

Effective Parameters to Design an Automatic Parking System

Hossein Borghei Razavi ^{1,*}, Arash Sherafati ²¹ Department of Computer and Electrical Engineering, Tehran North Branch, Islamic Azad University, Tehran, Iran² Computer Engineering Department, Shomal University, Amol, Mazandaran, Iran

*Correspondence: Hossein Borghei Razavi (bmesgh1@yahoo.com)

Abstract: The automated parking system is an extensive branch of smart transport systems. The smartness of such systems is determined by different parameters such as parking maneuver planning. Coding this control system includes vehicle parking and understanding the environment. A high-quality classification mask has been used on each sample to analyze the automated vehicle parking parameters. Mask region-based convolutional neural networks (R-CNN) was taught using a small computational workload titled faster R-CNN that operates in five frames per second. In this paper, the rapidly-exploring random tree (RRT) method was used for routing the parking space and a nonlinear model predictive control (NMPC) controller was added to develop this system. We add the line detection algorithm commands to the mask R-CNN algorithm. The results can be useful to design a secure automatic parking system as well as a powerful perception system.

Keywords: Vehicular Automation, Control System, Automated Driving Technology, Kinematic Analysis, Park SYSTEM

How to cite this paper: Borghei Razavi, H., & Sherafati, A. (2022). Effective Parameters to Design an Automatic Parking System. *Universal Journal of Computer Sciences and Communications*, 1(1), 17–34. Retrieved from <https://www.scipublications.com/journal/index.php/ujcsc/article/view/550>

Received: November 5, 2022

Accepted: December 13, 2022

Published: December 15, 2022



Copyright: © 2022 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The progress of technology, especially the fast development of the internet, communications, and artificial intelligence during the last few years, has to lead to the era of smart vehicles [1]. Yet, the practical use of such smart vehicles requires a long period. Even with the vast developments in the field of smart vehicles, there are still some problems that require attention. The driving environment is complex and diverse which creates many different decision-making challenges [2]. Some of the studies have used biological tools to solve these difficulties [3]. Furthermore, the vehicular automation service is designed based on a series of fixed rules that do not consider personalized driving styles. This fact seriously stops the application of driving with smart vehicles. Usually, automated driving systems include the comprehension, decision-making, and control stages [2]. The configuration of this system architecture is compatible with human driving behavior. The smart vehicle gathers the sensor information using different sensors, fuses this different sensor information, creates the multi-environment information, and transfers them to decision-making planning layers [4]. Then, the needed decisions are made and the driving information is created based on the limitations and rules determined by the system. Finally, the control module performs these orders for achieving automated driving. Some statistical and simulation methods that were used in biological phenomena are also effective in gaining this goal [5-7]. In the meanwhile, the decision-making connection allows smart vehicles to reach an environmental understanding of driving as one of the main aspects of automated driving technology. Automated vehicles are also known as driverless vehicles, computer-driven vehicles, and mobile wheel robots. They are a type of smart vehicle that can be implemented without a driver using computers. The purpose of this study is to propose an automated learning method for personal decision-making

about driving. First, the driving data are gathered and analyzed from different drivers to set learning goals. Then, these elements are personalized and introduced.

2. Measurements

One of the most important parts of driverless cars is the measurements and processing to reach proper recognition of the park, dynamic objects (e.g. children, pets or animals, and bicycles), and static obstacles around the car. Figure 1 shows the stages for receiving data/information about driverless cars. There are two stages performed to accurately recognize the environment and objects. These stages are done after receiving information from sensors.

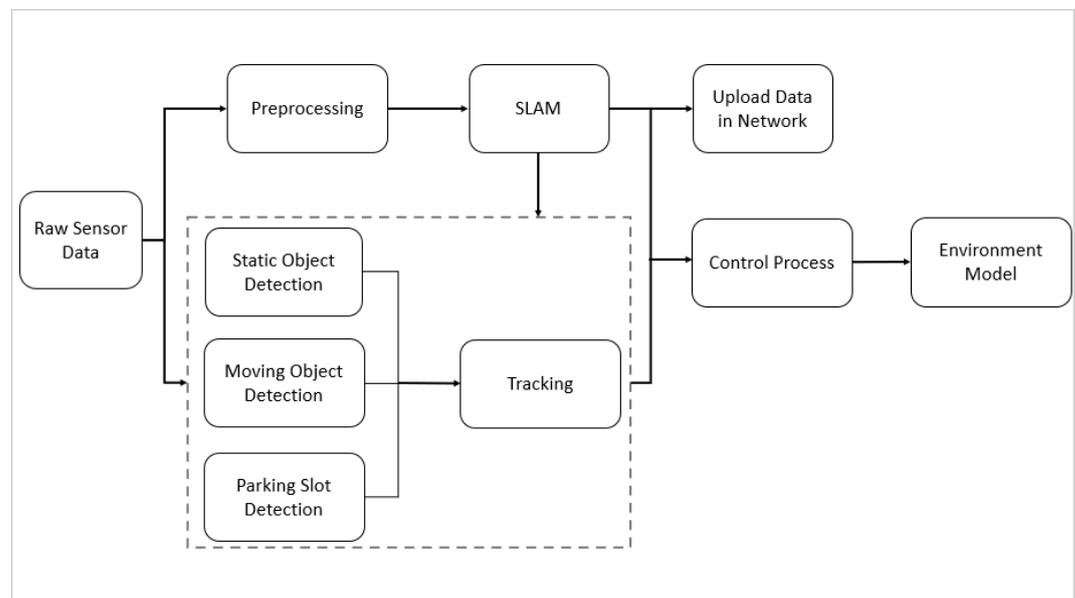


Figure 1. Procedure of receiving data, detecting and controlling system.

A pre-processing is performed to recognize outlier data from the correct ones. These data are used for simultaneous localization and mapping (SLAM) in such a way that it produces a map of the region and according to whole measurements of sensors, localizes the vehicle in the map simultaneously; and then loaded to improve maps and data/information of that region in the network (local or worldwide). They are used to recognize and track dynamic/static objects as well as the suitable place for the car to be parked; and eventually, our perception system estimates their coming behavior.

Incomplete information management is one of the important requirements of the perception system. The reasons for being created incomplete information can be different factors such as the presence of a defect in sensors, not being calibrated, presence of the noise around the environment of the car (e.g. changes in weather conditions; the movement of objects), as well as internal factors (e.g. fluctuations; failure of connections due to loosening and impact, or corrosion due to using low-quality materials, burning, and fluid penetration) and the presence of obstacles which cause the sensor not to activate truly (e.g. the presence of dirt or physical obstruction on the camera lens; placing on the environment which deflect the wave).

According to the advanced approaches, we assume that the SLAM stage is a solved problem and focuses on the recognition measurement, dynamic object routing, suitable park place, and static objects (for example, panels/signs, lights, and traffic signs). In this study, we assume that the information of sensors is completely accurate and without errors due to the simulated parking maneuver of the measurement part of the automated

parking in MATLAB software; and in the perception part, have investigated the reception of information by cameras. (The videos are recorded by the cameras).

3. Kinematic analysis of the parking mechanism

The main process of vehicle parking systems is divided into three major steps.

1. determining the parking location,
2. designing the path towards the parking location.
3. controlling the vehicle on the designed path toward the goal.

