

Self Driving Car Final Report

By Tommy Priest, Michal Wasowski and Thom Russell

Supervised by Muharrem Maloku

Abstract	1
Background and Aims	1
Project Goals:	2
Methods	2
Results	3
Discussion	4
Conclusion and Future Work	8
Lessons Learned	9
Group Reflection	10
References	11

Abstract

This project was focused on creating an affordable self-operating vehicle. The idea behind it is to develop an inexpensive educational tool for providing students with a basic understanding of robotics and programming principles and giving them an opportunity to gain useful experience in these fields.

The team's research concerned the area of simplifying the technology of the DonkeyCar software platform [1] and the Waveshare Pi Racer as a hardware base [2], calculating the most suitable computer vision models needed for the technology to function, and designing a system for use by people without any background in technology.

The project involved an iterative development process, beginning with gathering equipment and, later on, installing software and collecting data for machine learning. The vehicle was developed using open-source technologies, including Python, the DonkeyCar framework and machine learning tools based on TensorFlow. The team used various tools for effective communication and project management when working on the project.

The project proved the effectiveness of low-cost and autonomous vehicle platforms in the educational field. Repeated tests and iterations of solution delivery helped the team define major aspects of building and training small autonomous vehicles. The project lays the foundation for future developments and educational projects in the field of artificial intelligence.

Background and Aims

The idea for this project was proposed to us by a staff member in the department, who suggested DonkeyCar, also known as Pi Racer, as a suitable open-source platform to build around. Taking that suggestion as a starting point, it was down to us to decide what to actually do with it and to shape it into something with a clear purpose beyond the bursary period itself.

Artificial intelligence, robotics, and autonomous systems are increasingly playing an important role in different domains. Despite this fact, students usually find it difficult to work with such technologies because of the price and complexity of the existing machines and software. With the help of initiatives like DonkeyCar, it is possible to solve these problems through the use of inexpensive autonomous driving platforms.

The goal of this project was to determine whether an inexpensive autonomous vehicle can be transformed into a teaching instrument that can be used by students of different fields. The aim was to improve on the available open-source solutions, make building and using DIY platforms easier, and incorporate computer vision technologies and machine learning into the autonomous driving process. We settled on framing the project explicitly around outreach and education, which is reflected in the working title under which the project was submitted: Building and Enhancing a DIY Autonomous Vehicle Platform for Learning and Outreach. It was also important to develop an interdisciplinary framework for education, inclusive of different learning processes and forms of education.

The main goal of this project was to create a low-cost autonomous driving platform suitable for education.

Project Goals:

- Creating a working Pi Racer from affordable and commonly available equipment.
- Finding out what machine learning and computer vision solutions could be used in the work.
- Preparing effective documentation for reproduction and use of the product.
- Figuring out how to make the product usable for those who don't have any technical background.
- Providing opportunities for users to be engaged in further learning processes.

Methods

The methodology used in this project corresponds to the Agile development approach, which breaks down the work process into a number of iterations, such as assembly of the hardware, configuration of the software, data acquisition, development of a machine learning model, testing and improvement. This way, improvements and modifications were constantly being made while executing the project, as soon as certain issues appeared.

The Pi Racer [2] was assembled with the help of the DonkeyCar open-source platform [1] and a Raspberry Pi, camera module, and other easy-to-afford hardware. Python and the DonkeyCar framework were used to set up the software environment, while TensorFlow was applied to help train the computer vision models. Driving data were obtained by manually driving the remote-controlled vehicle around an oval training track, and this data was used to train the models anticipating steering commands for an autonomous vehicle. The track itself was a simple layout with no complex sequences of turns, but its corners were fairly sharp, so accurately predicting steering angle through these corners was the main technical challenge rather than the complexity of the route.

The vehicle was performance tested many times, evaluating parameters such as navigation accuracy and precision, responsiveness, reliability, and usability. Various hardware and software configurations were tested, when needed, in order to enhance performance and simplify implementation.

