Article

Deep Learning-Based NMPC for Local Motion Planning of Last-Mile Delivery Robot

Muhammad Imad 1,†, Oualid Doukhi 1,*,†, Deok Jin Lee 1,*,†, Ji chul Kim 2 and Yeong Jae Kim 2

1 Department of Mechanical Design Engineering, Jeonbuk National University, Jeonju-si 54896, Korea
2 Department of Smart Machine Technology, Korea Institute of Machinery & Materials (KIMM), Daejeon 34103, Korea
* Correspondence: doukhioualid@jbnu.ac.kr (O.D.); deokjlee@jbnu.ac.kr (D.J.L)
† These authors contributed equally to this work.

Abstract: Feasible local motion planning for autonomous mobile robots in dynamic environments requires predicting how the scene evolves. Conventional navigation stakes rely on a local map to represent how a dynamic scene changes over time. However, these navigation stakes depend highly on the accuracy of the environmental map and the number of obstacles. This study uses semantic segmentation-based drivable area estimation as an alternative representation to assist with local motion planning. Notably, a realistic 3D simulator based on an Unreal Engine was created to generate a synthetic dataset under different weather conditions. A transfer learning technique was used to train the encoder-decoder model to segment free space from the occupied sidewalk environment. The local planner uses a nonlinear model predictive control (NMPC) scheme that inputs the estimated drivable space, the state of the robot, and a global plan to produce safe velocity commands that minimize the tracking cost and actuator effort while avoiding collisions with dynamic and static obstacles. The proposed approach achieves zero-shot transfer from a simulation to real-world environments that have never been experienced during training. Several intensive experiments were conducted and compared with the dynamic window approach (DWA) to demonstrate the effectiveness of our system in dynamic sidewalk environments.

Keywords: semantic segmentation; obstacle avoidance; zero-shot transfer; sidewalk autonomous delivery robots

1. Introduction

The autonomous systems domain is developing extensively worldwide in terms of a variety of new utilities and the degree of interest among long-established players in the civilian and military communities [1,2]. Autonomous robotic systems have great potential for enhancing safety and real-time operational efficiency [3]. Furthermore, owing to the mercurial growth of the e-commerce industry, the need for efficient, affordable, and feasible last-mile deliveries has become more critical [4]. Sidewalk autonomous delivery robots (SADRs) are one of these potential technologies and have been the focus of research in recent years [5]. The SADRs were designed and engineered to deal with unexpected real-world scenarios without human intervention. Therefore, SADRs need to achieve their tasks in ambivalent and amorphous conditions, which pushes them to rely on their perception systems to perceive the surrounding environment while avoiding obstacles in real-time, which is also a performance indicator for mobile robots. Considering these factors, a mobile delivery robot with autonomous navigation potential has a higher application value, which is a vital area for the further expansion of mobile robot applications and is considered a critical factor in maximizing the intelligence level of the logistics industry.
The deployment of SADRs relies primarily on three major tasks: perception of the environment, path planning, and subsequent vehicle controls. Whereas the latter has proven to be a solved problem, the first two are still limited and have to be improved. Some of the suggested solutions involve the implementation of GPS devices accompanied by three-dimensional LiDARs and inertial measurement (IMU) [6,7], while a few utilize 2D LiDARs [8,9]. However, a wide range of implementation of these methods is quite difficult owing to the high cost, complexity, and unpredictability in specific scenarios. Existing methods have addressed these problems by dividing the robot navigation task into two modules: a global planning module, which creates trajectories from the robot’s current position to a given target goal [10], and a path-following module, which keeps the robot close to the planned path [11]. However, both modules depend on environmental characteristics and robot dynamics, making them sensitive and requiring re-tuning for each scheme. Moreover, these methods suffer from the stochastic dynamic behavior of obstacles and high computational costs when applied to highly unstructured environments.

More recently, researchers have proposed modular and end-to-end-based approaches that have proven effective in various vision-based robotic perception tasks [12,13]. A model-based cognitive mapping and planning approach was successfully illustrated in first-person visual navigation [14,15]. Reinforcement learning (RL) has been utilized for robot goal-driven navigation using visual input data [16]. Applying end-to-end supervised learning to steer a drone in addition to a manually labeled image-action pair was presented in [17]. An end-to-end training methodology for object manipulation tasks using unlabeled video data was described in [18]. Although these learning-based approaches seem captivating, they usually require an immense amount of training data. Piling up learnable data that should cover all types of scenarios for learning a control policy is costly and time-consuming and poses several challenges. Training a supervised learning-based model from scratch requires a large, diverse dataset and considerable human effort to label the data. However, RL depends on trial-and-error experiences, limiting the ability of dainty robots to perform various tasks. Moreover, online training in real-world scenarios is costly and prone to failure, thereby minimizing the learning capability of most RL algorithms.

A substitute for the aforementioned methods is to train the robot in a virtual environment to increase its capability and reduce training costs. Most methods based on this idea rely on gathering labeled data from a virtual environment [19–21]. However, the data collected from these virtual simulated environments have a considerable reality gap and are task-specific. For instance, to collect a sidewalk dataset for a customized delivery robot, it is impossible to use the available self-driving car simulators in a sidewalk environment because the data collected with a car configuration cannot be used to train a delivery robot. Therefore, defining a customized simulated environment, collecting data from that environment, training a model, and then transferring that model from the virtual to the real world have been challenging problems in computer vision and robotics.

