Shuai Wang: Writing and review; Yu Kang: Review; Qingchen Liu: Review.

Funding This work was supported in part by the National Natural Science Foundation of China under Grant 61922076, Grant 61873252, and Grant 61725304.

Availability of data and materials Our CULane-ULOD Dataset is open source at https://rec.ustc.edu.cn/share/9f97cc30-00f2-11ec-b059-a7276527f2db. The demo video of our proposed autonomous scheme is available in https://rec.ustc.edu.cn/share/f6c83d00-ff47-11eb-bf42-07a0d8061db3.

Code Availability The project is open source at https://github.com/phosphenesvision/ULODNet.

Declarations

Conflict of interest/Competing interests (check journal-specific guidelines for which heading to use) The authors have no relevant financial or non-financial interests to disclose.

References

1. Pan, X., Shi, J., Luo, P., Wang, X., Tang, X.: Spatial as deep: Spatial cnn for traffic scene understanding. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32, p. 1 (2018)
2. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European conference on computer vision, Springer, pp. 740–755 (2014)
3. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770–778 (2016)
4. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 779–788 (2016)
5. Neven, D., De Brabandere, B., Georgoulis, S., Proesmans, M., Van Gool, L.: Towards end-to-end lane detection: an instance segmentation approach. In: 2018 IEEE intelligent vehicles symposium (IV), IEEE, pp. 286–291 (2018)
6. Qin, Z., Wang, H., Li, X.: Ultra fast structure-aware deep lane detection, arXiv:2004.11757 (2020)
7. Qian, Y., Dolan, J., Yang, M.: DLT-Net: Joint detection of drivable areas, lane lines, and traffic objects. IEEE Trans. Intell. Transp. Syst. **21**(11), 4670–4679 (2019)
8. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 580–587 (2014)
9. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn:, Towards real-time object detection with region proposal networks, arXiv:1506.01497 (2015)
10. Redmon, J., Farhadi, A.: Yolo9000: better, faster, stronger. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 7263–7271 (2017)
11. Redmon, J., Farhadi, A.: Yolov3:, An incremental improvement, arXiv:1804.02767 (2018)
12. Bochkovskiy, A., Wang, C.Y., Liao, H.Y.M.: Yolov4:, Optimal speed and accuracy of object detection, arXiv:2004.10934 (2020)
13. Tusimple benchmark: https://github.com/TuSimple/tusimple-benchmark.AccessedSeptember (2020)
14. Zou, Z., Shi, Z., Guo, Y., Ye, J.: Object detection in 20 years:, A survey, arXiv:1905.05055 (2019)
15. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2117–2125 (2017)
16. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision, pp. 2961–2969 (2017)
17. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3431–3440 (2015)
18. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: ssd: Single shot multibox detector. In: European conference on computer vision, Springer, pp. 21–37 (2016)
19. Huang, R., Pedoeem, J., Chen, C.: yolo-lite: a real-time object detection algorithm optimized for non-gpu computers. In: 2018 IEEE International Conference on Big Data (Big Data), IEEE, pp. 2503–2510 (2018)
20. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision, pp. 2980–2988 (2017)
21. Almeida, T., Santos, V., Mozos, O.M., Loureno, B.: Comparative analysis of deep neural networks for the detection and decoding of data matrix landmarks in cluttered indoor environments. Journal of Intelligent & Robotic Systems **103**(1), 1–14 (2021)
22. Silva, I., Perico, D.H., Homem, T., Bianchi, R.: Deep reinforcement learning for a humanoid robot soccer player. J. Intell. Robot. Syst. **102**(3), 69 (2021)
23. Law, H., Deng, J.: Cornernet: Detecting objects as paired keypoints. In: Proceedings of the European conference on computer vision (ECCV), pp. 734–750 (2018)
24. Law, H., Teng, Y., Russakovsky, O., Deng, J.: Cornernet-lite:, Efficient keypoint based object detection, arXiv:1904.08900 (2019)
25. Duan, K., Bai, S., Xie, L., Qi, H., Huang, Q., Tian, Q.: Centernet: Keypoint triplets for object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 6569–6578 (2019)
26. Tian, Z., Shen, C., Chen, H., He, T.: Fcos: Fully convolutional one-stage object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 9627–9636 (2019)
27. Liu, Z., Zheng, T., Xu, G., Yang, Z., Liu, H., Cai, D.: Training-time-friendly network for real-time object detection. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 11685–11692 (2020)
28. Aly M.: Real time detection of lane markers in urban streets. In: 2008 IEEE Intelligent Vehicles Symposium, IEEE, pp. 7–12 (2008)
29. Wang, Y., Teoh, E.K., Shen, D.: Lane detection and tracking using b-snake. Image Vis. Comput. **22**(4), 269–280 (2004)
30. Jung, S., Youn, J., Sull, S.: Efficient lane detection based on spatiotemporal images. IEEE Trans. Intell. Transp. Syst. **17**(1), 289–295 (2015)
31. Tabelini, L., Berriel, R., Paixao, T.M., Badue, C., De Souza, A.F., Oliveira-Santos, T.: Polylanenet:, Lane estimation via deep polynomial regression, arXiv:2004.10924 (2020)
32. Zheng, T., Fang, H., Zhang, Y., Tang, W., Yang, Z., Liu, H., Cai, D.: Resa:, Recurrent feature-shift aggregator for lane detection, arXiv:2008.13719 (2020)
33. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition, arXiv:1409.1556 (2014)

Zhanpeng Zhang received the B.E. degree in automation from University of Science and Technology of China, Hefei, China, in 2019. He is currently pursuing the M.S. degree in Automation with the University of Science and Technology of China, Hefei, China. His current research interests include perception and control in robotics.

Jiahu Qin received the first Ph.D. degree in control science and engineering from Harbin Institute of Technology, Harbin, China, in 2012, and the second Ph.D. degree in systems and control from the Australian National University, Canberra, ACT, Australia, in 2014. He is currently a Professor with the Department of Automation, University of Science and Technology of China, Hefei, China. His current research interests include multiagent systems, cyber-physical systems, and complex dynamical networks.

Shuai Wang received the B.E. degree in automation from Northeast Forestry University, Harbin, China in 2016. He is currently a Ph.D. candidate in control science and engineering at the University of Science and Technology of China, Hefei, China. His current research interests include perception and control in robotics.

Yu Kang received the Dr. Eng. degree in control theory and control engineering from the University of Science and Technology of China in 2005. From 2005 to 2007, he was a Post-Doctoral Fellow with the Academy of Mathematics and Systems Science, Chinese Academy of Sciences. He is currently a Professor with the Department of Automation, University of Science and Technology of China. His current research interests include adaptive/robust control, variable structure control, mobile manipulator, and Markovian jump systems.

Qingchen Liu received the Ph.D. degree in system and control from Australian National University, Canberra, ACT, Australia, in 2018. He is currently an EuroTec Research Fellow within the Chair of Information-Oriented Control, Technical University of Munich, Munich, Germany. His current research interests include networked systems, distributed computation, and multiagent systems.