Differential equations are used to describe the kinematics of driving vehicles forward [8].

$$\begin{cases} \dot{x} = \frac{dx(t)}{dt} = v(t) \cdot \cos \theta(t) \\ \dot{y} = \frac{dy(t)}{dt} = v(t) \cdot \sin \theta(t) \\ \dot{v} = \frac{dv(t)}{dt} = \alpha(t) \\ \dot{\theta} = \frac{d\theta(t)}{dt} = \frac{v(t) \cdot \tan \varphi(t)}{l} \\ \dot{\varphi} = \frac{d\varphi(t)}{dt} = \omega(t) \end{cases} \quad (1)$$

$t \in [t_0, t_f]$ refers to time, t_f shows parking process end time, (x,y) are the middle point along the rear wheel axis, θ is the movement angle relative to the x-axis, v is the (y,x) point velocity (as defined in Figure 2a), a is acceleration, φ is the front wheel steering degree, ω is the angular velocity, and l is the distance between the two axes. n (length of the front bulge), m (length of the rear bulge), and $2x$ (vehicle width) are the other geometric parameters. $\alpha(t)$ and $\omega(t)$ are selected as the control variable in the motor system.

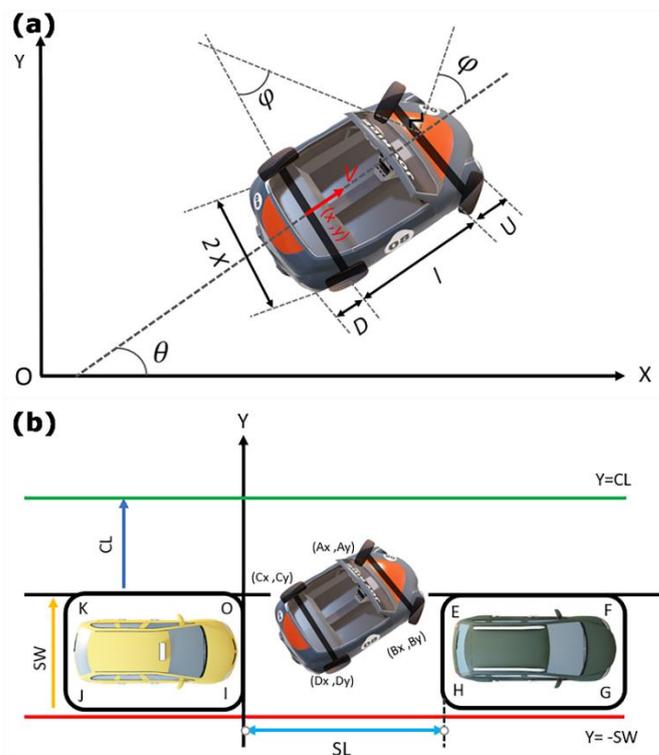


Figure 2. (a) Parametric symbols related to vehicle size and kinematics. (b) Parallel parking scenario diagrams. In which $SL > 0$ indicates the parking space length, $SW > 0$ indicates the parking space width and $CL > 0$ is the line width. The $y=CL$ and $y=-SW$ functions determine the borders between acceptable parking spaces.

4. Physical and environmental limitations

4.1. Physical limitations

The mechanical and physical limitations of the state variables in control must be considered besides the vehicle kinematics that was described in the previous section using differential equations. These limitations are described in detail in the following.

The reasons behind imposing limitations on $\alpha(t)$, $v(t)$ and $\varphi(t)$ are clear. In previous literature, $\omega(t)$ was limited extensively to planning continuous curvature paths. This is because the momentary curvature $K(t) = \frac{\sin \varphi(t)}{l}$ and its derivative $\dot{K} = \frac{dK(t)}{dt} = \frac{\omega(t) \cos \varphi(t)}{l}$ must be limited to prevent nonlinear paths. Usually, nonlinear paths are not recommended due to adverse rubber wear.

$\varphi(t)$ is mechanically limited, if there are no limitations on implementing it $\omega(t)$, \dot{K} and then the continuous curvature attribute cannot be guaranteed. Thus far, the mechanical and physical limitations of vehicles have been formulated. Furthermore, we have the collision-avoiding rule that must be obeyed during the movements of an independent vehicle in the environment [8]. These limitations are described in detail in the following [9].

$$\begin{cases} |\alpha(t)| \leq \alpha_{max} \\ |v(t)| \leq v_{max} \\ |\varphi(t)| \leq \varphi_{max} \\ |\omega(t)| \leq \omega_{max} \end{cases}, \forall t \in [t_0, t_f] \quad (2)$$

4.2. Environmental Limitation

This subsection is about avoiding collisions with the environment based on precise geometry. Unlike previous studies that assume such obstacles create ideal gaps, we only need the parked vehicle to avoid collision with other parked vehicles during the maneuver. By considering the vehicles as rectangles, the first section introduces the method of the precise method of defining a rectangle outside of another one. Then, the collision-avoiding limitations are formulated [10-12]. If the parking space is between two properly parked cars (Figure 2b), it is expected that the parked vehicle finally enters the parking lot.

$$\begin{cases} -SW \leq A_y(t) \leq CL \\ -SW \leq B_y(t) \leq CL \\ -SW \leq C_y(t) \leq CL \\ -SW \leq D_y(t) \leq CL \end{cases}, \text{for } t \in [t_0, t_f] \quad (3)$$

where

$$\begin{cases} A_x = x + (U + l) \cdot \cos \theta - b \cdot \sin \theta \\ A_y = y + (U + l) \cdot \sin \theta - b \cdot \cos \theta \\ B_x = x + (U + l) \cdot \cos \theta - b \cdot \sin \theta \\ B_y = y + (U + l) \cdot \sin \theta - b \cdot \cos \theta \\ C_x = x - D \cdot \cos \theta - b \cdot \sin \theta \\ C_y = y - D \cdot \sin \theta - b \cdot \cos \theta \\ D_x = x - D \cdot \cos \theta - b \cdot \sin \theta \\ D_y = y - D \cdot \sin \theta - b \cdot \cos \theta \end{cases} \quad (4)$$

The collision-avoiding limitations can be precisely defined using the "Triangle Area Criterion" as follows: It must be outside the EFGH area, and the four corners of EFGH must be outside the ABCD area (Figure 2b).

The parallel parking process starts with a determined configuration with all mode variables as $t=0$. When $t = t_f$, the vehicle must stop in the parking space, this shows that:

$$\left\{ \begin{array}{l} v(t_f) = 0 \quad \text{and} \\ A_y(t_f) \leq 0 \\ B_y(t_f) \leq 0 \\ C_y(t_f) \leq 0 \\ D_y(t_f) \leq 0 \\ 0 \leq A_x(t_f) \leq SL \\ 0 \leq B_x(t_f) \leq SL \\ 0 \leq C_x(t_f) \leq SL \\ 0 \leq D_x(t_f) \leq SL \end{array} \right. \quad (5)$$

5. Vehicle design and simulation

We need a modeled vehicle for analyzing effective parameters in designing a control system for automated driving. This vehicle must have the standards of a real vehicle. These standards include implementing different internal vehicle systems such as power distribution systems, suspension and brake systems, and command and steering systems; while being able to transfer internal and external vehicular information (from different sensors and cameras) to the operator. The operator will be able to edit the proper program for correct actions and timely decision-making, test this program in the simulated environment and then test the program once more on a real vehicle if the program was able to properly perform with low-risk. With inspiring the biological phenomena, many variables involve the ability of the operator [13-16]. The ego vehicle model was used for analyzing effective parameters in designing a control system for automated driving. MATLAB 2020b was used as the simulation environment (Figure 3).