This paper presents a deep learning-based mapless local motion-planning technique for an autonomous mobile robot. First, a realistic simulator was developed based on Unreal Engine [22]. Second, a synthetic dataset was collected to train the deep learning module. The developed algorithm exploits semantic segmentation and a local motion planner property to provide low-level safe motion commands that drive SADRs (see Figure 1) on a sidewalk while avoiding dynamic and static obstacles. The proposed technique fuses information from the semantic segmentation module, GPS, and IMU to provide a robust motion plane in challenging situations without relying on mapping. A nonlinear model predictive (NMPC) [23] was used in the proposed method, which is considered a popular solution in local trajectory planning, owing to the advantages of movement fluency while considering the robot’s kinematic constraints, such as linear and angular speed limit.

However, existing traditional NMPC is prone to failure if there is no accurate obstacle information. Accordingly, a new NMPC cost function was proposed. It uses semantic segmentation to define the drivable area for the SADRs and enables the ability to perceive the obstacles in the surrounding environment, which allows the robot to avoid colliding
with them. Computer simulations and real-world demonstrations show that the efficiency of the improved NMPC is increased in terms of motion safety and smoothness.

For training and experimentation, a dataset was collected from different simulated sidewalk scenarios, including harsh weather conditions, such as fog and snow. The dataset was labeled manually to train a semantic segmentation model. A transfer learning technique with a few frames from each scenario was used to increase the model generalization ability and achieve zero-shot sim-to-real transfer. The proposed architecture was separated into two modules: perception and local motion planner modules. The perception module converts RGB images into semantic segmentation using the trained model, which defines the drivable area. The local motion planner uses the obtained segmentation results to plan a set of trajectories within 6 seconds of prediction. A cost function was designed to select the most feasible trajectory that achieves obstacle avoidance and global path tracking. The perception module uses a convolution neural network-based encoder–decoder architecture, where MobileNetv2 [24] serves as an encoder, and on top of that, a simple yet efficient decoder has been defined to uplift low-level information. As the segmentation output generated by the synthetic and real data perception models is invariant, the model was deployed directly into the real world without further training and fine-tuning. The proposed architecture is more general and can be deployed in various scenarios and tasks. Unlike [25], which uses multiple-class segmentation, which increases the computational cost of the model, we use a binary segmentation mask to bridge the visual and planning modules. The mobile delivery robot is supposed to steer on sidewalks and streets and not on the main road. Therefore, we do not need to segment obstacles, such as cars, motorbikes, or trucks. The segmentation mask was divided into binary classes: drivable areas and obstacles, including people, light poles, and benches.

To demonstrate the validity of our algorithm, the proposed model and the DeeplabV3 [26], PSPNET [27], FCN [28], and U-Net [29] were trained on simulated data, and their performances were compared in benchmark tasks in both the virtual and real world. In addition, the local motion-planning module was compared to the dynamic window approach. The obtained results demonstrate the effectiveness of the proposed method in terms of obstacle avoidance, motion smoothness, and sim-to-real perception system transferability.

The remainder of this paper is organized as follows: Section II introduces related work. In Section III, the proposed modular architecture is discussed. Section IV describes the evaluation setup, tasks, and scenarios. Section V presents experimental results. Finally, Section VI concludes the proposed work.
2. Related Work

In recent years, vision-based autonomous navigation has become a popular field in applications ranging from intelligent vehicles to autonomous robots [25,30]. For instance, Zaidner [31], established a data fusion scheme for navigation, which effectively fused the localization data from multiple sensors; visual odometry (VO), GPS, inertial navigation system (INS), and wheel odometry. However, as highlighted by the author, there is a trade-off between cost and accuracy, and the data fusion algorithm can fail if the sensors significantly differ from each other. In [32], the autonomous navigation of mobile robots based on cognitive development was proposed, which achieves the expected navigation target by simulating animals, but the navigation path is not optimal. In [33], autonomous navigation based on fuzzy logic and reinforcement learning was investigated. Although the simulation results show that the method can significantly improve the behavior and learning speed of a robot, autonomous navigation cannot be implemented quickly. In [34], a mobile robot path planning method based on deep reinforcement learning was proposed. The learned policy selects the optimal mobile action that allows the mobile robot to reach the target position while avoiding the obstacles.

Several other studies have been proposed for autonomous and collision avoidance when obstacle locations and the global environment are known [35,36]. For instance, this scheme [37] presents a mobile robot navigation control system based on the integration of laser SLAM localization and real-time collision avoidance control to provide personnel guidance for daily life services. However, these techniques depend highly on the environmental map representation method (metric or feature-based). One of the predominant advantages of our scheme over these other methods is that it does not require a prior map of the surroundings. With the current developments of edge computing systems, alternative approaches have been proposed for reactive motion planning, where the system depends on the instant perception of its surrounding environment for decision-making.

The work in [38] proposed an end-to-end method for training deep learning models to infer the best action given an input color image. Another study [39] presented a data-driven motion planner based on a CNN model, which was trained using an existing motion planner. However, these studies are still limited in their generalization capabilities for new inexperienced scenarios. These issues have caused the research community to shift its focus to different technologies.

More recently, alternative approaches to robot motion planning have been based on semantic segmentation techniques. The work in [40] proposed a vision-based navigation system where in the first phase, a supervised generative network is trained to map outdoor images with rich information. Second, binary semantic segmentation was used to differentiate between obstacles and roads. Although the results verify the importance of the proposed method, running a two-stage perception module is computationally expensive and the proposed solution does not provide information on how binary segmentation is used in navigation.