Beyond proposing the initial concept, most of the practical work, assembling the hardware, setting up the software environment, calibrating the vehicle, collecting driving data and training the model, was carried out independently by the three of us, and we returned to the staff member who had suggested the platform mainly when we hit specific hardware issues we couldn't resolve on our own.

The project was otherwise carried out in a collaborative manner between the three of us, with the help of Microsoft Teams, GitHub and joint documentation for development coordination and

progress monitoring. No ethical approval was needed as no human participants took part in the study and no private data was collected.

Results

The project resulted in the successful construction and configuration of a small-scale autonomous vehicle, the Pi Racer, based on the DonkeyCar platform. The vehicle was assembled using a Raspberry Pi 4, camera module, RC car chassis, motor controller and servo system. The required software environment was installed and configured, allowing the vehicle to collect camera data and receive control commands.

During the initial configuration process, the vehicle required calibration of PWM values for several parameters, including minimum and maximum motor speed, left and right steering angles, and reverse movement. After calibration, the vehicle was able to respond correctly to manual control commands using a gamepad.

The first training dataset was collected by manually driving the vehicle around the oval track for approximately 15 minutes. After training the machine learning model for approximately 20 minutes on a separate computer, the vehicle was able to perform autonomous driving on the track. However, the initial model mainly learned to perform turns in one direction and showed limited performance when approaching turns in the opposite direction, struggling in particular to predict a sharp enough steering angle through the corners. An additional approximately 20 minutes of driving data was collected, including examples of steering in the opposite direction. After retraining the model with the expanded dataset, the vehicle was able to perform turns in both directions and maintain continuous autonomous movement on the predefined track.

During testing, the final model successfully completed autonomous driving without continuous human control. The vehicle was able to remain on the track and correct its direction based on camera input. Earlier testing stages showed occasional problems, including insufficient steering corrections and the vehicle leaving the track. After increasing the amount and variety of training data, these issues were reduced.

The project also identified several technical challenges during development. The longest configuration tasks involved installing the DonkeyCar environment, setting up required software dependencies on the Raspberry Pi, calibrating hardware parameters and resolving issues with reverse movement control.

Discussion

The results of this project demonstrate that a low-cost autonomous vehicle platform can be successfully developed and used as an educational tool for introducing students to artificial intelligence, robotics and computer vision. The main aim of the project was to evaluate whether an affordable DonkeyCar-based platform could provide a practical learning experience, and the results indicate that this aim was achieved.

One of the most important findings of the project was the impact of training data quality and diversity on autonomous driving performance. The initial model was trained using approximately 15 minutes of manually collected driving data. Although this allowed the vehicle to navigate the track autonomously, the model mainly learned steering behaviour for one direction. This showed that collecting data from only a limited range of driving situations can negatively affect the ability of the model to generalise. After additional data was collected, including turns in the opposite direction, model performance improved significantly and the vehicle was able to maintain autonomous movement in both directions. This demonstrates the importance of creating balanced and representative datasets when developing machine learning-based systems.

A related challenge was that the hardware itself made the Pi Racer difficult to drive smoothly by hand, particularly around the sharp corners of the oval track. The chassis was sensitive to small steering inputs, and it took considerable practice to hit a consistent turning angle rather than overcorrecting or cutting corners too early or too late. Plotting the recorded driving input (`user_angle` and `user_throttle`) against the model's predicted output (`pilot_angle` and `pilot_throttle`) made this visible directly: predictions followed the general driving pattern well on the straights but picked up much of the noise present in the underlying manual input around the corners, where steering angle was least consistent. This meant that a single short driving session was rarely enough: a much larger volume of training data, gathered across multiple sessions and covering a wider range of steering situations, was needed before the model's predictions began to track the intended driving line consistently rather than reproducing noise in the training set.