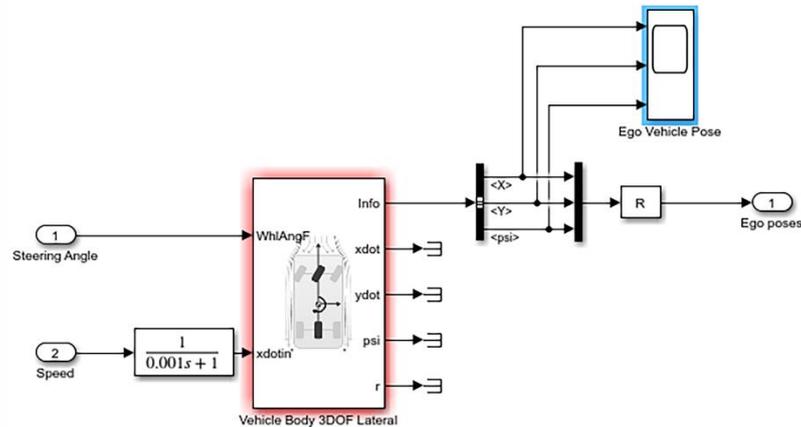


Figure 3. Ego vehicle model.

6. Routing

Routing is one of the most important parameters in the correct parking of a car. It determines how a car should locate in the parking place. One of the most important methods raised in path planning (motion planning) is the use of random search. In this pattern, the rhythms, have the features such as replying with less time, lack of a need to define the space accurately, and capability to employ the different situations of robots, causing these algorithms to be the most popular methods raised in path planning. One of the random search techniques is the rapidly-exploring random tree (RRT) method.

6.1. RRT method

This algorithm solves the path planning problem using the random search of the configuration space. In this method, the root is selected randomly with a uniform distribution. If the selected point belongs to the restricted space, another point is selected. A new point named X_{new} which is as far away from the root as the vehicle is chosen by selecting the point from the open space along the connection line between this point and the tree root. If the X_{new} can be connected to the root with a line segment that totally belongs to the open space, it is added to the tree as a new node [17].

6.2. RRTS method

The rapidly-exploring random tree star (RRTS) method is sampled from the environment. But, the difference between this method and RRT is in this random exploring tree. After a completely random environmental sampling, if this new node is valid, it will be added to the tree. Now by considering the disc with a new center, the tree nodes that have a higher connection cost to this new node will have their paths corrected. The paths obtained from this method (Figure 4a) will be shorter than other methods [18].

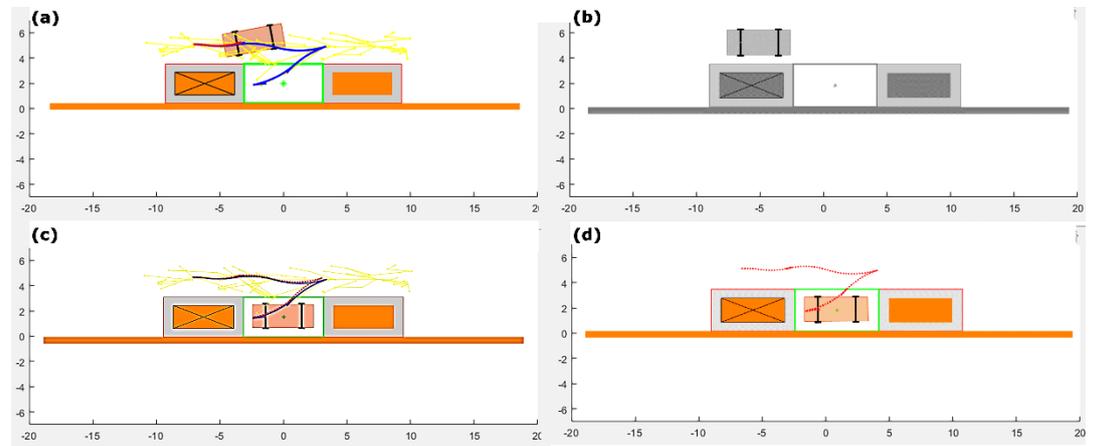


Figure 4. (a) Routing using rapidly-exploring random tree. (b) Designing the parking space and ego vehicle. (c) Automated parking process with the rapidly-exploring random tree method with the nonlinear model predictive control controller. (d) Automated parking simulation process with the nonlinear model predictive control controller.

7. Designing a nonlinear model predictive control

The nonlinear model predictive control (NMPC) samples each interval by controlling an open-loop with a limited horizons solution and determines a controller action for controlling nonlinear system feedback. A general form of nonlinear systems can be shown as:

$$\begin{cases} \dot{q}(t) = f(q(t), u(t)), q(0) = q_0, \\ \text{subject to } u(t) \in U, q(t) \in Q, t \in [0, \infty] \end{cases} \quad (6)$$

In which, $u(t) \in R^M$, $q(t) \in R^N$ are respectively the mode and input graphs. The N and M sub-versions are the dimensions of these mode and input graphs. The Q and U sets are the abbreviations of possible system modes and inputs.

The implemented inputs of each system are presented by a control problem solution with optimizing an open-loop with limited horizons, which are sampled while being solved.

$$\min_{u(\tau)} j(q(\tau), u(\tau)) \quad (7)$$

Subject to

$$\begin{cases} \dot{q}(\tau) = f(q(\tau), u(\tau)), q(0) = q_0, \\ q(\tau) \in Q, u(\tau) \in U, \forall \tau \in [t, t + T] \end{cases} \quad (8)$$

where T is the predicted horizon and $j(q(\tau), u(\tau))$ shows the goal function defined as follows:

$$j(q(t), u(t)) = \int_t^{t+T} F(q(\tau), u(\tau)) d\tau + E(q(t + T)) \quad (9)$$

which F is the cost function that defines optimized performance. E is the final cost. $q(\cdot)$ is the predicted mode graph which is created by the $u(\cdot): [t, t + T] \rightarrow U$ input signal under the q_0 initial conditions. In general, q_0 is the measurement of the current mode sensor in the vehicle while executing the real system. The inputs implemented on the sequence robot were able to present optimized solutions that were equal to $\delta: u = u^*(\cdot, q(\delta))$ in each moment. Then, the nominal closed-loop system is defined as follows.

$$\dot{q}(t) = f(q(t), u = u^*(\cdot, q(\delta))) \quad (10)$$

The aforementioned NMPC strategy was used for forcing the vehicle to follow the designed path, while the following quadratic cost functions will be replaced in Equation (9).

$$F(q, u) = e_q(\tau)^T \tilde{Q} e_q(\tau) + u(\tau)^T \tilde{R} u(\tau) \quad (11)$$

$$E(q) = e_q(t + T)^T \tilde{Q}_N(q(t + T)) e_q(t + T) \quad (12)$$

where $\bar{q}(\tau)$ from $e_q(\tau) = q(\tau) - \bar{q}(\tau)$ is the reference modes of the vehicle shown as $(\tau) = [x_d(\tau) y_d(\tau) \varphi_d(\tau) u_d(\tau) v_d(\tau) r_d(\tau)]^T$.

