

# **A Deep Learning Based Implementation for Self-Driving Car.**

VU THANH DAT

ICT Department, FPT University, Hanoi, Vietnam

Email: datvthe140592@fpt.edu.vn

PHAN QUANG HUY

ICT Department, FPT University, Hanoi, Vietnam

Email: huypqhe141762@fpt.edu.vn

NGUYEN DINH TRA

ICT Department, FPT University, Hanoi, Vietnam

Email: trandhe140661@fpt.edu.vn

TRAN HAI ANH

ICT Department, FPT University, Hanoi, Vietnam

Email: anhthhe141545@fpt.edu.vn

NGO TUNG SON

ICT Department, FPT University, Hanoi, Vietnam

Department of Computer and Information Sciences, Universiti Teknologi PETRONAS, Tronoh, Malaysia

Email: sonnt69@fpt.edu.vn

Today, self-driving cars are a part of our life. It has received much attention in recent years. Many big companies and developers have invested a lot in this area and developed their own autonomous driving car platforms. The intriguing area of self-driving car motivates us to build a self-driving platform. This paper proposes the self-driving car's architecture and its software components that have been solved in FPT's contest. Lane detection in different environmental conditions, dodging obstacles, and detecting traffic signs. In this competition, the vehicle is equipped with limited hardware such as a single low-cost camera, an Nvidia Jetson TX2 board. We analyze the results obtained in the game in the simulator. We see that our method has overcome limited hardware but still achieved good results in complex problems. The final product has been used to compete in the Digital Race competition 2020 - a competition held annually by FPT Corporation.

**CCS CONCEPTS • Computer systems organization → Embedded systems; Redundancy; Robotics;**

**Additional Keywords:** autonomous vehicle, neural networks, object detection, semantic segmentation, computer vision.

**ACM Reference Format:**

Vu Thanh Dat, Phan Quang Huy, Nguyen Dinh Tra, Tran Hai An, and Ngo Tung Son, 2021. A deep learning based implementation for self-driving car. In ICSCA 2021: 2021 10th International Conference on Software and Computer Applications, February 23-26, 2021, Kuala Lumpur, Malaysia. ACM, New York, NY, USA, XX pages. <https://doi.org/> (the doi information will be completed after uploading copyright to press).

## 1 INTRODUCTION

### 1.1 Research Context

Autonomous cars are closer and closer to becoming a reality. Enormous challenges are exciting for autonomous vehicles to be entirely driven. Digital race is a competition organized annually by FPT Corporation to create a playground to promote Vietnamese students' innovative spirit in the fourth industrial revolution. After three years of organization, in the 4th season, the problems have been upgraded in complexity, requiring an advanced computer vision application, machine learning, and deep learning. There are emotional obstacle problems and more complex garage entry than the previous year and lane detection and sign detection problems. The contest organizers allocated vehicles with the Traxxas Desert Racer model with 1/7 real vehicles' size with the Nvidia Jetson TX2 board's processing hardware with GPU computing support 256 CUDA cores speed up to 1.3 TFLOPs. The sonic sensor, LiDAR 2D, is added to handle dynamic collision detection on the sensor part. In this research, we discussed autonomous cars' main objectives: perception, motion planning, navigation. First of all, lane detection is the main task to teach the car how to see. It is a part of the perception Layer. Our goal is to set our privileges so that the vehicle can see even in edge-situations such as snow, unqualified light. With grown technology, road traffic is more and more complicated. Therefore, traffic signs are essential research for real-time recognition. The third task, dynamic obstacle detection, is a challenging task. Not only static objects but also dynamic objects, our car has to recognize to avoid accidents. Nevertheless, we construct a motion planner with the above as input to have the desired output as an adaptive, safe path planner.

Path planning for automobile cars needs to discover a collision-free path through the robot's current circumstance with obstacles from the determined start location to the objective target area while fulfilling certain advancement conditions [1]. When a portable car performs a territory inclusion task, culmination and effectiveness of inclusion are significant components. To accomplish fulfillment effectively, we adopt a divide and conquer strategy. We built up a novel cell decomposition algorithm that partitions a given territory into a few cells [2]. For lane analysis, using semantic segmentation performs excellent benefits. The road is constructed after processing RGB images as input. To localize traffic signs, segmentation segments region of interests. Therefore, we can draw the bounding boxes from those regions. After segmentation of the road, steering angle computation for efficient navigation base on drivable road. After road region extraction using segmentation, the steering angle is based on the road's extracted edges. For receiving signals, path loss occurs when the signal is not sufficiently efficient. The model needs to construct interpolated roads based on which vehicles have been seen. The results were achieved by using Gaussian processes interpolation algorithms.

### 1.2 Related Researches

The main points we discussed above are the self-driving car consisting of three modules: perception, motion planning, navigation. We have three problems recognizing the object as traffic signs, dynamic obstacles, and lane detection to perception. Learning-based pattern recognition has been widely studied [3, 4]. They leverage CNNs for pattern recognition, such as traffic sign detection problems [5]. Object detection aims to recognize the

pattern and describe each detected object's locations in the image using a bounding box. So far, there are few one-stage approaches towards object detection as SSD (Single-shot detection) [6] or Yolo (You only live once) [7], to run in real-time applications and achieved good results. A popular approach is based on LiDAR-2D [8, 9, 10] or implementing a depth-map to detect near objects [11] to dynamic obstacle detection. We use LiDAR 2D to scan and implement K-Means [12] to cluster data points with obstacle avoidance. If cluster in our region, we request the system to respond to an action. Many methods to perform as hough-line transform [13] or road segmentation use modern Deep learning approaches [14, 15, 16, 17] to Lane detection. There is an essential task of the self-driving car.

