Design and Testing of an Autonomous Driving System for Student Competitions

Anonymous Author(s)
Affiliation
Address
`email`

Abstract

This paper details the development of an autonomous vehicle for the iDEAS Engineering Competition, utilizing advanced machine learning and image processing techniques for autonomous track navigation. It presents a theoretical analysis of modern autonomous driving technologies, including object detection and semantic segmentation, and compares modular and end-to-end architectures. The implementation on the Jetson Nano board demonstrates its capability to run two neural networks at 10 frames per second. Testing confirmed the vehicle's ability to meet competition requirements, such as stopping at traffic signs, avoiding obstacles, and performing parking maneuvers, showcasing the system's efficiency and reliability.

1 Introduction

In recent years, autonomous driving has emerged as a rapidly growing field of research with significant practical applications in transportation. This technology aims to develop transportation systems that operate without human intervention, enhancing both safety and efficiency. Autonomous driving systems rely on advanced image processing and object recognition technologies, utilizing methods based on artificial intelligence. Implementing such technology in an autonomous vehicle represents a crucial step towards a future with safer, more efficient, and more accessible transportation for everyone.

The complexity of autonomous driving lies in understanding and interpreting the surrounding environment, a task that is straightforward for humans but challenging for computers to replicate accurately. This problem can only be addressed through advanced AI and machine learning systems, which enable vehicles to process vast amounts of sensor data and make real-time decisions in a constantly changing environment.

The motivation for this project stems from a passion for cars and AI, further fueled by participation in competitions like the iDEAS Engineering Competition and the NXP Cup. These experiences solidified my interest in the field and provided the impetus to develop a functional and efficient autonomous driving system.

The objectives of this thesis are to design and implement an autonomous car capable of meeting the requirements of the iDEAS competition, navigating the track, and adhering to all constraints and rules. This project aims to exceed previous limitations and create an autonomous vehicle that performs optimally in the competition, demonstrating accumulated progress and knowledge.

Various methods and tools were employed to achieve these objectives, including training two neural networks to run on an NVIDIA Jetson Nano board. The training process utilized online solutions like Google Colaboratory[1] for access to powerful GPUs and continuous model validation on a personal laptop.

Additionally, the Unity[6] simulation environment was used to recreate the competition conditions, allowing for testing and optimizing the algorithms in a controlled setting. Advanced techniques such as the bird's eye view perspective were implemented to improve trajectory planning.

The developed autonomous system is modular, meaning each component, from perception to planning and control—operates independently and can be updated without affecting the rest of the system. This modular approach,

Submitted to 1st Best Romanian AI Thesis Awards (BRAIT 2024). Do not distribute.

combined with advanced techniques, enabled the creation of a robust and high-performing autonomous driving system that meets the competition's requirements.

Main contributions:

- Developed a modular autonomous driving system capable of performing complex tasks such as obstacle avoidance, traffic signal recognition, and parking maneuvers.
- Demonstrated the feasibility of running sophisticated neural networks for object detection and semantic segmentation on a resource-constrained platform like the Jetson Nano.
- Developed a pipeline for simultaneous camera calibration and perspective transformation, which simplifies navigation planning and can be seamlessly integrated into any C++ project.

2 Related Work

In the field of autonomous driving, significant strides have been made in developing systems capable of navigating complex environments. Previous research has explored various methodologies and technologies, with a strong focus on image processing, object detection, and semantic segmentation.

Yurtsever et al. classify autonomous driving systems based on connectivity and architectural design [14]. Systems can be categorized as either ego or connected. Ego systems integrate all autonomous functions within a single, self-sufficient vehicle, which is prevalent due to lower costs and simplicity. Connected systems enable communication between vehicles, infrastructure, and other smart devices using technologies such as V2X, V2V, V2I, and V2P, enhancing overall safety and efficiency. However, this technology requires extensive infrastructure and is still in the conceptual stage [13].

Regarding architectural design, autonomous systems are typically modular or end-to-end. Modular systems divide functions into distinct components, such as perception, trajectory planning, and vehicle control, allowing greater flexibility and scalability. In contrast, end-to-end systems generate control actions directly from sensor data, learning to perceive and act similarly to humans, simplifying the overall process through advanced neural networks[9],[10].

Despite advancements, there is a noticeable gap in implementing state-of-the-art techniques like object detection and semantic segmentation on resource-constrained platforms like the Jetson Nano. Most studies focus on more powerful hardware, limiting practical deployment in real-world applications.

This paper addresses this gap by demonstrating the feasibility of running sophisticated neural networks for object detection and semantic segmentation on the Jetson Nano. The project showcases a robust solution for autonomous navigation, capable of performing tasks such as obstacle avoidance, traffic signal recognition, and parking maneuvers in a simulated environment.

In summary, while significant progress has been made in autonomous driving, this research focuses on practical implementation of advanced AI techniques on limited hardware, extending the applicability of these technologies and providing a foundation for future developments in accessible and efficient autonomous systems.