Unlike the previous method, the authors of [41] exploited a custom-trained segmentation network and a low-RGB camera to produce smooth trajectories and stable control in different vineyard scenarios. Although the method showed outstanding results, this framework cannot be directly translated into an unstructured, dynamic environment, such as a sidewalk.

Other work, such as [42], proposed a vision-based navigation scheme that enables autonomous movement in indoor scenes; only a webcam was used as a perception sensor. However, as the name suggests, this method is restricted only to the indoor environment. A vision-based navigation scheme was presented in [43]. It uses a deep CNN to recognize road regions in front of the robot. Although the results demonstrate the possibility of visual navigation using only a single camera, the proposed method requires a large dataset based on real-world scenarios for training, which is time-consuming.
3. Proposed Methodology

Many conventional motion-planning methods can only handle static obstacles. The proposed algorithm does not require obstacles to be static for navigation. This section describes an autonomous navigation technique for mobile delivery robots. The objective is to retrieve an optimal trajectory $T^*$ given the estimated drivable area $d_t$ from the perception module.

3.1. Problem Formulation

The proposed modular architecture contains a perception module (PM) and motion planner module (MPM) see Figure 2. PM is responsible for estimating the drivable area $d_t$. The MPM takes the drivable surface $d_t$, robot pose $P_t$, and global waypoint $w_t$ as inputs and returns the best trajectory $T^*$ that minimizes the optimization objective $J$, which ensures obstacle avoidance, motion smoothness, and reaching the global goal.

![Figure 2. The overall block diagram of the proposed algorithms: (A) Unreal-Engine-based simulation of dynamic city with moving pedestrian. (B) The perception module based on semantic segmentation. (C) Local motion planner based on NMPC.](image)

3.2. Perception Module

The primary function of the perception module is to estimate the drivable area, $d_t$, and pass it to MPM. During the training stage, $d_t$ is learned from a synthetic image $x_{syn}^t$ generated from an Unreal-Engine-based simulator. Afterward, in the testing phase, the perception module uses real color images, $x_{real}^t$. Its function is expressed as:

$$d_t = S(x_{real}^t, \Theta^*),$$

where $S$ represents the segmentation model, and $\Theta^*$ denotes the optimal model parameter.

The proposed model $S$ employed a transfer learning-based encoder–decoder architecture, as shown in Figure 3, where the classification model is transferred into a semantic segmentation model, which not only minimizes the need for a large dataset but also decreases the training time and increases the model accuracy. MobileNetV2 was adopted as an encoder for model $S$, containing a convolutional layer combined with downsampling layers to produce a low-resolution tensor for high-level information. The architecture of the encoder is listed in Table 1. The decoder consists of matching convolutional layers coupled with upsampling layers to increase the size of the spatial tensor and to generate a high-resolution segmentation output. However, simply piling the encoder and the decoder architecture can result in a loss of low-level information. Therefore, the segmentation map boundaries initiated by the decoder will be faulty. Consequently, the decoder was allowed to access the low-level features produced by the encoder layers through skip connections. The intermediate outputs of the encoder are anchored to the inputs to the median layers of the decoder at the relevant position, as shown in Figure 3. Model $S$ was designed to achieve the two main characteristics. First, although the visual appearances of $x_{real}^t$ and $x_{syn}^t$ differ, their semantic segmentation output is invariant, which implies that model $S$ allows zero-shot sim-to-real transfer while being
trained entirely using synthetic data. Second, segmentation model $S$ only uses a monocular camera as input, which is cheaper than depth cameras or laser rangefinders. A synthetic dataset of 800 images with their corresponding binary mask was used for training. First, all the obtained images were reshaped to a fixed input dimension, $24 \times 24$, and normalized with the value from 0 to 1. Afterward, the model was trained for 50 epochs with a learning rate $lr = 0.0003$ and a batch size of 4. The Nvidia GTX 1080 GPU with 16 GB of memory and Cuda 11 with Tensorflow 2.2 were utilized for training. With the mentioned hardware configuration, the training phase of the segmentation head takes approximately 30 min.

![Figure 3. Training of a pre-trained model using transfer learning and testing in real-world scenarios.](image)

Table 1. The overall encoder architecture. $t$ is the expansion factor, $c$ is the number of output channels, $n$ is the number of repeated layers and $s$ is the stride.

| Input Operator | Input | Operator | t | c | n | s |
|----------------|-------|----------|---|---|---|---|
| $224^2 \times 3$ | conv2D | - | 32 | 1 | 2 |
| $112^2 \times 32$ | bottleneck | 1 | 16 | 1 | 1 |
| $112^2 \times 16$ | bottleneck | 6 | 24 | 2 | 2 |
| $56^2 \times 24$ | bottleneck | 6 | 32 | 3 | 2 |
| $14^2 \times 64$ | bottleneck | 6 | 96 | 3 | 1 |
| $14^2 \times 96$ | bottleneck | 6 | 160 | 3 | 2 |
| $7^2 \times 160$ | bottleneck | 6 | 320 | 1 | 1 |
| $7^2 \times 320$ | conv2D $1 \times 1$ | - | 1280 | 1 | 1 |
| $7^2 \times 1280$ | avgpool $7 \times 7$ | - | - | 1 | - |
| $1^2 \times 1280$ | conv2D $1 \times 1$ | - | k | - | 60 |