\tilde{Q} and \tilde{R} are determined weight matrixes that punish deviations from the intended values.

\tilde{Q}_N is the terminal punishment matrix for improving the NMPC algorithm stability. We concluded that the proposed algorithm can be stabilized with rather long horizons and configured \tilde{Q} , \tilde{R} and T .

We implement the NMPC algorithm in (7) and (8) in real-time to solve the effective control problem based on the dynamic model of the vehicle. This helps us achieve an exact optimized path tracing for the vehicle. The NMPC was designed in the MATLAB simulated environment (Figure 5).

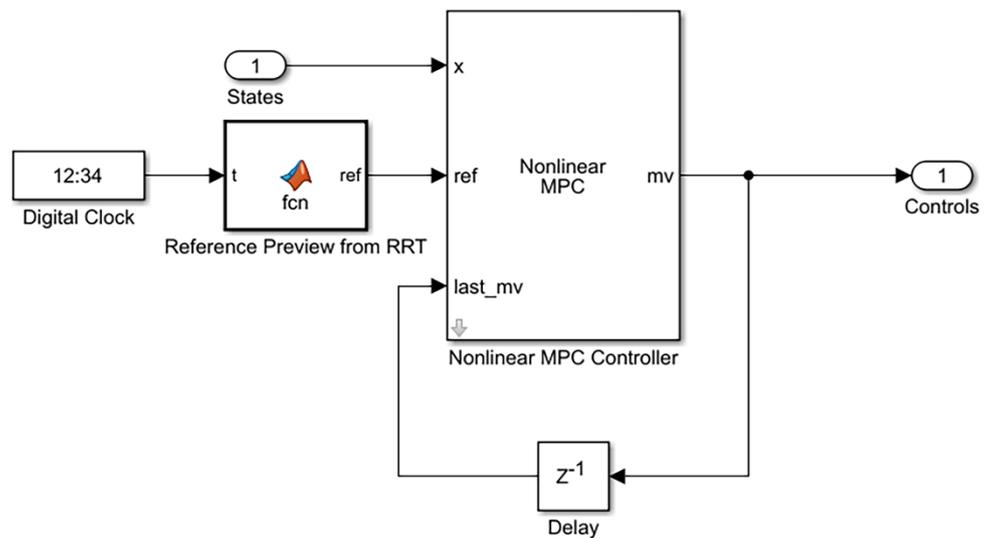


Figure 5. Nonlinear model predictive control diagram block.

8. Effects of environmental parameters

Automated vehicles have provided many interesting and unique opportunities in different contexts such as the technical fields. This study analyzes the problem of finding the road line as one of the major problems of automated vehicles using computer vision. They continuously push our society forward with deep learning algorithms and create new opportunities in the movement area. An autonomous vehicle can go anywhere a traditional vehicle can and do anything an experienced human driver can do, but the correct training is essential. Recognizing lines is the first stage of teaching independent driving to a vehicle (Figure 6). The line recognition method will be analyzed using film (Figure 7).

- I. *Recording Video Files:* Each frame of these recordings will be decoded as they are being filmed (i.e. they will be turned into a sequence of pictures).
- II. *Gray Scaling the Pictures:* The videos are recorded in RGB format. These RGB videos are turned into grayscale because processing a single-color channel is easier than processing three.
- III. *Noise Reduction:* Noise can create false edges; therefore, these pictures must be smoothed before containing. A Gaussian filter was used for this.
- IV. *Canny Edge Detector:* Calculates the slope in all directions of matte images while following the edges through big obesity changes.

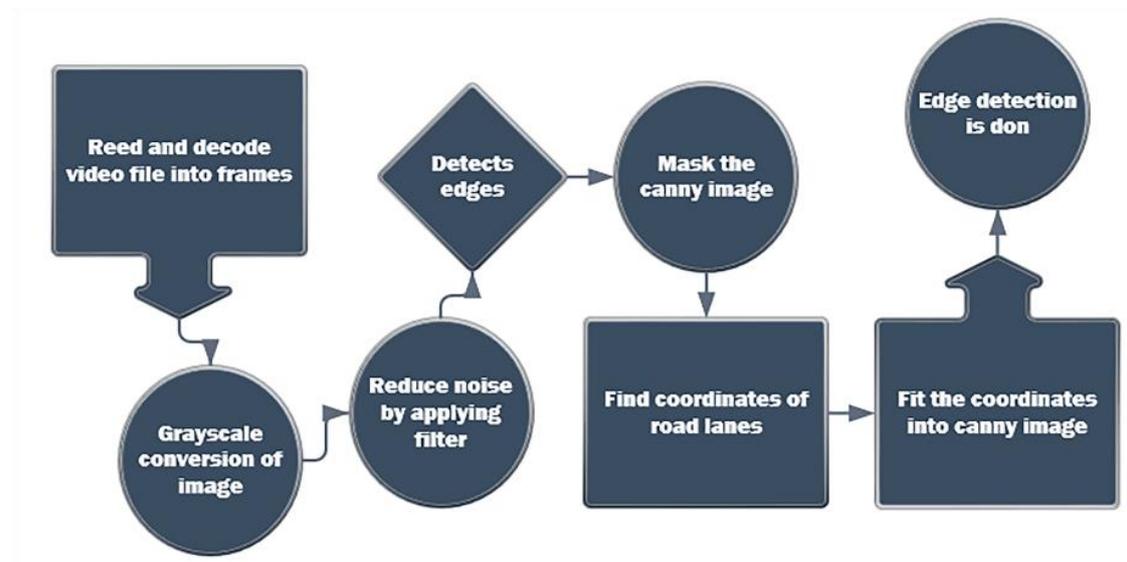


Figure 6. Stages of line detection.

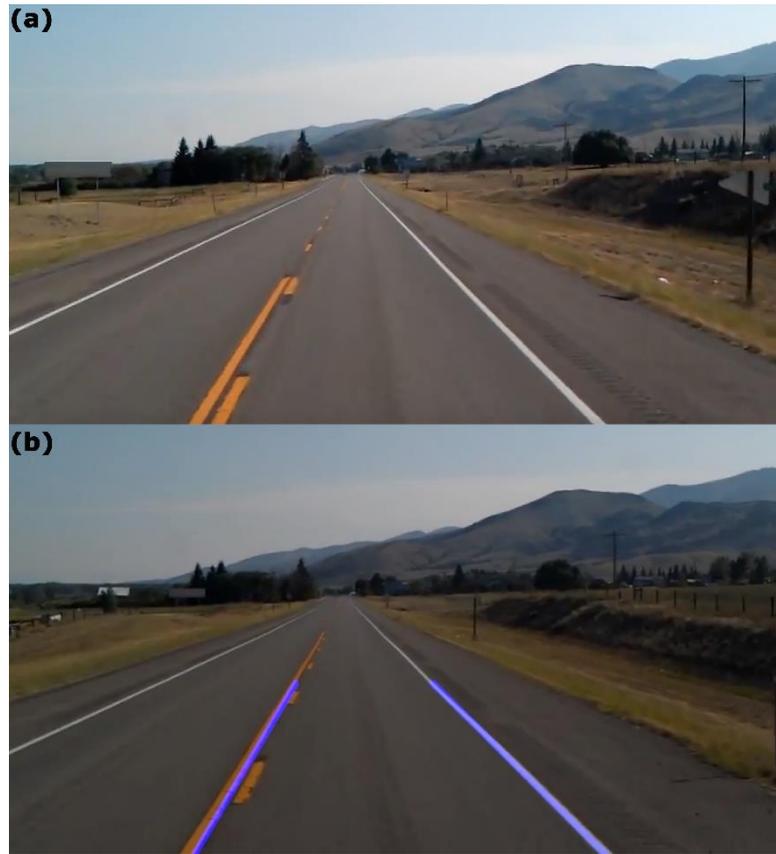


Figure 7. (a) Input film. (b) Output of the processed film.