So far, deep learning is a robust framework to build models that can be learned as humans. There is our approach to self-driving cars. Semantic segmentation is a computer vision domain in which we use CNN architecture [18] to feature target objects. The goal of semantic segmentation is to label each pixel of the class's region. The famous segmentation architecture is U-Net [19]. The architecture consists of two parts. The first part is the contraction path (also called the encoder) to capture the image's context. The encoder is a traditional stack of convolutional and max-pooling layers. The second part is the symmetric path (also called the decoder), which enables precise localization using transposed convolutions. U-Net, perform a good result in the semantic segmentation task. In this paper, we focus on PSP-Net [20] is developed based on U-Net.

PSP-Net is the winner in ILSVRC 2016 (Semantic Segmentation / Scene Parsing). It got the champion of ImageNet Scene Parsing Challenge 2016 and arrived in first place on PASCAL VOC 2012 and Cityscapes datasets at that moment. PSP-Net solved three problems in semantic segmentation is the mismatched relationship, confusion categories, and inconspicuous classes. ResNet [21] is used with a dilated network strategy for extracting features and use sub-region average pooling to concrete for context aggregation. Finally, it generates the final prediction map. PSP-Net outperforms U-Net about all tasks, and it is so good to run in real-time applications. After we obtain image segmentation, we classify the traffic sign based on the sign's region, segmented by computing the centroid of the segmented-region to localize and crop the image to a small CNN architecture.

To motion planning, the final goal of self-driving cars on how to predict angles to steer. The popular approach is the car's path planning on a scanned map [22, 23]. The standard process in which papers mention a scanned map by combining multiple sensors (fusion sensor) as the depth camera, lidar to build a plan is called a global map. They initialize a starter-point and end-point and try to find the shortest path by implementing the Dijkstra algorithm [24] or A\* algorithm [25]. During the self-driving model, generate local maps and local directions. All local paths ensure we can find a final target on the global map. This approach is one of the most efficient methods to steer a self-driving car. This method's disadvantage requires plans that must have many specific features, such as the wall's edge. Therefore, a scanned map that we obtain from Digital Race's organizers' request cannot use this method because our plan is flat, and LIDAR does not have to get any features. Besides, map scanning requires powerful hardware. So far, almost all methods related to path planning often utilize on GPU and the frame we only archive approximately 5-10 fps on Jetson-TX2 [26]. For Digital Race, we need to steer with a high speed (10-15 m/s) to finish the race soon, so path planning is not suitable in this case.

### 1.3 Contribution

In this paper, we focus on developing perception and motion planning. To be clear, we build an architecture neural network to segment the road and classify traffic signs. To dynamic obstacle avoidance, we use LiDAR-

2D to scan the region that can have obstacles. Conclusively, we compute the centroid of the road to predict angle and perform experiments to measure the road has expected the center's degree of error. For the digital race contest, sensors are provided by the organization contain an Orbbec Astra camera, RP-LiDAR A2, IMU MPU9250, and Jetson-TX2. To build an end-to-end autonomous car require high-priced sensors as LiDAR 3D as Waymo LiDAR or powerful GPU hardware like the GTX series. Therefore, the proposed system is suitable for the environment as contests related to autonomous cars or lightweight systems like self-driving golf cars, small autonomous trucking on amazon services, and real-time applications.

## 2 PROPOSED METHOD

Scene parsing is one of the crucial topics in computer vision: to segment objects, parse images to different regions and assign each pixel a category label such as sky, building, car, and boat. Pyramid Scene Parsing Network (i.e., PSP-Net) has surpassed FCN. PSP-Net provides global contextual prior due to FCN. The needed global context is diversity, divided into three parts: Mismatched Relationship, Confusion Categories, and Inconspicuous Classes. Achieves accurate pixel-level prediction, knowledge for information of scene context is essential. PSP-Net gives the ability to understand not only the local-level context but also the global-level environmental context. Due to our problems, recognizing inside roads and ambiguous roads outside is challenging when applying PSP-Net and achieving incredible results.

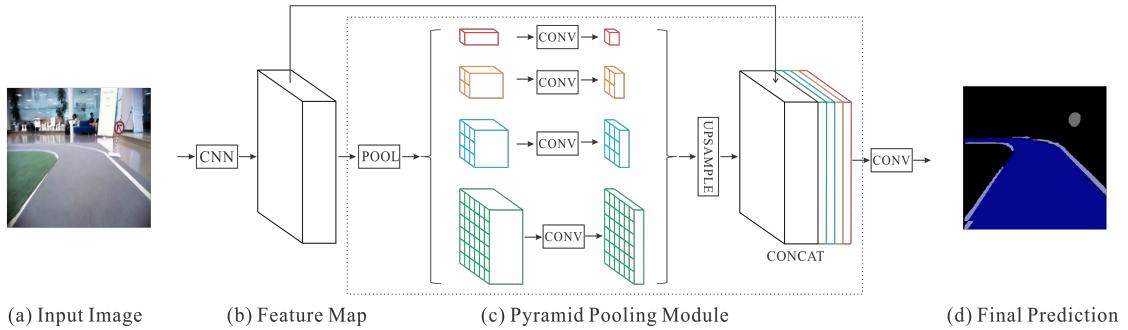


Fig 1: PSP-Net architecture.