3.3. Local Motion Planning Module

Local motion planning is crucial for robots operating in unstructured dynamic environments. Feasible and safe trajectories that allow obstacle avoidance are crucial in such environments. The proposed MPM combines optimal motion planning and optimal control into a unified nonlinear model predictive control (NMPC) with constraints on states and manipulated variables, intending to minimize a customized cost function $J$ that guarantees global waypoint tracking, actuator effort minimization, and obstacle avoidance by exploiting the segmented drivable area obtained from the perception module. At each iteration of the NMPC, the following optimization problem (see Equation (2)) was solved based on the
latest robot state measurement, \( x_k \) at time \( k \). The first optimized control input \( U \) within the specified horizon is applied until a new NMPC update arrives.

\[
\min_{U(k)} \sum_{k=k_s}^{k_f} J(x(k), U(k), k), \quad (2a)
\]

subject to

\[
x(k_s) = x_s, \quad (2b)
\]

\[
x(k+1) = x(k) + v(k)\cos(\psi(k)), \quad (2c)
\]

\[
y(k+1) = y(k) + v(k)\sin(\psi(k)), \quad (2d)
\]

\[
\psi(k+1) = \psi(k) + v_{\psi}(k), \quad (2e)
\]

\[
v_{\text{min}} \leq U_1 \leq v_{\text{max}}, \quad (2f)
\]

\[
v_{\text{min}} \leq U_2 \leq v_{\psi_{\text{max}}}, \quad (2g)
\]

where \( x(k) \) is the state, \( U(k) = [U_1, U_2] \) is the linear and angular control input at time \( t \) and \( J \) is a time-varying cost function. The objective is to retrieve a control input that minimizes the cost \( (2a) \) subject to the robot’s initial state \( (2b) \), system kinematics \( (2c), (2d), (2e) \), and general inequality \( (2f) \) and \( (2g) \) constraints. As shown in Equation \( (3) \), the cost function \( J \) contains two parts, \( J_c \) for obstacle avoidance and \( J_s \) for waypoint tracking.

\[
J(x, k, U) = J_c(x, k) + J_s(x, k, U). \quad (3)
\]

The \( J_s(x, k, U) \) has a quadratic form with the goal of minimizing the distance between the robot and the given waypoint \( w_r \), while considering the minimization of the actuator efforts, as presented in Equation \( (4) \).

\[
J_s(x, k, U) = \frac{1}{2} \| x - w_r \|^2 Q + \frac{1}{2} \| U - U_r \|^2 R, \quad (4)
\]

where \( R \) and \( Q \) are the positive semi-definite weighting matrices. \( U_r \) denotes the equilibrium control input vector.

\( J_c(x, k) \) is represented by a one-sided quadratic weighting function. It uses the estimated drivable area, \( d_t \), to enable the NMPC obstacle avoidance capability. For obstacle checking, a pseudo-laser scan was generated by taking the top view of the drivable area \( d_t \). Subsequently, the contour of the drivable area is determined, and the distance from the camera to the drivable area contour is computed. This allows the generation of a vector of distances that resembles the output of 2D LiDAR (see Figure 4). The closest distance to the robot, \( c_d(x, k) \), was used to generate a soft constraint that allowed collision-free motion, as presented in Equation \( (5) \).

\[
J_c(x, k) = \frac{1}{2} (\max\{0, c_d(x, k)\})^2. \quad (5)
\]

The proposed MPM-based NMPC not only considers static obstacles such as poles and trees but also introduces a dynamic obstacle avoidance factor to evaluate the next motion to avoid collision in advance.
4. Experimental Results and Discussion

Intensive experiments were conducted in simulated and real-world environments to evaluate the feasibility of the proposed method. A simulation environment was built based on the Unreal Engine game editor. Training for the perception module was performed by generating sample images of a simulated robot equipped with a localization system and a front-facing color camera. The robot starts from an initial position and attempts to reach the goal destination while avoiding collisions with obstacles.

4.1. Perception Module Verification in Simulation

This sub-section briefly discusses the performance evaluation of semantic segmentation models used for drivable area estimation in a simulated environment. For semantic segmentation, different evaluation matrices can be used to demonstrate the performance of the proposed deep learning model [44]. The evaluation matrices used in this study were as follows:

4.1.1. Pixel Accuracy ($P_{\text{acc}}$)

This is the most extensively used valuation benchmark for semantic segmentation []. It is defined as the accuracy of pixel-wise forecasting, where the comparison has been made in each pixel one-at-a-time with the ground truth mask, given as:

$$P_{\text{acc}} = \frac{\sum_{i=0}^{k} (P_{ii})}{\sum_{i=0}^{k} \sum_{j=0}^{k} (P_{ij})},$$  \hspace{1cm} (6)

where $k$ represents the total number of pixels in the test image, and $P_{ii}$ is predicted pixels as class $i$, and the ground truth is represented as $P_{ij}$, which is the number of pixels of class $i$ predicted as class $j$.

4.1.2. Intersection Over Union (IoU)

The IoU metric, also known as the Jaccard index, is a commonly used evaluation metric for calculating the performance of segmentation models. It is generally used to calculate the performance by calculating the intersection and union between the ground truth and prediction, expressed as:

$$\text{IoU} = J_{d}(A, B) = \frac{|A \cap B|}{|A \cup B|},$$  \hspace{1cm} (7)

where $B$ shows the predicted segmentation maps, and $A$ represents the ground truth.