The purpose of the Canny algorithm is to provide three main criteria including low error rate; proper detection of the available edges, proper localization; the distance between the pixels of the recognized edges and the real edges must be minimized, and minimum answers; only one detector answer for each edge.

9. Calculations

1. The noises must be filtered: The Gaussian filter was used for this. An example of a size 5 Gaussian core is shown in the following.

$$K = \frac{1}{159} \begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix} \quad (13)$$

2. The picture slope intensity was determined: The same procedure is followed for sobel [19].
 - a. A pair of convulsion masks (in direction of x and y) was used.

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} \quad (14)$$

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix} \quad (15)$$

- b. Power and slope direction were determined.

$$G = \sqrt{G_x^2 + G_y^2} \quad (16)$$

$$\theta = \arctan\left(\frac{G_y}{G_x}\right) \quad (17)$$

The direction was rounded out to one of the four possible angles (i.e. 0, 45, 90, or 135 degrees).

1. The non-maximum repression was implemented. This approach will remove any pixel that is not part of an edge. Therefore, only the thin lines (candidate edges) will remain.
2. Hysteresis of the last stage. Canny uses the (upper and lower) thresholds.
 - a. If the pixel slope is higher than the upper threshold, it will be accepted as an edge.
 - b. If the pixel slope is less than the lower threshold, it will be rejected.
 - c. If the pixel slope is between these two thresholds, it will only be accepted if it has a connection with a higher threshold pixel.

9.1. Area zone

This stage only considers the area covered by the road line. Here a mask is created that acts as our road image. Furthermore, the bitwise AND process is performed between each pixel of our console picture and this mask. Finally, the canny image is covered and the intended area is shown using the traced polygon converter mask.

9.2. Hough line transform

The hough line transform is a figure transformation used for determining straight lines. Here, the probable line transformation is used to determine the output as the ending of those recognized lines

9.3. Recognizing moving objects and traffic signs

The vehicle detection and tracking model is one of the most important parts of implementing autonomous driving systems. Vehicle recognition and tracking systems must be fast and accurate enough to be used in real work applications such as traffic control and management, Autonomous Automobiles, and others.

10. Item detection

A human can recognize the different elements of a picture (Figure 4b), their positions, and their connection with one short glance. Such actions are rather easy for humans and the accurate and fast vision of humans can easily perform much more complex tasks such as driving. Of course, recognizing and positioning the object surrounding the vehicle is one of the significant parts of driving that humans are highly skilled in performing. Now, if we create fast and accurate object recognition and positioning algorithms, we could create autonomous automobiles without the need for special sensors. Recognizing and positioning objects is one of the old and important research fields in computer vision. In computer vision, recognizing and positioning an object in an image is simply known as object detection. Here the "Object Detection" term is used instead of item recognition and positioning. This section is about one fast and accurate item detection system called You only look once (YOLO). Many different developed methods have been introduced for this algorithm; here, we want to analyze one developed YOLO algorithm for evaluating parameters that affect item detection in the autonomous automobile's controller system.

11. Mask R-CNN

Mask region-based convolutional neural networks (R-CNN) is an item detection method based on the YOLO algorithm. This is a visual concept method that presents flexibility and stability alongside a quick training and inference time. This sample classification has different challenges such as the need for accurate real-time classification of each sample while correctly recognizing all available objects in the image. Mask R-CNN is the method analyzed in this study. We developed the Faster R-CNN method by adding a branch for predicting mask classification in the regions of interest (ROI) parallel to other available branches for the bounding-box classification and regression. The mask branch is a small fully complex network (FCN) that implements a classification prediction mask using the pixel-to-pixel method in each region of interest. The faster R-CNN framework (that facilitates a wide range of flexible architectural designs) can be easily trained and executed in the mask R-CNN. Furthermore, it only adds a small computational workload threshold to the mask branch to facilitate a fast system and speedy testing. In reality, the mask R-CNN is a visual development of Faster R-CNN, but the correct creation of the branch mask is vital for good results. Most importantly, Faster R-CNN is not designed for a pixel-to-pixel alignment of network inputs and outputs. This is clear in the ROI Pool method which implements the main process in the samples by performing a spatial quantification and extracting the features [20]. The authors have proposed an open and unvalued layer titled ROIAlign to solve the noncompliance problem that confidently saves the exact spatial distances. Although the ROIAlign might seem like a minor change, it has an impressive effect on mask accuracy and can increase it by 10 to 50%, which shows a higher range in more accurate localization criteria. The execution of mask R-CNN requires multiple stages which are:

11.1. Image improvement

We use a new approach to remove the noise and increase the contrast in a united context using the fog-based algorithm [21] and fuzzy coverage algorithm [22]. This approach consists of two stages. The first stage is related to removing the superpixel-based adaptive noise, and the second increasing adaptive brightness contrast. Before the increase, in contrast, denoization is done, so that the noise has been removed by the contrast increase prior to amplification [23].

11.2. Network architecture

The mask R-CNN method was sampled with multiple network architectures to show the generality of this intended approach. For clarity (I) the convolutional backbone architecture was used for extracting features of the whole image while (II) the network head for determining the bounding box (classification and regression) and predicting the mask in each region of interest.

11.3. Training

Similar to the faster R-CNN method, a region of interest is considered positive if the Intersection over union (IoU) overlaps points with the ground-truth box are at least 0.5, or otherwise, it will be negative. The mask loss L_{mask} is only defined in positive ROIs. The mask is after recognizing the intersection point between an RoI and its related ground-truth mask. The image size will change to a scale of 800 pixels (shorter edge) [23].

11.4. Inference

During the testing, 300 is the suggested value for the C4 backbone, and 1000 is suggested for feature pyramid networks (similar to [24]). The box prediction branch is executed based on these suggestions with maximum suppression. Then the box mask detection branch is set to the highest value of 100. Although this is different from the parallel computing used during the tests, this will improve the inference speed and accuracy (by

using fewer ROIs that are more precise). The mask branch can predict the mask of each ROI but, we only use the K mask. K is the same predicted class by the classification branch. Then, the floating number mask $m \times m$ output is changed as much as the ROI and inserted in the 0.5 thresholds.