3 Approach

A modular system architecture offers significant advantages in autonomous driving due to its flexibility, scalability, and ease of development. Unlike end-to-end systems, which are difficult to modify without impacting the entire system, modular systems allow for better control over vehicle functions and facilitate specific hard coded solutions for tasks such as stopping at signs or obstacle avoidance. This approach also enhances the scalability and robustness of the system by allowing independent testing and maintenance of components. This rationale underpins the choice of a modular approach for this project, ensuring precise control and targeted optimizations.

3.1 Hardware

For the implementation and testing of the application, the hardware system used consists of two main components: a laptop and a NVIDIA Jetson Nano board. These are interconnected via a local network, enabling efficient and fast communication. The laptop transmits video streams to the Jetson Nano, while the Jetson Nano sends back control commands.

3.2 Simulator

Testing the system is done through a simulation in Unity[6], recreating a virtual track since access to the real track is only available during the competition. This simulation allows for evaluating functionality and performance in a controlled, reproducible environment. Unity provides a flexible platform for developing and testing 3D applications, with facilities for creating realistic physics and visuals.

Models of the track, traffic signs, and traffic lights were created in Blender[5] and imported into a predefined Unity scene, configured with lighting and visual settings. This setup ensures a realistic and detailed simulation environment for thorough testing.

3.3 Software

The application is implemented in C++, using the jetson-inference[4] library to facilitate neural network inference on the Jetson platform. TensorRT is utilized to optimize and accelerate neural network performance, ensuring real-time data processing essential for autonomous systems.

The perception module is responsible for collecting and interpreting environmental data, functioning as an interface for two neural networks. It pre-processes raw images from the camera by resizing, normalizing, and adjusting them for inference. Post-inference, it filters and interprets the outputs, extracting relevant information and discarding redundant or noisy data. Accurate detection of traffic signs, signals, and road surfaces is vital for autonomous driving. YOLOv3 tiny[12] is used for object detection, chosen for its balance of performance and efficiency. FastSCNN[11] is selected for road surface segmentation due to its maturity and community support, known to perform well on a Jetson Nano board[3]. The bird's eye view (BEV) perspective, transforming images to a top-down view, simplifies trajectory planning by making key elements like lanes and obstacles easier to identify and process.

To enhance system efficiency, a repository based on James Liao's work[2] was developed. This repository optimizes the process by combining camera calibration and inverse perspective mapping (IPM), correcting lens distortions and obtaining the BEV representation of the image simultaneously. It includes two applications: one for manual camera calibration using sliders to estimate distortion coefficients, and another for perspective transformation and mapping array aggregation. This streamlined approach improves significantly pre-processing time and overall system performance.

The planning module serves as the vehicle's decision-making core, using a state machine to manage transitions between actions such as stopping, waiting, moving, and parking based on perception data. It plans the vehicle's path, ensuring safe and efficient navigation.

The control module translates planning data into speed and steering commands, transmitted via a UDP socket to the Unity car controller. Initially intended to implement closed-loop controllers for longitudinal and lateral control, time constraints led to the decision to handle only reference command transmission.

3.4 Training the networks

To train the YOLOv3-tiny network, Google Colaboratory was used, leveraging its free GPU and TPU resources, which are crucial for deep learning tasks. The training process was based on a tutorial from DepthAI and adapted for this project's requirements[8]. The dataset for object detection includes seven classes, with images sourced from real-world data (30% for traffic signs, 55% for traffic lights) and synthetic images generated using Blender and a simulator (70% for traffic signs, 45% for traffic lights).

The segmentation network was trained on a personal laptop using PaddleSeg[7]. The dataset consists of bird's eye view images labeled into six classes, with a resolution of 1024x512 pixels. This dataset is composed of 40% real images and 60% synthetic images, totaling 3866 images. The real images, segmented during competitions, were essential for improving the model's generalization. Both datasets ensure the models can perform well under various conditions, balancing real and simulated data for robust training.

4 Results

The main goal of system testing is to verify the autonomous application's functionality and performance in conditions similar to the competition. This involves evaluating the accuracy of maneuvers, system response to different traffic scenarios, and overall stability and reliability.

The iDEAS Engineering Competition challenges students to design autonomous cars that navigate an eight shaped track, recognizing traffic signs, obeying traffic lights, and parking autonomously. The track includes seven traffic signs, two traffic lights, and a wireless charging station, testing the cars' ability to handle various real-world driving scenarios.

The vehicle was tested through multiple laps on the track to ensure it met the competition criteria, including stopping at red and yellow lights, halting at stop signs, parking correctly, and avoiding obstacles. In Unity, logic was implemented to verify if the vehicle responded correctly near signs or lights, counting the frequency and type of errors.

For the PARK sign, a 5-second stop was implemented instead of a complete stop to facilitate repeated lap testing. A total of 132 laps were conducted to evaluate the vehicle's performance, with the results summarized below.