We apply PSP-Net architecture with backbone EfficientNetB3, which is trained on ImageNet. The size of the input image is 144x144. Our model consists of four classes: line, road, traffic signs, background. EfficientNetB3 pre-trained model used for extracting feature maps. Figure 1 shows the prediction result from PSP-Net architecture. One of the critical problems in semantic segmentation is imbalanced classes. In our case, the background account for 60%, and the traffic signs only account for 1%. We use Focal Tversky Loss to solve this issue, and One of the critical problems in semantic segmentation is imbalanced classes. In our case, the background account for 60%, and the traffic signs only account for 1%. To solve this issue, we use Focal Tversky Loss and Weighted Cross-Entropy to optimize results. This idea has been extended in recent works where an exponent is applied to the Dice score combines of Dice and cross-entropy. Similarly, Focal Tversky Loss (FTL) function is defined as  $FTL_c = \sum_c (1 - TI_c)^{\frac{1}{\gamma}}$ . Where gamma varies in the range [1, 3], in practice, if a pixel is misclassified with a high Tversky index, the FTL is unaffected. However, if the Tversky index is small

and the pixel is misclassified, the FTL decreases significantly. We hypothesize using a lower alpha in our generalized loss function improves model convergence by shifting the focus to minimize TP predictions. Therefore, we train all models with alpha = 0.7 and beta = 0.3. It is important to note that in alpha = beta = 0.5, the Tversky index simplifies to the DSC. Moreover, when gamma = 1, the FTL simplifies to the TL.

To calculate the weight of classes, we compute the sum of pixels for each category over the training images' overall pixels. We also applied dice loss for finding boundary regions of types and categorical focal loss for focusing on the imbalanced type, which is so hard for segmentation. We want to obtain high precision and recall for hyper-parameters, so we adjusted beta = 1, and alpha used to specify the weight of different categories labels gamma is the focusing parameter which we set 2.0. We also operate another combination of Jaccard loss and focal loss, but it does not achieve FTL loss and WCE loss results. Metrics for evaluation, we use Mean IoU and F1-Score, precision, and recall. After segmentation from PSP-Net, a gray area presents locations where traffic signs are located. In our images, the color of traffic signs maps to the corresponding (128,128,128) RGB types. With morphological transformations such as opening and closing, noise reductions contain essential features, and the centroid of the sign is conducted from the perspective of the camera, distance matters. We apply a decision tree depending on the density of segmentation signs to get adaptive bounding boxes to rely on the car's traffic sign's distance. After transforming to detect that object, it forwards to the CNN network to classify six classes: left, right, straight, no turn left, no turn right, and stop sign. The architectural system consists of six Convolution 2D layers, three Max-pooling layers, four Drop-out layers, two Dense layers, and 1 Flatten layer. The input layer has a size of 32x32. Figure 2 displays the prediction result of the traffic sign's boundary region. We localized the centroid of the signing class and cropped a patch 32x32 on the camera's input image. As shown in Figure 3, we obtain a patch from cropping.



Fig 2: Objects segmentation



Fig 3: Traffic sign localization.

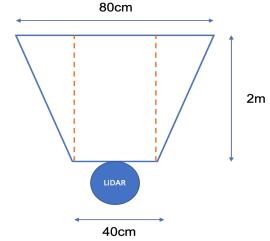


Fig 4: Object Detection using LiDAR

The detection of dynamic obstacles depends entirely on the LiDAR sensor. LiDAR scans the point densities against the regions we initialize. In this case, we observe the point densities in the trapezoidal area for environmental awareness and the rectangle for dynamic obstructions and signage. The trapezoid has a height of 2m, a broad base of 80cm, and a small bottom of 40cm. The rectangle has a height of 2m and a width of 40cm. Trapezoid serves to recognize the surrounding environment in which points are inside the acceptable range. The rectangle is the expected path on the map. Based on the point density where the two feedback shapes, we use the decision tree to identify are a dynamic obstacle or signs to send the message to the navigation module to make the final decisions.

As mentioned in the previous section, motion planning cannot be generated by some popular algorithms such as Dijkstra or A\* algorithms because of hardware limitations. The method we use is a decision tree that depends on the screenings and the organizers' rules. Firstly, with the photo that the organizers gave, we have

five checkpoints corresponding to 5 challenges that the organizers gave. Each passed checkpoint is counted as a score, and the team that completes and finishes the earliest is the winner. We use LiDAR to count the number of signs and use the decision tree algorithm to respond when encountering sign recognition. For example, when we recognize a no-left sign and based on the number of signs collected, the self-driving vehicle decides whether to turn right or keep going straight. In figure 4 shows my obstacle detection from LiDAR. We handle the self-driving cars' angle by computing the centroid of road segmentation to the navigation module. Our algorithm improves results to extract precisely the road region by finding the road region between two-lane lines. To get good two-lane lines segmented by PSP-Net, we use connected components and a dilation algorithm to remove noise and extract good two-lane lines. Figure 5 displays our software architecture that has 3 modules: perceptron, motion planning, navigation. We publish a message over a ROS topic, which is built by MQTT protocol.

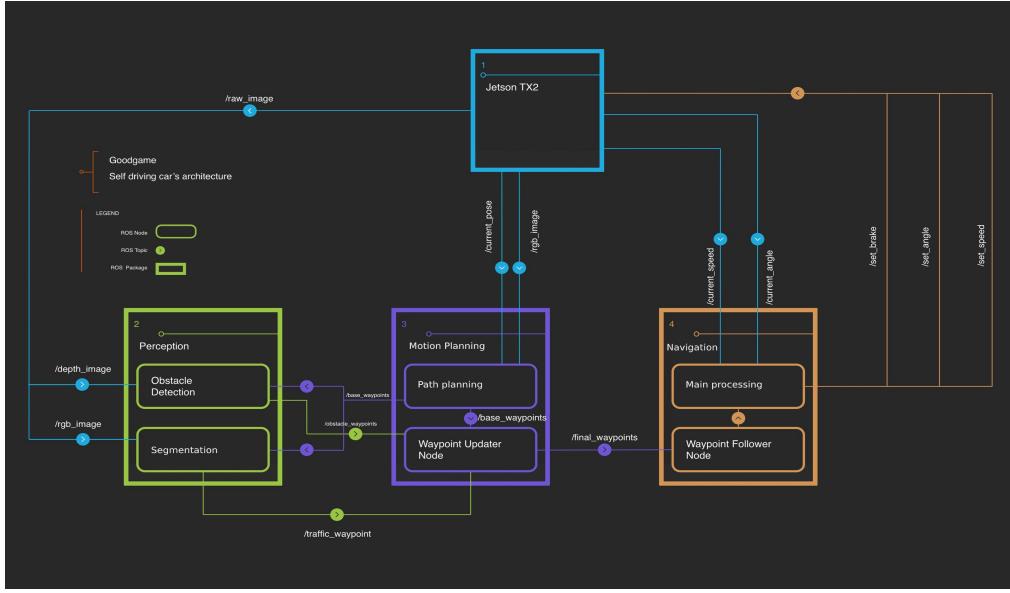


Fig 5: Overview of the proposed system.