11.5. Uncertainty of object recognition

We create an uncertainty estimation process inspiring Stochastic YOLO [25]. In the ideal stage, the elliptical neural network has a quite possible uncertainty context and presents more accurate measures [26, 27], but elliptical networks have a considerable memory footprint which is their negative aspect. In this way, there is a general stage of models that can produce the class labels and predict the uncertainty better [28, 29], however, they also include a clear memory effect and expensive training times. The use of Monte-Carlo Drop (MC Drop) sampling is the best exchange between the cost and strength of data in the object recognition tasks under a possible framework. We add the MC Drop to the random predictions, which in turn, creates uncertainty estimations, has lightweight in terms of calculation, and is scaled well at the inference time. Thanks to the elliptical neural network [30], the study and analysis of uncertainty measures have received a great deal of attention and there are lots of research attempts to bring random prediction to the object recognition models to ensure their reliability. Such attempts can be summarized in 4 groups [25]: (1) direct output learning of Gaussian parameters for each coordinate of the border box, (2) the use of elliptical aspects (for example, elliptical neural networks) for having a quiet possible model, (3) the use of elliptical approximation sampling based on MC Drop deviation, and (4) the use of the general stage of models which produce the distribution of predictions, can be approximated as the gaussian parameters.

According to the similarity of architecture, we use the same method as following:

Gaussian YOLOv3 [31] has been adapted to YOLOv3 [32] and related losses to output the gaussian parameters instead of single and certain coordinates.

This approach significantly reduced the false positive cases, while conserving a similar inference time compared to YOLOv3. Similarly, He et al. proposed a new Kullback-Leibler (KL) loss to learn localization (for example, variance) of the uncertainty with border boxes, which can enable a voting plan to select border boxes [33]. Another work is to run YOLOv3 using MC drop on a large scale towards the dataset of the pedestrian using the variance defined by MC drop in YOLOv3 architecture to measure the spatial uncertainty [34]. As a result, recognition of cavities can be accepted or rejected based on that variance. However, the spatial quality of border boxes was not directly assessed by the specific quantitative measures, and this recognition does not require immediate performance in real-time or safety-crucial situations such as the mask region-based convolutional neural network (mask R-CNN).

12. Comparing the affections of models for sedan automobiles and motorcycle

In this analysis, we have assumed that all the data were gathered from cameras and all obstacle and vehicle parking spaces were determined.

1. The parking space is designed in MATLAB. To avoid complexities, two rectangles were implemented as two parked vehicles, and a line was considered for separating the street (Figure 4c) for better aesthetics and higher accuracy.
2. Calling the vehicle in MATLAB: We only have to use the vehicle dimensions calling command to create a vehicle with customized dimensions and standards for performing all the intended tests and analyzing their results. Table 1 presents the specific dimensions considered for these tests.
3. Creating and connecting the intended vehicle in the designed parking space for performing the final test and analyzing the automated parallel parking process. After connecting the vehicle with the parking space, the time for determining the parking location is based on the space between the two rectangular vehicles

(Figure 4d). Then, the RRT algorithm is used for finding the most efficient and shortest path for parallel parking.

4. Afterward, a controller must be designed to control the parking process. The NMPC controller was used for this. This controller will dramatically reduce the parking error by performing predictions at different times. This will lead to a clean and nearly errorless automated parking process (Figure 4d).

Table 1. Dimensions of the ego vehicle.

Vehicle Classification	Length	Width	Height	Wheelbase	Front Overhang	Rear Overhang
Automobile (sedan)	4.7 m	1.8 m	1.4 m	2.8 m	0.9 m	1.0 m
Motorcycle	2.2 m	0.6 m	1.5 m	1.51 m	0.37 m	0.32 m

13. Automated parallel parking process simulation

We enter the MATLAB simulation environment and design the entire intended block such as parking, NMPC, ego vehicle model, and visualizer. Then, we connect all these blocks and execute the model. You can witness the parking process and its related graphs in Figure 8. As you can see in Figure 8, if we compare the routing and vehicle parking processes of the NMPC and RRT methods, we can conclude that there are different results in the (x,y) axis, the vehicle velocity is in the cosine domain of 2m/s to -2m/s while the steering wheel position changes are in the cosine domain of 40 to -40 degrees. We conclude that a controller must be designed for improving the vehicle parking process. This study uses the RRT method for parking routing. This method was developed by adding an NMPC controller and the results show that the automated parking process is improved by the adage of this controller. This method was used in the control system of drilling tools as well [5, 35].

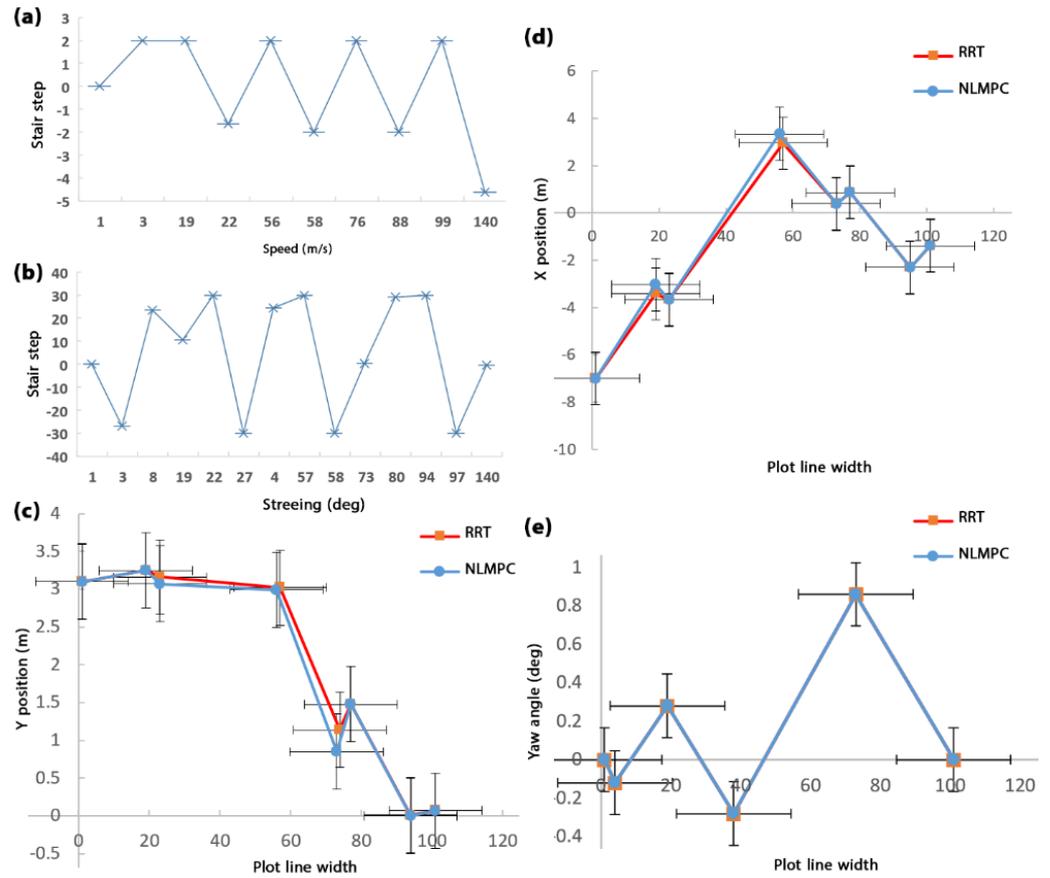


Figure 8. The nonlinear model predictive control controller graphs. (a) velocity, (b) steering. Comparison the results of RRT and NLMPC for (c) y position, (d) x position, and (e) yaw angle.

14. Analysis of environmental perception

After receiving information from the camera sensor and its image improvement in order to classify the images, a general and flexible perceptible framework has been presented. This approach effectively detects the items available in each image while creating classification masks for each sample. This method known as the mask R-CNN expands the faster R-CNN by adding an object mask prediction branch in parallel with other available branches to detect the connection box. Mask R-CNN can be easily trained by adding five frames per second small workload threshold to the faster R-CNN.