Table 1: Lap results

Metric	**Successes**	**Fails**
Number of completed laps	131	1
Number of completed laps with no collisions	132	0
Number of stops at STOP sign (>3 sec)	132	0
Number of stops at CROSS sign (>1 sec)	132	0
Number of stops at PARK sign (>5 sec)	132	0
Number of parking maneuvers performed	125	7
Number of regulatory crossings of the intersection	115	17

The vehicle successfully completed 131 out of 132 laps, effectively avoiding obstacles and correctly stopping at the STOP, CROSS, and PARK signs. There were 7 parking failures primarily due to issues with detecting the charging station sign, suggesting the need for a more robust detection model that could reduce the need for extensive filtering, thereby enhancing overall performance.

For the traffic light intersection, there were 17 failures, mostly during the green-to-yellow transition, resulting in one incomplete lap due to the car getting stuck in the intersection. This indicates the necessity for better detection and response mechanisms for traffic light transitions. The limited field of view and the current implementation contribute to these challenges.

Regarding segmentation, the system generally performed well, reliably identifying objects. However, in intersections without white lane markings, the segmentation accuracy decreased. Additionally, obstacle segmentation was not optimal but sufficient for the tasks.

Overall, the system demonstrated reliable performance, with several areas identified for enhancement, including expanding the field of view, optimizing detection algorithms, and utilizing a more diverse segmentation dataset to address decreased accuracy in specific scenarios.

5 Conclusion

In conclusion, this project successfully developed and tested an autonomous vehicle system tailored for the iDEAS Engineering Competition. The main findings demonstrate the system's capability to perform key tasks such as stopping at traffic signals, avoiding obstacles, and executing parking maneuvers with a high degree of accuracy. The use of NVIDIA Jetson Nano for running neural networks in real-time proved effective, showcasing its potential for AI applications despite its lower computational power compared to other platforms.

The project's significance lies in advancing the practical implementation of AI-driven autonomous systems, showcasing the feasibility and performance of modular architectures. The experimental results underscore the system's reliability and robustness in simulated conditions, offering valuable insights into the challenges and solutions in autonomous driving.

Future work should focus on real-world testing to further validate the system's performance in actual competition scenarios. Additionally, exploring advanced techniques such as simultaneous localization and mapping (SLAM) and more powerful hardware platforms could enhance the system's adaptability and precision. This project contributes by offering a solid foundation for developing autonomous vehicles, providing a starting point for further advancements and refinements in the field.

References

[1] Google colaboratory. `https://colab.research.google.com/`. Accessed: 24.06.2024.

[2] Ipm-master. `https://github.com/JamesLiao714/IPM-master`. Accessed: 24.06.2024.

[3] Jetson nano tensorrt fast-scnn 576x576 fp16. https://www.youtube.com/watch?v=Qy4S4_5JCAM, . Accessed: 24.06.2024.

[4] jetson-inference. https://github.com/dusty-nv/jetson-inference, . Accessed: 24.06.2024.

[5] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018. URL http://www.blender.org.

[6] Arthur Juliani, Vincent-Pierre Berges, Ervin Teng, Andrew Cohen, Jonathan Harper, Chris Elion, Chris Goy, Yuan Gao, Hunter Henry, Marwan Mattar, and Danny Lange. Unity: A general platform for intelligent agents, 2020. URL https://arxiv.org/abs/1809.02627.

[7] Yi Liu, Lutao Chu, Guowei Chen, Zewu Wu, Zeyu Chen, Baohua Lai, and Yuying Hao. Paddleseg: A high-efficient development toolkit for image segmentation, 2021.

[8] Luxonis. Yolov3-tiny training tutorial. https://github.com/luxonis/depthai-ml-training/blob/master/colab-notebooks/YoloV3_V4_tiny_training.ipynb. Ultima accesare: 26.06.2024.

[9] U Muller, J Ben, E Cosatto, B Fleep, and Y LeCun. Autonomous off-road vehicle control using end-to-end learning. *Courant Institute/CBLL, Arlington, VA, USA, Tech. Rep. DARPA-IPTO Final technical Report*, 458, 2004.

[10] Dean A Pomerleau. Alvinn: An autonomous land vehicle in a neural network. *Advances in neural information processing systems*, 1, 1988.

[11] Rudra P K Poudel, Stephan Liwicki, and Roberto Cipolla. Fast-scnn: Fast semantic segmentation network, 2019. URL https://arxiv.org/abs/1902.04502.

[12] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement, 2018. URL https://arxiv.org/abs/1804.02767.

[13] Muhammad Abdul Rehman, Muhammad Numan, Hassaan Tahir, Usama Rahman, Muhammad Waseem Khan, and Muhammad Zubair Iftikhar. A comprehensive overview of vehicle to everything (v2x) technology for sustainable ev adoption. *Journal of Energy Storage*, 74:109304, 2023. ISSN 2352-152X. doi: https://doi.org/10.1016/j.est.2023.109304. URL https://www.sciencedirect.com/science/article/pii/S2352152X23027020.

[14] Ekim Yurtsever, Jacob Lambert, Alexander Carballo, and Kazuya Takeda. A survey of autonomous driving: Common practices and emerging technologies. *IEEE Access*, 8:58443–58469, 2020. doi: 10.1109/ACCESS.2020.2983149.