### 3 EXPERIMENTS

For the self-driving car to run smoothly, the perceptron module must have a good performance. In this section, we focus on the experiments related to semantic segmentation and object detection task. The semantic segmentation task is the key to steering, and object detection is the key to planning on the motion planning module. We validate the Focal Tversky Loss on our datasets where the ROI class is significantly smaller than the background class and observe large performance gains for semantic segmentation. This dataset consists of 3,200 images. The average image size is 640 x 480 pixels, where each of the images presented one or more lesions. The data is resampled to 148 x 148 pixels with a 75-25 train-test split for our experiments. We have a total of 7 experiments, including testing the results of classification and segmentation algorithms, check the actual deviation from the road center when using the above algorithms, check the level of performance stability

on self-driving cars, self-driving car plotting, and finally check the deviation and stability when meeting the stop sign.

To present a fair evaluation of our PSP-Net and the focal Tversky loss, we do not augment our dataset or incorporate any transfer learning. The CDS 2019 experiment was trained for 50 epochs with a batch size of 8. The models were optimized using Adam with momentum, using an initial learning rate at 0.001, which  $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$  on every epoch. All experiments are programmed using the Keras framework with the Tensorflow backend and trained using an NVIDIA GTX 1080 TI GPU.

Table I. Performance on CDS 2019 Dataset with 3200 images on validation dataset

Model	Parameters	DSC	Precision	Recall
PSP-Net + JL	Class weights	$0.812 \pm 0.04$	$0.852 \pm 0.06$	$0.792 \pm 0.124$
PSP-Net + WDL	Class weights, beta = 1	$0.827 \pm 0.017$	$0.871 \pm 0.016$	$0.832 \pm 0.002$
PSP-Net + TL	alpha = 0.3, beta = 0.7	$0.829 \pm 0.01$	$0.847 \pm 0.027$	$0.892 \pm 0.015$
PSP-Net + FTL	alpha = 0.3, beta = 0.7	$0.831 \pm 0.02$	$0.901 \pm 0.010$	$0.912 \pm 0.01$
PSP-NET + WDL + WCE	Class weight, beta = 1	$0.842 \pm 0.017$	$0.891 \pm 0.021$	$0.93 \pm 0.014$
PSP-NET + FTL + FL gamma = 4/3	alpha = 0.7, beta = 0.3, gamma = 4/3	$0.86 \pm 0.022$	$0.92 \pm 0.013$	$0.95 \pm 0.015$
PSP-NET + FTL + WCE	alpha = 0.7, beta = 0.3, gamma = 4/3	$0.86 \pm 0.025$	$0.91 \pm 0.018$	$0.96 \pm 0.011$

DSC: Dice score coefficient

JL: Jaccard Loss

WDL: Weighted Dice Loss

TL & FTL: Tversky Loss and Tversky Focal Loss

FL & WCE: Focal Loss and Weighted Cross Entropy

Table 1 shows that the baseline PSP-Net trained with the Jaccard loss function has the worst performance. The large standard deviation in the precision and recall scores suggest the learning is not stable. In contrast, PSP-Net models trained with TL and FTL show increased DSC and more balanced precision-recall scores due to weighting alpha higher in the loss function than beta. Training the attention model with FTL combines the benefits of improved feature selection with focused training to outperform all other methods. Training PSP-Net with TL and FTL, we observe an improved DSC score. However, when the Tversky index is high for misclassified examples, the focal exponent gamma suppresses the error signal's contribution. Since alpha is weighted higher than beta, the model converges to the highest reported recall at 0.892, but lowest precision. To address this issue, when training the proposed pyramid architecture model, we supervise the last layer with TL so that a true error signal still propagates back when the model is close to convergence. As a result, our improved attention PSP-Net model with FTL combines WCE (last row) obtains slightly lower but overall better-balanced recall and precision, and, consequently, the best DSC score. We outperform the baseline by 3.5%, with a low spread of 0.25%.

The cityscape is a large-scale dataset that contains images of 50 different cities, which contains 30 classes include road class. Our architecture out-perform the rest of all models in the semantic segmentation tasks on the cityscape dataset. Our road class achieve  $96 \pm 0.05\%$  on F1-score. And meanwhile, the SOTA model: HRNet-OCR has to mean IoU is 85,1%. While the comparison is a bit lame, there are some of the factors which we can determine the model's performance.

Our simple architecture contains six convolutional layers to traffic sign detection. Two hidden layers with 512 neurons achieve a perfect result with accuracy approximate around 100%, and precision and recall approximate about  $95 \pm 2\%$ . To do that, the boundary region of a traffic sign in the semantic segmentation task needs to ensure to do not have the misclassified upper 15%. After obtaining the signed container on the segmentation image, we use the localize method to separate the input image sign. The sign separation is based on our experiment on the results obtained in Table 2. Based on the IoU scores we experimented with, from the center of the segmentation area, we localized according to the decision tree algorithm to ensure that the sign covered the localization area and resized to 32x32 before pushing the network neural network.