We require a dataset for training and executing the mask R-CNN. We can create a dataset by ourselves but this requires different objects, traffic signs, and people. Therefore, we have used the common objects in context (COCO) dataset to save time. A simple image processing method was used for determining the street and road lines that showed acceptable results.

14.1. Common objects in context (COCO) dataset:

It is a dataset based on the latest scientific developments. The COCO dataset includes images of 91 objects, which are easily recognizable by a 4-year-old child. With a total of 2.5 million samples labeled with 328 thousand images, the creation of a COCO dataset was performed based on the widespread participation of workers through new user interfaces to recognize the classification, sample detection, and sample division.

We require a system with graphical processors for this method. Google Collab is the most famous cloud processing system that safely facilitates heavy processing. We have

used Python in Google Collab. Then, we connected our account and implemented the process.

Access to a proper and standard film or information is one of the most important parts of this analysis. If our film or information is not produced correctly, we will face some problems in the environment understanding stage because our test results will be filled with errors. This could lead to disastrous or irreparable damages.

14.2. Environment understanding test and analysis stages:

- I. We want to recognize the road lines using a simple but practical method. These stages must be performed for this.
- II. We must modify and optimize the input image, extract the line dots graph from the grayscale, calculate the line slope, and share all of this information to determine the lines. Then, we must put a mask on each of these lines.
- III. Attention: This algorithm is only appropriate for linear and consistent paths. The deep learning method must be used to support all different roads and streets.
- IV. *Using a developed YOLO algorithm, known as Mask R-CNN*
- V. The system must receive appropriate training. The COCO dataset must be taught once to our system. Then, two graphs are extracted for analyzing the accuracy and waste (Figure 9, 10). The analysis of these graphs shows that our system is completely ready for item detection and it can be used in the video test stage (Figure 11).
- VI. *The combination of the YOLO-developed algorithm named mask R-CNN and the line recognition algorithm*

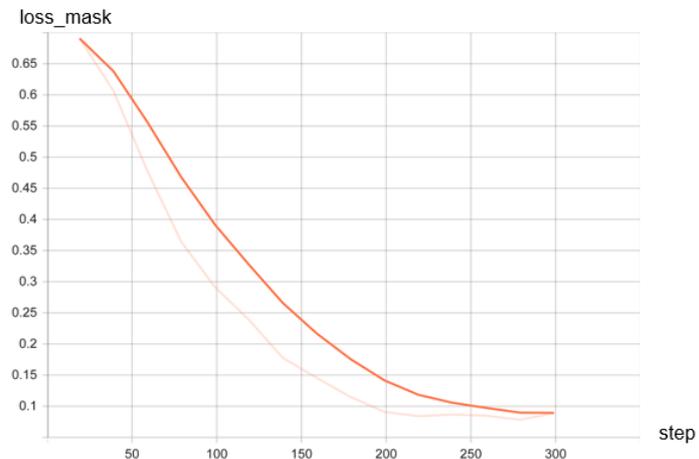


Figure 9. Loss mask in several steps.

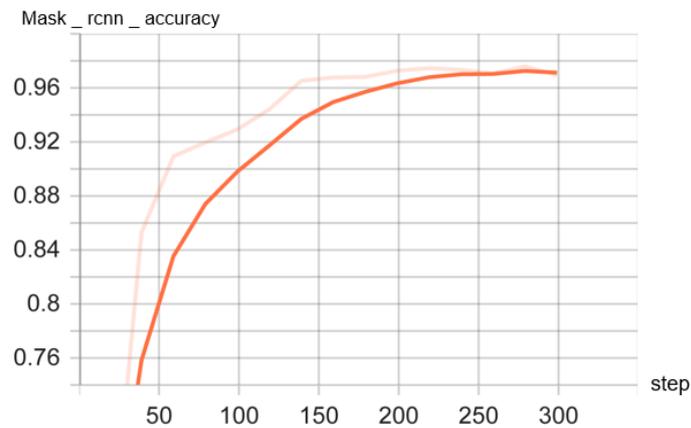


Figure 10. Mask accuracy in several steps.



Figure 11. The result of masking process.

Our last stage is to combine the investigated methods of this research to achieve a comprehensive and applicable technique because our perceptible system will be able to recognize many objects using the COCO dataset, but the volume of this dataset is large and the classification is not related to our work should be stored in the memory of our apparatus with this dataset. For this reason, we maintain the files required for driverless cars such as panels/signs, traffic signs, pedestrian detection, etc.; and the remaining ones are removed. We can recognize the lines in addition to the objects with the same small volume and add a line recognition algorithm to the mask R-CNN algorithm. We have not observed the combination of these algorithms in the previous research, but we do not have any claims for the novelty of a new algorithm or reaching it. Because, this work has been just tested to recognize the flat lines and eliminate our need; and for this reason, in other problems related to the combination of these two algorithms, we do not propose any other way.

There are limitations based on our assumptions and simplifications that are used in this analysis. Previous analytical and simulation studies have considered many new aspects with regard to the biological functions of the operators [36-38] of the analysis process that can be addressed in future work.

15. Conclusion

In this research, we have investigated two important parts of driverless cars to plan a control platform for a driverless car with analysis and evaluation of effective parameters. The first part includes the process control of the automated parking and the second one environmental perception. In automated parking, we are faced with challenges according to the recognition of problems such as the spatial, physical, and kinematic analysis of parking mechanism constraints as well as environmental limitations. However, with regard to the environmental and obstacle recognition information/data, finding a method that can perform the best routing is considered a challenge; and importantly since even the best-known routing methods have errors, therefore, it is very important to have a controller. The next challenge is to select the method and design of a controller to be able to control a routing process and reduce errors. RRTs method is used to create the random routes in such a way as to select its smallest path to the destination. To prevent this issue, we used a nonlinear predictive controller, because this controller causes the errors to be reduced by the path predictions and the impact not to occur in the future time, and in the end, parking is performed completely and automatically. The second part, i.e. environmental perception, is very important same as the parking process, and associated with the challenges of correct recognition and proper classification without system training. To recognize the objects correctly, we improve the image quality and remove its noise. Then, we decrease the recognition errors using the uncertainty process with the help of the Gaussian method. We could recognize the dynamic objects and traffic signs without system training using a dataset named COCO; and also, add a simple code based on the image processing to the object recognition algorithm to detect the road lines.