4.1.3. Mean-IoU (mIoU)

Another comprehensively used metric for semantic segmentation models is mean IoU. This was intended as the average value of IoU overall label classes. Substantially, they are
used to outline the potential of semantic segmentation models. It usually ranges between 0 and 100 and is given as:

\[
mIoU = \frac{1}{k+1} \sum_{i=0}^{k} \frac{t_p}{\sum_{j=0}^{k} f_n + \sum_{j=0}^{k} f_p - f_n},
\]

where \( k \) represents the total classes, \( t_p \) is the number of true positives, and \( f_p \) and \( f_n \) are false positives and false negatives, respectively.

In this study, the segmentation model was trained on a synthetic dataset and tested in a real-world setting. First, the model was trained from the first layer without using pretrained weights, owing to the small size of the dataset. Subsequently, training the model from scratch precedes very strong overfitting, as shown in Figure 5, which shows that the training loss does not decrease. In the second test, the model was initiated using ImageNet classification pretrained weights. Figure 5 shows that the training loss converges smoothly without overfitting. The predicted binary semantic segmentation results for both training scenarios are shown in Figure 6. It is clear that transfer learning helps to increase the model accuracy outperforming the model trained from scratch, as the mIoUs shows.

A comparative study is conducted to demonstrate the effectiveness of the proposed segmentation model. DeeplabV3 [26], PSPNET [27], FCN [28], and U-Net [29] architectures were used as a baselines for the proposed model. Figure 7 shows that DeepLabV3 demonstrated preferable outcomes in the context of model accuracy. However, owing to the substantial number of parameters in DeepLabV3, it is not suitable for real-time operation on a single-board device.

In contrast, our proposed method achieves acceptable performance, outperforming U-NET, FCN, and PSPNET with a trivial set of parameters, and manages to run in real-time as well. Table 2 presents a detailed comparison of the proposed scheme and baselines. DeepLabV3 shows high accuracy in terms of the PA, MPA, and mIoU. However, owing to their large parameter values, a large amount of computational power is required. By contrast, our model can be run using only a standard computer. The sample segmentation outputs from the proposed model and U-Net model are shown in Figure 8, where U-NET fails to segment the free space precisely. Distinctly, our model does not have false positives.

![Figure 5. Comparison between model from scratch and model initialized with pre-trained classification weights.](image)
Figure 6. Comparison of the proposed method with baseline methods in terms of accuracy.

Figure 7. Binary semantic segmentation results from synthetic data in different simulation environment scenarios. The first column presents the image input and the corresponding ground truth (GT) semantic label. The second column shows the obtained results from the model that was pretrained. The third column presents the obtained results from the model that was trained from scratch.
Table 2. Comparison of the proposed scheme with the baseline methods in terms of parameters, pixel accuracy PA, MPA, and IoU.

| Method      | Parameters | PA%  | MPA% | MIoU% |
|-------------|------------|------|------|-------|
| Deeplab-V3  | 41.25 M    | 97.46| 93.84| 96.05 |
| PSPNET-MobileNet | 63.967 M | 84.91| 72.53| 77.03 |
| FCN8-VGG    | 134.286 M  | 83.19| 74.63| 78.43 |
| U-Net-RGB   | 31.033 M   | 76.31| 68.54| 78.50 |
| Ours        | 3.984 M    | 94.41| 91.59| 93.78 |

Figure 8. Semantic segmentation results of the proposed method and the U-Net model. The first column presents the image input and the corresponding ground truth (GT) semantic label. The second column presents the results obtained from our model. The third column presents the results obtained from the U-Net model.

4.2. Motion Planning Module Verification in Simulation

Simulation experiments were conducted to validate the proposed local motion planning module. First, a dynamic sidewalk environment was created using the Unreal Engine game editor, as shown in Figure 9. It contains animated walking pedestrians that act as dynamic obstacles and different static obstacles, such as bus stands, road borders, and public seats. The objective of the mission was to navigate the robot from the initial starting point to the destination. The proposed algorithm was benchmarked using the conventional dynamic window approach, DWA [45]. Our method demonstrates its potential while moving through a narrow sidewalk and more complex maneuvers, such as dynamic collision avoidance. However, the DWA showed a jerky motion, and the robot crashed before reaching the final goal, as depicted in Figure 10D. Moreover, the proposed approach automatically changes the robot’s heading and avoids obstacles and then returns to the sidewalk center once the obstacle is passed (see Figure 10 and the attached video). Figure 11
shows the path taken by the robot. The blue line represents the path while following the smooth commands, \( v_x, v_\phi \), generated by the proposed scheme. The commanded velocities are shown in Figure 12. The first row represents the linear forward velocity \( v_x \). It can be observed that our algorithm generates smoother commands without violating the speed limit of 1.5 m/s. At the time stamp \( t = 6 \times 10^5 \), the speed was approximately 0 m/s. There was no available drivable area in this scenario owing to the walking person who took over the entire free space, and the robot speed increased when the person moved away. The second row presents the commanded angular velocities. Compared to the DWA command, our method shows a more stable velocity command while respecting the imposed constraints. It is recommended that the reader watch supplementary videos to better understand the simulated scenario.

**Figure 9.** Simulation environment: (A) mobile robot with a forward facing camera. (B) Dynamic and static obstacles are included in the sidewalk. (C) Given a set of waypoints \( w_1, w_2, w_3 \), the robot must reach the goal point.
Figure 10. Different emerging behaviors during simulation experiments. At each snapshot, a long exposed trajectory is shown. (A–C) The robot adjusts its heading and forward velocity to avoid obstacles along the predicted trajectory using the proposed approach. (D) Snapshot shows where the robot crashed using the DWA algorithm.