.Table II. Performance on Traffic Sign Task with 22,000 images

IoU Conditional	Accuracy	Precision	Recall
Upper 95%	$0.98 \pm 0.001$	$0.986 \pm 0.002$	$0.947 \pm 0.001$
90% – 95%	$0.970 \pm 0.005$	$0.962 \pm 0.004$	$0.911 \pm 0.003$
75% – 90%	$0.924 \pm 0.014$	$0.902 \pm 0.002$	$0.915 \pm 0.015$
65% – 75%	$0.90 \pm 0.001$	$0.882 \pm 0.013$	$0.902 \pm 0.001$
Lower 50% – 65%	$0.885 \pm 0.006$	$0.852 \pm 0.004$	$0.875 \pm 0.001$

Table 3 shows that the CNN trained baseline depends on semantic segmentation task and achieve the worst result on the lower IoU region. To have a good performance on the traffic sign detection task, our segmentation task must achieve a good result on IoU and Dice score. To find the IoU conditional, we did many experiments. We created a decision tree algorithm so that we cut a 32x32 patch on the original frame from the camera from the centroid of the sign region. The patch matches the ground-truth traffic sign to test the IoU score from the Dice score, which is predicted by the segmentation task.

Table III. IoU Conditional experiment

Dice Score	IoU Conditional
0.90 - 0.95	Upper 95%
0.75 - 0.90	90% - 95%
0.60 – 0.75	75% - 90%
0.50 – 0.60	65% - 75%
0.45 – 0.50	Lower 50% - 65%

Figure 6 shows our result on the semantic segmentation task. As demonstrated, the sign class's F1 score is 0.92. Therefore, when we localized by sign's centroid of boundary region, we obtain a patch 32x32 with IoU score  $> 95\%$  to ground truth. After obtaining the signed container on the segmentation image, we use the localize method to separate the input image sign. The sign separation is based on our experiment on the results obtained in Table 3. Based on the IoU scores we experimented with, from the center of the segmentation area, we localized according to the decision tree algorithm to ensure that the sign covered the localization area and resized to 32x32 before pushing the network neural network.

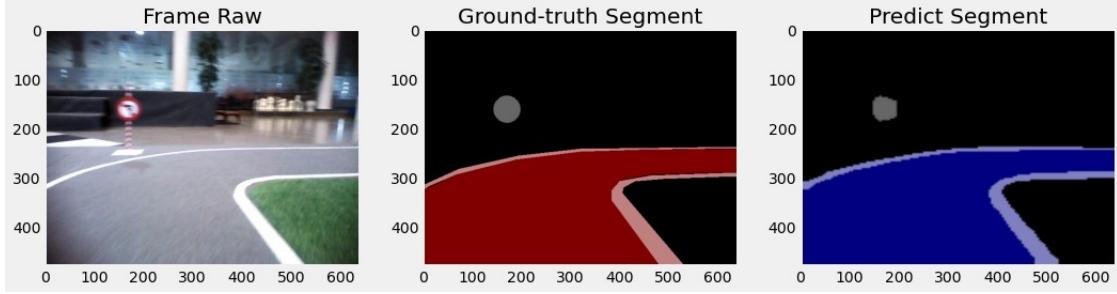


Fig 6. Semantic segmentation task.

In the third experiment, we want to check whether our vehicles move correctly under the set's environmental conditions. To overcome all organizers' challenges, we construct a map that contains 50 checkpoints on the entire map, from which we could identify errors across the whole road frame. According to the proposal set out by the organizers, the car must go in the roadway between two white lines marking the road. From there, we have identified a system of center lines that line the middle of the road. In our diagram below, we draw the centerline with red stripes. Through the segmentation system, the center of the road is predicted to follow the road's center. We perform this experiment to obtain the error of the ratio of the difference to the roadway width.

We label our car and get the car's center point and marked with a blue dot. From this blue point, we determine how much that point deviates from the centerline system. From this distance, we scale it with the width of the road. As you can see, there is a yellow straight line, which is the roadway's actual width at that time. We determine this segment by projecting the vehicle's horizontal axis to that respective road width. We find the intersection between the yellow line and the centerline. We calculate a distance from the center of the blue line that the car passes in the vehicle's center. Then scale it by the width to get the error to be calculated. Figure 7 is the result which we obtain.

Table IV. Deviation of the centroid road

Round	Measure deviation	Angle deviation
D-Shape	$3.67 \pm 2.2$ cm	$2.10 \pm 0.2$ degree
B-Shape	$6.40 \pm 4.1$ cm	$7.45 \pm 0.8$ degree
All-Shape	$4.95 \pm 1.5$ cm	$4.32 \pm 0.6$ degree

In Table 4, we give the experiment of eccentricity with the investigation within 10 rounds of the self-driving car, equivalent to 50 samples (each round, we plot 2 samples in D-Shape, 3-Shape). In D-Shape, the self-driving car's centroid is the most stable with a 2.10 angle deviation equivalent to 95.3% accuracy (compared with the maximum angle allowed to move is 45 degrees). But in B-Shape, the road's length is not stable, so at the intersection, the road's deviation has a considerable variation leading to an accuracy of 85%. To summarize, after going 10 rounds, the average deviation is 4.32 degrees ~ 90.4%.