Funding: None

Conflicts of interest: No potential conflict of interest was reported by the authors

Acknowledgements: None

Reference

- [1] Chen, Z., Zhang, Y., Wu, C., & Ran, B. (2019). Understanding individualization driving states via latent Dirichlet allocation model. *IEEE Intelligent Transportation Systems Magazine*, 11(2), 41-53.
- [2] Ziegler, J., Bender, P., Schreiber, M., Latégahn, H., Strauss, T., Stiller, C., ... & Zeeb, E. (2014). Making bertha drive—an autonomous journey on a historic route. *IEEE Intelligent transportation systems magazine*, 6(2), 8-20.
- [3] Gholampour, S., Fatourae, N., Seddighi, A. S., & Seddighi, A. (2017). Numerical simulation of cerebrospinal fluid hydrodynamics in the healing process of hydrocephalus patients. *Journal of Applied Mechanics and Technical Physics*, 58(3), 386-391.
- [4] Xue, J., Van Gelder, P. H. A. J. M., Reniers, G., Papadimitriou, E., & Wu, C. (2019). Multi-attribute decision-making method for prioritizing maritime traffic safety influencing factors of autonomous ships' maneuvering decisions using grey and fuzzy theories. *Safety Science*, 120, 323-340.
- [5] Gholampour, S., & Deh, H. H. H. (2019). The effect of spatial distances between holes and time delays between bone drillings based on examination of heat accumulation and risk of bone thermal necrosis. *Biomedical engineering online*, 18(1), 1-14.
- [6] Gholampour, S., Gholampour, H., & Khanmohammadi, H. (2019). Finite element analysis of occlusal splint therapy in patients with bruxism. *BMC Oral Health*, 19(1), 1-9.
- [7] Gholampour, S., & Gholampour, H. (2020). Correlation of a new hydrodynamic index with other effective indexes in Chiari I malformation patients with different associations. *Scientific Reports*, 10(1), 1-13.
- [8] Shariati, A., Shamekhi, A. H., Ghaffari, A., Gholampour, S., & Motaghd, A. (2019). Conceptual Design Algorithm of a Two-Wheeled Inverted Pendulum Mobile Robot for Educational Purposes. *Mechanics of Solids*, 54(4), 614-621.
- [9] Li, B., & Shao, Z. (2015). A unified motion planning method for parking an autonomous vehicle in the presence of irregularly placed obstacles. *Knowledge-Based Systems*, 86, 11-20.
- [10] Priisalu, M., Pirinen, A., Paduraru, C., & Sminchisescu, C. (2022, January). Generating scenarios with diverse pedestrian behaviors for autonomous vehicle testing. In *Conference on Robot Learning* (pp. 1247-1258). PMLR.
- [11] Li, B., Wang, K., & Shao, Z. (2016). Time-optimal maneuver planning in automatic parallel parking using a simultaneous dynamic optimization approach. *IEEE Transactions on Intelligent Transportation Systems*, 17(11), 3263-3274.

-
- [12] Vorobieva, H., Minoiu-Enache, N., Glaser, S., & Mammari, S. (2013, April). Geometric continuous-curvature path planning for automatic parallel parking. In 2013 10th IEEE international conference on networking, sensing and control (ICNSC) (pp. 418-423). IEEE.
- [13] Gholampour, S. (2021). Computerized biomechanical simulation of cerebrospinal fluid hydrodynamics: Challenges and opportunities. *Computer Methods and Programs in Biomedicine*, 200, 105938-105938.
- [14] Gholampour, S., & Bahmani, M. (2021). Hydrodynamic comparison of shunt and endoscopic third ventriculostomy in adult hydrocephalus using in vitro models and fluid-structure interaction simulation. *Computer Methods and Programs in Biomedicine*, 204, 106049.
- [15] Gholampour, S., & Fatouraei, N. (2021). Boundary conditions investigation to improve computer simulation of cerebrospinal fluid dynamics in hydrocephalus patients. *Communications biology*, 4(1), 1-15.
- [16] Gholampour, S., & Mehrjoo, S. (2021). Effect of bifurcation in the hemodynamic changes and rupture risk of small intracranial aneurysm. *Neurosurgical Review*, 44(3), 1703-1712.
- [17] LaValle, S. M. (1998). Rapidly-exploring random trees: A new tool for path planning.
- [18] Karaman, S., & Frazzoli, E. (2011). Sampling-based algorithms for optimal motion planning. *The international journal of robotics research*, 30(7), 846-894.
- [19] Gao, W., Zhang, X., Yang, L., & Liu, H. (2010, July). An improved Sobel edge detection. In 2010 3rd International conference on computer science and information technology (Vol. 5, pp. 67-71). IEEE.
- [20] He, K., Zhang, X., Ren, S., & Sun, J. (2015). Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 37(9), 1904-1916.
- [21] Dong, X., Wang, G., Pang, Y., Li, W., Wen, J., Meng, W., & Lu, Y. (2011, July). Fast efficient algorithm for enhancement of low lighting video. In 2011 IEEE International Conference on Multimedia and Expo (pp. 1-6). IEEE.
- [22] Deng, G. (2010). A generalized unsharp masking algorithm. *IEEE transactions on Image Processing*, 20(5), 1249-1261.
- [23] Li, L., Wang, R., Wang, W., & Gao, W. (2015, September). A low-light image enhancement method for both denoising and contrast enlarging. In 2015 IEEE International Conference on Image Processing (ICIP) (pp. 3730-3734). IEEE.
- [24] Lin, T. Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2117-2125).
- [25] Azevedo, T., de Jong, R., Mattina, M., & Maji, P. (2020). Stochastic-yolo: Efficient probabilistic object detection under dataset shifts. *arXiv preprint arXiv:2009.02967*.
- [26] Welling, M., & Teh, Y. W. (2011). Bayesian learning via stochastic gradient Langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)* (pp. 681-688).
- [27] Blundell, C., Cornebise, J., Kavukcuoglu, K., & Wierstra, D. (2015, June). Weight uncertainty in neural network. In *International conference on machine learning* (pp. 1613-1622). PMLR.
- [28] Ovadia, Y., Fertig, E., Ren, J., Nado, Z., Sculley, D., Nowozin, S., ... & Snoek, J. (2019). Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *Advances in neural information processing systems*, 32.
- [29] Wei, P., Ball, J. E., & Anderson, D. T. (2018). Fusion of an ensemble of augmented image detectors for robust object detection. *Sensors*, 18(3), 894.
- [30] Kendall, A., & Gal, Y. (2017). What uncertainties do we need in bayesian deep learning for computer vision?. *Advances in neural information processing systems*, 30.
- [31] Choi, J., Chun, D., Kim, H., & Lee, H. J. (2019). Gaussian yolov3: An accurate and fast object detector using localization uncertainty for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 502-511).
- [32] Redmon, J., & Farhadi, A. (2018). Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*.
- [33] He, Y., Zhu, C., Wang, J., Savvides, M., & Zhang, X. (2019). Bounding box regression with uncertainty for accurate object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 2888-2897).
- [34] Myojin, T., Hashimoto, S., Mori, K., Sugawara, K., & Ishihama, N. (2019, September). Improving reliability of object detection for lunar craters using Monte Carlo dropout. In *International Conference on Artificial Neural Networks* (pp. 68-80). Springer, Cham.
- [35] Hassanalideh, H. H., & Gholampour, S. (2020). Finding the optimal drill bit material and proper drilling condition for utilization in the programming of robot-assisted drilling of bone. *CIRP Journal of Manufacturing Science and Technology*, 31, 34-47.
- [36] Gholampour S. FSI simulation of CSF hydrodynamic changes in a large population of non-communicating hydrocephalus patients during treatment process with regard to their clinical symptoms. *PLoS One*. 30, 13(4), e0196216.
- [37] Taher M, Gholampour S. Effect of ambient temperature changes on blood flow in anterior cerebral artery of patients with skull prosthesis. *World neurosurgery*. 2020, 135, e358-65.
- [38] Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., ... & Zitnick, C. L. (2014, September). Microsoft coco: Common objects in context. In *European conference on computer vision* (pp. 740-755). Springer, Cham.