Figure 11. The path followed by the robot while avoiding obstacles in a simulated environment.
4.3. Hardware Experiments

The hardware used for the real-time experimentation is Scout 2.0, developed for industrial applications by Agile Robotics, which has a compact body and low power consumption. In addition, scout 2.0 has its central control system, allowing us to customize complex modes of operation. Users can communicate with the central controller through the CAN Bus protocol, and an open-source SDK and robot operating system (ROS) package are provided.

The optimized neural network and the presented controller were embedded into the robot PC. It comprises a CPU Intel Core i7-7567U processor @ 3.5 GHz Turbo and DDR4 RAM of 32 GB. With such configurations, the camera attached to the Scout provides RGB images at 30 FPS, while an optimized deep neural network can process the RGB images on the CPU at 22 FPS (sensor configuration for the real-time experimentation is shown Figure 1). Intensive real-world experiments have shown that the proposed approach performs comparably to the simulation results of a real mobile robot. Motions such as heading and forward speed adjustment and aligning to the sidewalk center were observed. The accompanying videos were available for real-time experiments. Figure 13b shows the environment in which the experiments were conducted. Figure 13a shows the real-time results of the semantic segmentation of the drivable area on a sidewalk. The method effectively translates from the simulation into the real world and steers clear of those objects that were not comprehended in the labeling class.

Figure 14 shows that the robot always reaches the desired waypoints successfully without crashing with the pedestrian and the road cones that were added to make the foot path more complex. Moreover, Figure 15 presents the evolution of two speeds (forward and angular) of the mobile robot. For example, the first row shows that the mobile robot tracked the desired speed without overshoot. As soon as obstacles or turns start to appear, the NMPC system provides larger angular velocities to change the robot heading, as the bottom row presents. Once the robot is oriented toward the waypoint and there is free
space, the angular speed becomes approximately zero. The obtained results demonstrate the effectiveness of our approach in terms of obstacle avoidance and autonomous navigation.

Figure 13. Zero-shot sim-to-real transfer: The segmentation model was deployed into a real robot. The robot performs autonomous navigation in real-world settings. (a) Examples of segmentation results on a real sidewalk. (b) Test site used for real-world experiments.

Figure 14. The followed path by the robot while avoiding obstacles in real-world settings.
5. Conclusions

In this paper, we presented a vision-based NMPC that enables a mobile delivery robot to autonomously navigate cluttered environments, deal with dynamic obstacles, and reach specified goal points to perform delivery missions. In contrast to the existing methods, the problem is formulated as an optimization problem that considers collision-free space and kinematic constraints over a specified preceding horizon. To achieve collision avoidance, a semantic segmentation deep learning model was created and trained from a synthetic dataset collected from a simulated dynamic sidewalk environment that was built using Unreal Engine software. The proposed segmentation model involves a transfer learning technique to reduce training time and increase sim-to-real transferability. The estimated drivable free space is used with a relaxed barrier function in the NMPC formulation without increasing the computational complexity of the problem. The delivery robot could automatically discover complex motions and avoid cluttered scenes with moving obstacles. In addition, real-world tests verified that the proposed algorithm can achieve zero-shot sim-to-real transfer. The proposed vision-based NMPC improves the safety and autonomy of the last-mile delivery robot.

Supplementary Materials: The supplementary videos for the actual tests can be accessed at https://youtu.be/_DCzwDDRok0?list=PLF6aV0nw8UMyqvBiYEGdUR6tMVQ9oqMNF.

Author Contributions: Conceptualization, O.D. and M.I.; methodology, O.D. and M.I.; software, O.D. and M.I.; validation, O.D. and D.J.L.; formal analysis, O.D.; investigation, O.D. and M.I.; resources, O.D. and D.J.L.; data curation, O.D. and M.I.; writing—original draft preparation, M.I.; writing—review and editing, O.D.; visualization, O.D. and M.I.; supervision, O.D. and D.J.L.; Project administration, J.c.K. and Y.J.K.; funding acquisition, D.J.L. All authors have read and agreed to the published version of the manuscript.
Funding: This work was supported by the Unmanned Vehicles Core Technology Research and Development Program through the National Research Foundation of Korea (NRF), Unmanned Vehicle Advanced Research Center (UVARC) funded by the Ministry of Science and ICT, the Republic of Korea (2020M3C1C1A02084772). It was also supported by the Technology Innovation Program (or Industrial Strategic Technology Development Program-Development and Demonstration of Automatic Driving Collaboration Platform to Overcome the Dangerous Environment of Databased Commercial Special Vehicles) (20018904, Development and Field Test of Mobile Edge for Safety of Special Purpose Vehicles in Hazardous Environment) funded By the Ministry of Trade, Industry and Energy (MOTIE, Korea).