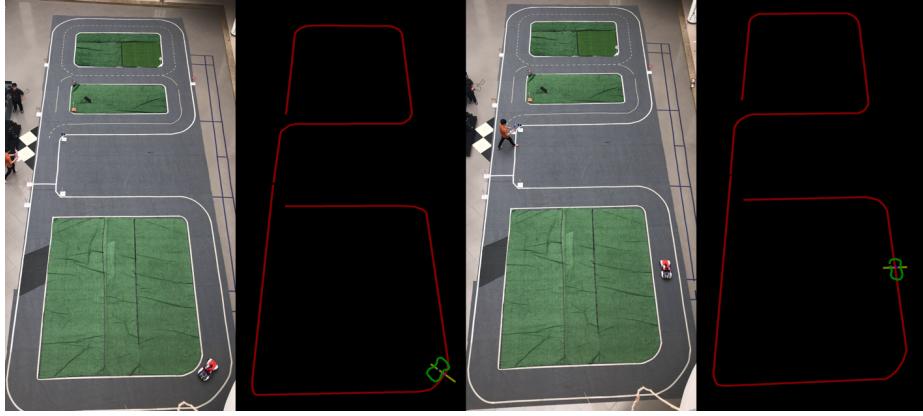


Fig 7. Centroid deviation experiment.

In the fifth experiment, Figure 8 toward checkpoint 5 denoting a vehicle challenge our car needs to stop when there is a stop sign. As required by rules following organizers, the two front wheels need to cross the starting line, where the starting line is a horizontal white line next to the stop sign. We designed the experiment to check the car's center of gravity variation to ensure that the vehicle's front wheels cross the starting line. We use the LIDAR sensor to get the actual distance value to the sign from the stop sign position, thereby determining the time it takes to get the car to the target point. When the angle returns to the LIDAR's sensor with a theta angle values 0 radians, the RPLiDARA2 sensor is located at the center of the vehicle that is the intersection of the longitudinal and latitude axis of the car. From the experimental data we collected, when the car stopped across the stop sign, the distance from the LIDAR to the top of the car was equal to distance  $d_1 = 25$  cm, and from here, we measured the length of the wheel to the starting line. In our condition, the wheel must pass the starting line with a distance  $d_2$  in the range of 5 - 10 cm, which is the desired figure that the car has to score correctly.



Fig 8. Stop deviation experiment.

In Table 5, we test the deviation of a stopping point with the investigation within ten samples (each round the car completed, we collected one piece). After completing the B-Shape route, the car has two positions to stop sign that depends on which traffic signs the car met. To sum up, the distance  $d_2$  with an average deviation is 7.5 cm in the range of 5 to 10 cm. Our car has been tested with accuracy achieved 90% for ten tries that 9 of 10 tries the vehicle cross the starting line correctly.

Table V. Deviation rate of the stop center point.

Round	Measure deviation
Straight sign-Turn	$6.8 \pm 0.7$ cm
Right sign-Turn	$8.4 \pm 0.5$ cm
Average	$7.50 \pm 0.55$ cm

Figure 9 demonstrates the car's log activities through a starting point to end-point after completing one round contains all problem sets. We denote the roadway marked red, and the line marked blue, and the vehicle is marked green. We log the green bounding box as the vehicle's positions. We collected 14 samples and plotted them all on one frame of the segmented map. The car meets the right traffic sign to go on the allowed part of the road. In the above case, the car follows the straight direction sign, so in the B-shaped area, the vehicle enters the right side of the street from the signpost to go straight up the top of the map. In left turns, right turns, and large steering angles, the vehicle can still move on its part of the road and has not crossed the white line beyond the organizers' permitted limits.

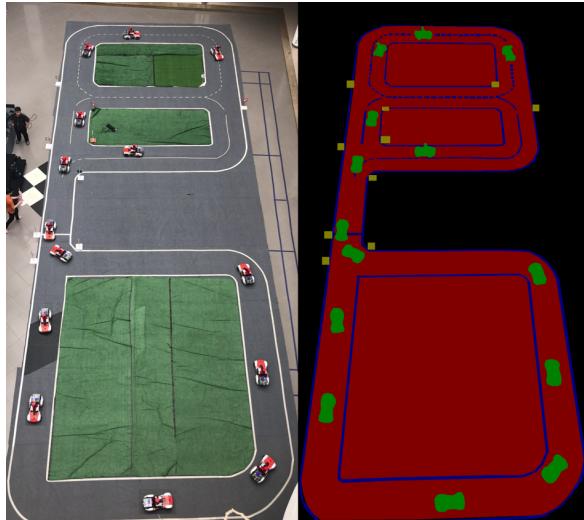


Fig 9. Plotting of the car position

In the last experiment, we perform the above experiments with three conditions: The maximum speed that the car can achieve with the average frame rate of 30, the initial battery condition is 100%, the physical condition of the vehicle is in good ergonomics. The self-driving vehicle has completed ten laps with  $925 \pm 10$  seconds. The car can handle it stably at eight rounds and only got 85% in the 9th and 10th rounds due to the battery condition when we measured below 70% of the original. These are external influencing factors, and we prepare scenarios for the above conditions. For example, when the battery state is only below 85%, we experiment that reduce the speed by 10% compared to the initial rate; the performance is approximately initial despite the round completion race's pace more slowly. Therefore, our system achieves stable performance in battery status, reaching 70% or more, and the average frame rate is 30 to 40 fps.

## 4 CONCLUSION

Our research implements an end-to-end solution for the vehicle to create the right angle to overcome challenges such as lane detection, traffic sign recognition, and dynamic obstacle detection as part of the organizers' problems. With the Jetson TX2 hardware, the car's real-time processing speed reached an average of 35 FPS. We won first place in the qualifying round at FPT University. The above rate ensures the vehicle can move stably and handle more complicated problems such as avoiding pedestrians and backing up into the parking area. Our proposed method is for robots that can solve problems with similar environmental conditions, and the hardware limit of computation is not large.

Our goal is to solve problems to get the fastest results in the competition. Our model needs more data to train for a background with more color and detail, ambiguous with the foreground. LiDAR sensors should have a stable environment to identify moving objects. If there are more noise variables in a car moving, the car can make the wrong decision in mixed driving tracks.

In the future, the solution we propose to integrate gathering ambient high perception data and making more complex decisions. That is the question need to have higher capacity calculation and hardware upgrade request. For real-time processing, it is necessary to speed up the processing speed to 20 FPS. Since our problems mostly deal with competition savers, high processing speed is essential for the car.

## REFERENCES

- [1] M. S. Ganeshmurthy and G. R. Suresh Path planning algorithm for autonomous mobile robot in dynamic environment. 2015 3rd International Conference on Signal Processing, Communication and Networking (ICSCN), Chennai, 2015, pp. 1-6, doi: 10.1109/ICSCN.2015.7219901.
- [2] J. W. Kang, S. J. Kim, M. J. Chung, H. Myung, J. H. Park and S. W. Bang Path Planning for Complete and Efficient Coverage Operation of Mobile Robots. 2007 International Conference on Mechatronics and Automation, Harbin, 2007, pp. 2126-2131, doi: 10.1109/ICMA.2007.4303880.
- [3] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks..In Proceedings of the European Conference on Computer Vision (ECCV), pages 818–833, 2014.
- [4] L. Jiao et al. A Survey of Deep Learning-Based Object Detection, in IEEE Access, vol. 7, pp. 128837-128868, 2019, doi: 10.1109/ACCESS.2019.2939201.
- [5] A. Vennelakanti, S. Shreya, R. Rajendran, D. Sarkar, D. Muddegowda and P. Hanagal. Traffic Sign Detection and Recognition using a CNN Ensemble.. 2019 IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, NV, USA, 2019, pp. 1-4, doi:10.1109/ICCE.2019.8662019.
- [6] Leibe, Bastian; Matas, Jiri; Sebe, Nicu; Welling, Max (2016). [Lecture Notes in Computer Science] Computer Vision – ECCV 2016 Volume 9905 || SSD: Single Shot MultiBox Detector., 10.1007/978-3-319-46448-0(Chapter 2), 21–37. doi:10.1007/978-3-319-46448-0\_2.
- [7] Redmon, Joseph; Divvala, Santosh; Girshick, Ross; Farhadi, Ali (2016). [IEEE 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) - Las Vegas, NV, USA (2016.6.27-2016.6.30)] 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) - You Only Look Once: Unified, Real-Time Object Detection., (), 779–788. doi:10.1109/CVPR.2016.91.
- [8] Y. Peng; D. Qu; Y. Zhong; S. Xie; J. Luo; J. Gu (2015). The obstacle detection and obstacle avoidance algorithm based on 2-D lidar., (), -. doi:10.1109/IClnfa.2015.7279550.
- [9] Song, Hyunwoo; Lee, Kwangkook; Kim, Dong Hun (2018). [IEEE 2018 Joint 10th International Conference on Soft Computing and Intelligent Systems (SCIS) and 19th International Symposium on Advanced Intelligent Systems (ISIS) - Toyama, Japan (2018.12.5-2018.12.8)] 2018 Joint 10th International Conference on Soft Computing and Intelligent Systems (SCIS) and 19th International Symposium on Advanced Intelligent Systems (ISIS) - Obstacle Avoidance System with LiDAR Sensor Based Fuzzy Control for an Autonomous Unmanned Ship., (), 718–722. doi:10.1109/SCIS-ISIS.2018.00119.
- [10] Madhavan, T.R.; Adharsh, M. (2019). [IEEE 2019 International Conference on Computer Communication and Informatics (ICCCI) - Coimbatore, Tamil Nadu, India (2019.1.23-2019.1.25)] 2019 International Conference on Computer Communication and Informatics (ICCCI) - Obstacle Detection and Obstacle Avoidance Algorithm based on 2-D RPLIDAR., (), 1-4. doi:10.1109/ICCCI.2019.8821803.
- [11] Hu, Jia; Niu, Yifeng; Wang, Zhichao (2017). [IEEE 2017 Chinese Automation Congress (CAC) - Jinan (2017.10.20-2017.10.22)] 2017 Chinese Automation Congress (CAC) - Obstacle avoidance methods for rotor UAVs using RealSense camera., (), 7151–7155. doi:10.1109/CAC.2017.8244068.
- [12] Mannor, S., Jin, X., Han, J., Jin, X., Han, J., Jin, X., ... Zhang, X. (2011). K-Means Clustering. Encyclopedia of Machine Learning, 563–564. doi:10.1007/978-0-387-30164-8\_425.