Institutional Review Board Statement: Not applicable.
Informed Consent Statement: Not applicable.
Data Availability Statement: Not applicable.
Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations
The following abbreviations are used in this manuscript:

- NMPC: nonlinear model predictive control
- SADR: sidewalk autonomous delivery robot
- DWA: dynamic window approach
- IMU: inertial measurement unit
- RL: reinforcement Learning
- CNN: convolution neural networks

References
1. Arnold, R.D.; Yamaguchi, H.; Tanaka, T. Search and rescue with autonomous flying robots through behavior-based cooperative intelligence. *J. Int. Humanit. Action* 2018, 3, 1–18. [CrossRef]
2. Kaur, T.; Kumar, D. Wireless multifunctional robot for military applications. In Proceedings of the 2015 2nd International Conference on Recent Advances in Engineering & Computational Sciences (RAECS), Chandigarh, India, 21–22 December 2015; pp. 1–5.
3. Petillot, Y.R.; Antonelli, G.; Casalino, G.; Ferreira, F. Underwater robots: From remotely operated vehicles to intervention-autonomous underwater vehicles. *IEEE Robot. Autom. Mag.* 2019, 26, 94–101. [CrossRef]
4. Gao, K.; Xin, J.; Cheng, H.; Liu, D.; Li, J. Multi-mobile robot autonomous navigation system for intelligent logistics. In Proceedings of the 2018 Chinese Automation Congress (CAC), Piscataway, NJ, USA, 30 November–2 December 2018; pp. 2603–2609.
5. Jennings, D.; Figliozzi, M. Study of sidewalk autonomous delivery robots and their potential impacts on freight efficiency and travel. *Transp. Res. Rec.* 2019, 2673, 317–326. [CrossRef]
6. Wu, Y.; Li, Y.; Li, W.; Li, H.; Lu, R. Robust lidar-based localization scheme for unmanned ground vehicle via multisensor fusion. *IEEE Trans. Neural Netw. Learn. Syst.* 2020, 32, 5633–5643. [CrossRef] [PubMed]
7. Li, J.; Qin, H.; Wang, J.; Li, J. Openstreetmap-based autonomous navigation for the four wheel-legged robot via 3d-lidar and ccd camera. *IEEE Trans. Ind. Electron.* 2021, 69, 2708–2717. [CrossRef]
8. Cheng, Y.; Wang, G.Y. Mobile robot navigation based on lidar. In Proceedings of the 2018 Chinese Control And Decision Conference (CCDC), Shenyang, China, 9–11 June 2018; pp. 1243–1246.
9. Li, J.; Ran, M.; Xie, L. Efficient trajectory planning for multiple non-holonomic mobile robots via prioritized trajectory optimization. *IEEE Robot. Autom. Lett.* 2020, 6, 405–412. [CrossRef]
10. Muhammad, A.; Ali, M.A.; Shano, I.H. A novel algorithm for mobile robot path planning. In Proceedings of the 2021 IEEE 11th IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE), Penang, Malaysia, 3–4 April 2021; pp. 48–52.
11. Wang, N.; Xu, H. Dynamics-constrained global-local hybrid path planning of an autonomous surface vehicle. *IEEE Trans. Veh.* 2020, 69, 6928–6942. [CrossRef]
12. Karsli, M.; Satılım, Y.; Şara, M.; Tufan, F.; Eken, S.; Sayar, A. End-to-end learning model design for steering autonomous vehicle. In Proceedings of the 2018 26th Signal Processing and Communications Applications Conference (SIU), Izmir, Turkey, 2–5 May 2018; pp. 1–4.
13. Xiao, Y.; Codevilla, F.; Gurram, A.; Urfalioğlu, O.; López, A.M. Multimodal end-to-end autonomous driving. *IEEE Trans. Intell. Transp. Syst.* 2020, 23, 537–547. [CrossRef]
14. Devo, A.; Mezzetti, G.; Costante, G.; Fravolini, M.L.; Valigi, P. Towards generalization in target-driven visual navigation by using deep reinforcement learning. *IEEE Trans. Robot.* 2020, 36, 1546–1561. [CrossRef]
15. Gupta, S.; Davidson, J.; Levine, S.; Sukthankar, R.; Malik, J. Cognitive mapping and planning for visual navigation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2616–2625.

16. Shao, K.; Zhao, D.; Zhu, Y.; Zhang, Q. Visual navigation with actor-critic deep reinforcement learning. In Proceedings of the 2018 International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, Brazil, 8–13 July 2018; pp. 1–6.

17. Smolyanskiy, N.; Kamenev, A.; Smith, J.; Birchfield, S. Toward low-flying autonomous mav trail navigation using deep neural networks for environmental awareness. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017; pp. 4241–4247.

18. Finn, C.; Levine, S. Deep visual foresight for planning robot motion. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 2786–2793.

19. James, S.; Johns, E. 3d simulation for robot arm control with deep q-learning. arXiv. 2016, arXiv:1609.03759.

20. Peng, X.B.; Andrychowicz, M.; Zaremba, W.; Abbeel, P. Sim-to-real transfer of robotic control with dynamics randomization. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21–25 May 2018; pp. 3803–3810.

21. Tobin, J.; Fong, R.; Ray, A.; Schneider, J.; Zaremba, W.; Abbeel, P. Domain randomization for transferring deep neural networks from simulation to the real world. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017; pp. 23–30.

22. Sanders, A. An Introduction to Unreal Engine 4; AK Peters/CRC Press: Boca Raton, FL, USA, 2016.

23. Grüne, L.; Pannek, J. Nonlinear model predictive control. in Nonlinear Model Predictive Control; Springer: Berlin/Heidelberg, Germany, 2017; pp. 45–69.

24. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.-C. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 4510–4520.

25. Valenti, F.; Giaquinto, D.; Musto, L.; Zinelli, A.; Bertozzi, M.; Broggi, A. Enabling computer vision-based autonomous navigation for unmanned aerial vehicles in cluttered gps-denied environments. In Proceedings of the 2018 21st International Conference on Intelligent Transportation Systems (ITSC), Maui, HI, USA, 4–7 November 2018; pp. 3886–3891.

26. Vurtulu, S.C.; Şahin, Y.H.; Unal, G. Semantic segmentation with extended deeplabv3 architecture. In Proceedings of the 2019 27th Signal Processing and Communications Applications Conference (SIU), Sivas, Turkey, 24–26 April 2019; pp. 1–4.

27. Yan, L.; Liu, D.; Xiang, Q.; Luo, Y.; Wang, T.; Wu, D.; Chen, H.; Zhang, Y.; Li, Q. Psp net-based automatic segmentation network model for prostate magnetic resonance imaging. Comput. Methods Programs Biomed. 2021, 207, 106211. [CrossRef] [PubMed]

28. Villa, M.; Dardenne, G.; Nasan, M.; Letissier, H.; Hamitouche, C.; Stindel, E. Fcn-based approach for the automatic segmentation of bone surfaces in ultrasound images. Int. J. Comput. Assist. Surg. 2018, 13, 1707–1716. [CrossRef] [PubMed]

29. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Munich, Germany, 5–9 October 2015; pp. 234–241.

30. Ejaz, M.M.; Tang, T.B.; Lu, C.-K. Vision-based autonomous navigation approach for a tracked robot using deep reinforcement learning. IEEE Sensors J. 2020, 21, 2230–2240. [CrossRef]

31. Zaidner, G.; Shapiro, A. A novel data fusion algorithm for low-cost localisation and navigation of autonomous vineyard sprayer robots. Biosyst. Eng. 2016, 146, 133–148. [CrossRef]

32. Cai, J.; Ruan, X.; Yu, N. Autonomous navigation of mobile robot based on cognitive development. Comput. Eng. 2018, 44, 9–16.

33. Cherroun, L.; Boumehraz, M. Fuzzy logic and reinforcement learning based approaches for mobile robot navigation in unknown environment. Mediterr. J. Meas. Control. 2013, 9, 109–117.

34. Xin, J.; Zhao, H.; Liu, D.; Li, M. Application of deep reinforcement learning in mobile robot path planning. In Proceedings of the 2017 Chinese Automation Congress (CAC), Jinan, China, 20–22 October 2017; pp. 7112–7116.

35. Harapanahalli, S.; Mahony, N.O.; Hernandez, G.V.; Campbell, S.; Riordan, D.; Walsh, J. Autonomous navigation of mobile robots in factory environment. Procedia Manuf. 2019, 38, 1524–1531. [CrossRef]

36. Rehman, N.U.; Kumar, K. Implementation of an autonomous path planning & obstacle avoidance ugv using slam. In Proceedings of the 2018 International Conference on Engineering and Emerging Technologies (ICEET), Lahore, Pakistan, 22–23 February 2018; pp. 1–5.

37. Song, K.-T.; Chiu, Y.-H.; Kang, L.-R.; Song, S.-H.; Yang, C.-A.; Lu, P.-C.; Ou, S.-Q. Navigation control design of a mobile robot by integrating obstacle avoidance and lidar slam. In Proceedings of the 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Miyazaki, Japan, 7–10 October 2018; pp. 1833–1838.

38. Kim, Y.-H.; Jang, J.-I.; Yun, S. End-to-end deep learning for autonomous navigation of mobile robot. In Proceedings of the 2018 IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, NV, USA, 12–14 January 2018; pp. 1–6.

39. Pfeiffer, M.; Schaebule, M.; Nieto, J.; Siegwart, R.; Cadena, C. From perception to decision: A data-driven approach to end-to-end motion planning for autonomous ground robots. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Miyazaki, Japan, 24 July 2017; pp. 1527–1533.

40. Zhang, Y.; Chen, H.; He, Y.; Ye, M.; Cai, X.; Zhang, D. Road segmentation for all-day outdoor robot navigation. Neurocomputing 2018, 314, 316–325. [CrossRef]
41. Aghi, D.; Cerrato, S.; Mazzia, V.; Chiaberge, M. Deep semantic segmentation at the edge for autonomous navigation in vineyard rows. In Proceedings of the 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Prague, Czech Republic, 27 September–1 October 2021; pp. 3421–3428.

42. Adachi, M.; Shatari, S.; Miyamoto, R. Visual navigation using a webcam based on semantic segmentation for indoor robots. In Proceedings of the 2019 15th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS), Sorrento-Naples, Italy, 26–29 November 2019; pp. 15–21.

43. Miyamoto, R.; Nakamura, Y.; Adachi, M.; Nakajima, T.; Ishida, H.; Kojima, K.; Aoki, R.; Oki, T.; Kobayashi, S. Vision-based road-following using results of semantic segmentation for autonomous navigation. In Proceedings of the 2019 IEEE 9th International Conference on Consumer Electronics (ICCE-Berlin), Berlin, Germany, 8–11 September 2019; pp. 174–179.

44. Lin, J.; Wang, W.-J.; Huang, S.-K.; Chen, H.-C. Learning based semantic segmentation for robot navigation in outdoor environment. In Proceedings of the 2017 Joint 17th World Congress of International Fuzzy Systems Association and 9th International Conference on Soft Computing and Intelligent Systems (IPSACISIS), Otsu, Japan, 27–30 June 2017; pp. 1–5.

45. Karakaya, S.; Ocak, H. A novel local motion planner: Navibug. *J. Intell. Robot. Syst.* **2020**, *100*, 987–1003. [CrossRef]