- [13] Wei, Xianwen; Zhang, Zhaojin; Chai, Zongjun; Feng, Wei (2018). [IEEE 2018 International Conference on Intelligent Robotic and Control Engineering (IRCE) - Lanzhou, China (2018.8.24-2018.8.27)] 2018 IEEE International Conference on Intelligent Robotic and Control Engineering (IRCE) - Research on Lane Detection and Tracking Algorithm Based on Improved Hough Transform., (), 275–279. doi:10.1109/IRCE.2018.8492932.
- [14] Yang, Xiaofei; Li, Xutao; Ye, Yunming; Lau, Raymond Y. K.; Zhang, Xiaofeng; Huang, Xiaohui (2019). Road Detection and Centerline Extraction Via Deep Recurrent Convolutional Neural Network U-Net. *IEEE Transactions on Geoscience and Remote Sensing*, (), 1–12. doi:10.1109/TGRS.2019.2912301.
- [15] Vosselman, George; Coenen, Maximilian; Rottensteiner, Franz (2017). Contextual segment-based classification of airborne laser scanner data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 128(), 354–371. doi:10.1016/j.isprsjprs.2017.03.010.
- [16] Zhang, C., Wang, L., & Yang, R. (2010). Semantic Segmentation of Urban Scenes Using Dense Depth Maps. *Lecture Notes in Computer Science*, 708–721. doi:10.1007/978-3-642-15561-1\_51.
- [17] Zhao, Hengshuang; Shi, Jianping; Qi, Xiaojuan; Wang, Xiaogang; Jia, Jiaya (2017). [IEEE 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) - Honolulu, HI (2017.7.21-2017.7.26)] 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) - Pyramid Scene Parsing Network., (), 6230–6239. doi:10.1109/cvpr.2017.660.
- [18] Fleet, David; Pajdla, Tomas; Schiele, Bernt; Tuytelaars, Tinne (2014). [Lecture Notes in Computer Science] Computer Vision – ECCV 2014 Volume 8689 || Visualizing and Understanding Convolutional Networks., 10.1007/978-3-319-10590-1(Chapter 53), 818–833. doi:10.1007/978-3-319-10590-1\_53.
- [19] Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, 234–241. doi:10.1007/978-3-319-24574-4\_28.
- [20] Zhao, Hengshuang; Shi, Jianping; Qi, Xiaojuan; Wang, Xiaogang; Jia, Jiaya (2017). [IEEE 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) - Honolulu, HI (2017.7.21-2017.7.26)] 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) - Pyramid Scene Parsing Network., (), 6230–6239. doi:10.1109/cvpr.2017.660.
- [21] He, Kaiming; Zhang, Xiangyu; Ren, Shaoqing; Sun, Jian (2016). [IEEE 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) - Las Vegas, NV, USA (2016.6.27-2016.6.30)] 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) - Deep Residual Learning for Image Recognition., (), 770–778. doi:10.1109/CVPR.2016.90.
- [22] Marin-Plaza, Pablo; Hussein, Ahmed; Martin, David; Escalera, Arturo de la (2018). Global and Local Path Planning Study in a ROS-Based Research Platform for Autonomous Vehicles. *Journal of Advanced Transportation*, 2018(), 1–10. doi:10.1155/2018/6392697.
- [23] Lee, Unghui; Yoon, Sangyol; Shim, HyunChul; Vasseur, Pascal; Demonceaux, Cedric (2014). [IEEE 2014 IEEE 4th Annual International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER) - Hong Kong, Hong Kong (2014.6.4-2014.6.7)] The 4th Annual IEEE International Conference on Cyber Technology in Automation, Control and Intelligent - Local path planning in a complex environment for self-driving car., (), 445–450. doi:10.1109/cyber.2014.6917505.
- [24] Huijuan Wang, Yuan Yu, Quanbo Yuan, (2011). [IEEE 2011 Second International Conference on Mechanic Automation and Control Engineering (MACE) - Inner Mongolia, China (2011.07.15-2011.07.17)] 2011 Second International Conference on Mechanic Automation and Control Engineering - Application of Dijkstra algorithm in robot path-planning., (), 1067–1069. doi:10.1109/mace.2011.5987118.
- [25] Zeng, Zheng; Sun, Wei; Wu, Wei; Xue, Min; Qian, Lin (2019). [IEEE 2019 IEEE 15th International Conference on Control and Automation (ICCA) - Edinburgh, United Kingdom (2019.7.16-2019.7.19)] 2019 IEEE 15th International Conference on Control and Automation (ICCA) - An Efficient Path Planning Algorithm for Mobile Robots., (), 487–493. doi:10.1109/ICCA.2019.8899975.
- [26] Bokovoy, Andrey; Muravyev, Kirill; Yakovlev, Konstantin (2019). [IEEE 2019 European Conference on Mobile Robots (ECMR) - Prague, Czech Republic (2019.9.4-2019.9.6)] 2019 European Conference on Mobile Robots (ECMR) - Real-time Vision-based Depth Reconstruction with NVidia Jetson., (), 1–6. doi:10.1109/ecmr.2019.8870936.
- [27] (2018). An Empirical Evaluation of Ten Depth Cameras: Bias, Precision, Lateral Noise, Different Lighting Conditions and Materials, and Multiple Sensor Setups in Indoor Environments. *IEEE Robotics & Automation Magazine*, (), 1–1. doi:10.1109/MRA.2018.2852795.
- [28] Zeglazi, Oussama; Rziza, Mohammed; Amine, Aouatifi (2017). [IEEE 2017 International Conference on Wireless Networks and Mobile Communications (WINCOM) - Rabat, Morocco (2017.11.1-2017.11.4)] 2017 International Conference on Wireless Networks and Mobile Communications (WINCOM) - An enhanced cross-scale adaptive cost aggregation for stereo matching., (), 1–5. doi:10.1109/wincom.2017.8238214.
- [29] H. Hirschmuller, "Stereo Processing by Semiglobal Matching and Mutual Information," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 2, pp. 328-341, Feb. 2008, doi: 10.1109/TPAMI.2007.1166.
- [30] Q. Wang; S. Xu; H. Xu (2014). A Fuzzy Control Based Self-Optimizing PID Model for Autonomous Car Following on Highway., (), -. doi:10.1109/WCSN.2014.87.

**Authors' background / The form itself will not be published, Please Delete it on your final papers**

Your Name	Title*	Research Field	Personal website
-----------	--------	----------------	------------------

VU THANH DAT	Engineer	Computer Science	
PHAN QUANG HUY	Engineer	Computer Science	
NGUYEN DINH TRA	Engineer	Computer Science	
TRAN HAI AN	Engineer	Computer Science	
Ngo Tung Son	Senior lecture	Computer Science	<a href="http://science.fpt.edu.vn/?human=ths-ngo-tung-son">http://science.fpt.edu.vn/?human=ths-ngo-tung-son</a>

\*Title can be chosen from: master student, Phd candidate, assistant professor, lecture, senior lecture, associate professor, full professor, Engineer, Senior Engineer, etc.