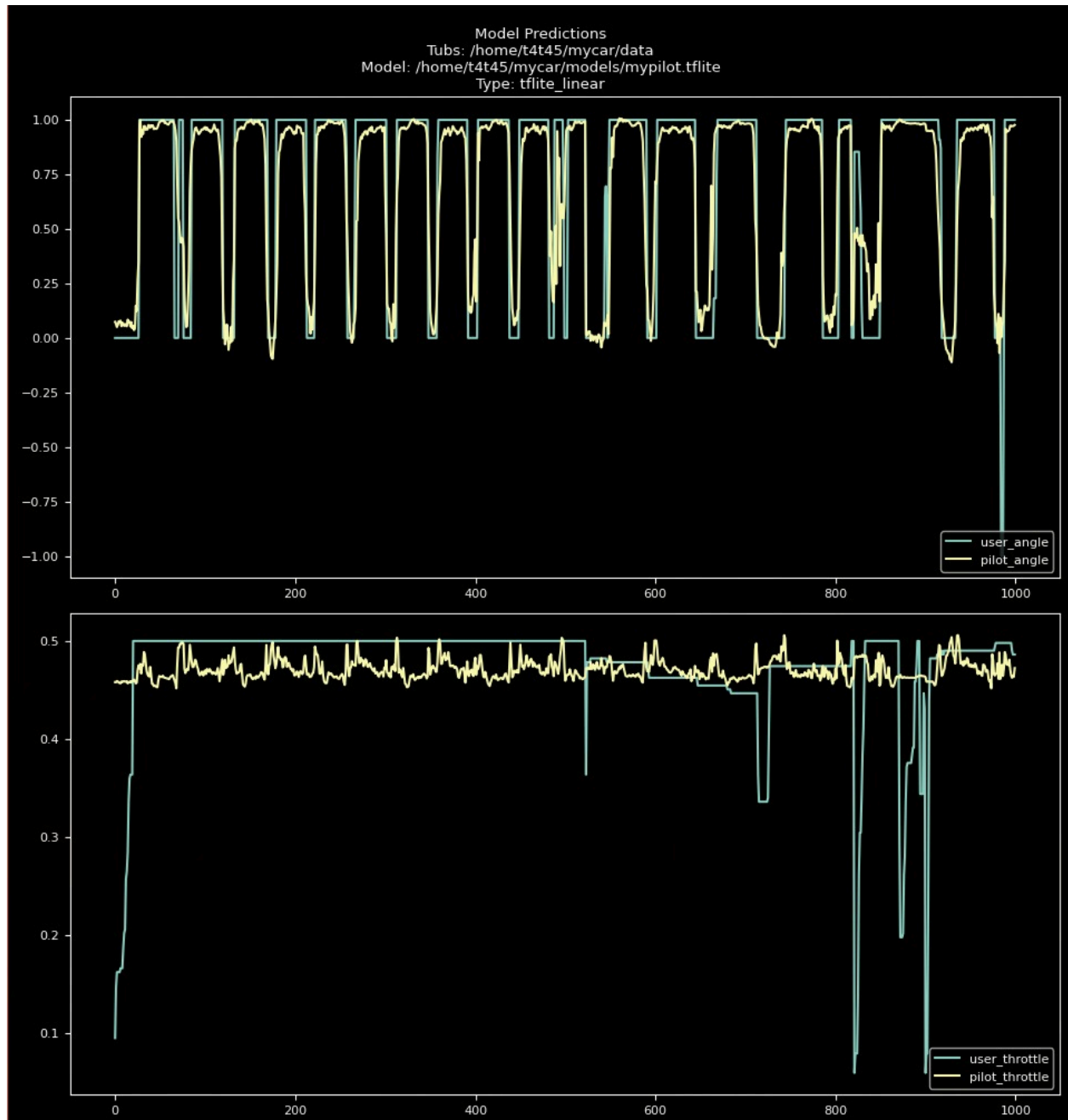


Figure 1: Model predictions, TensorFlow Lite (tflite_linear) export.

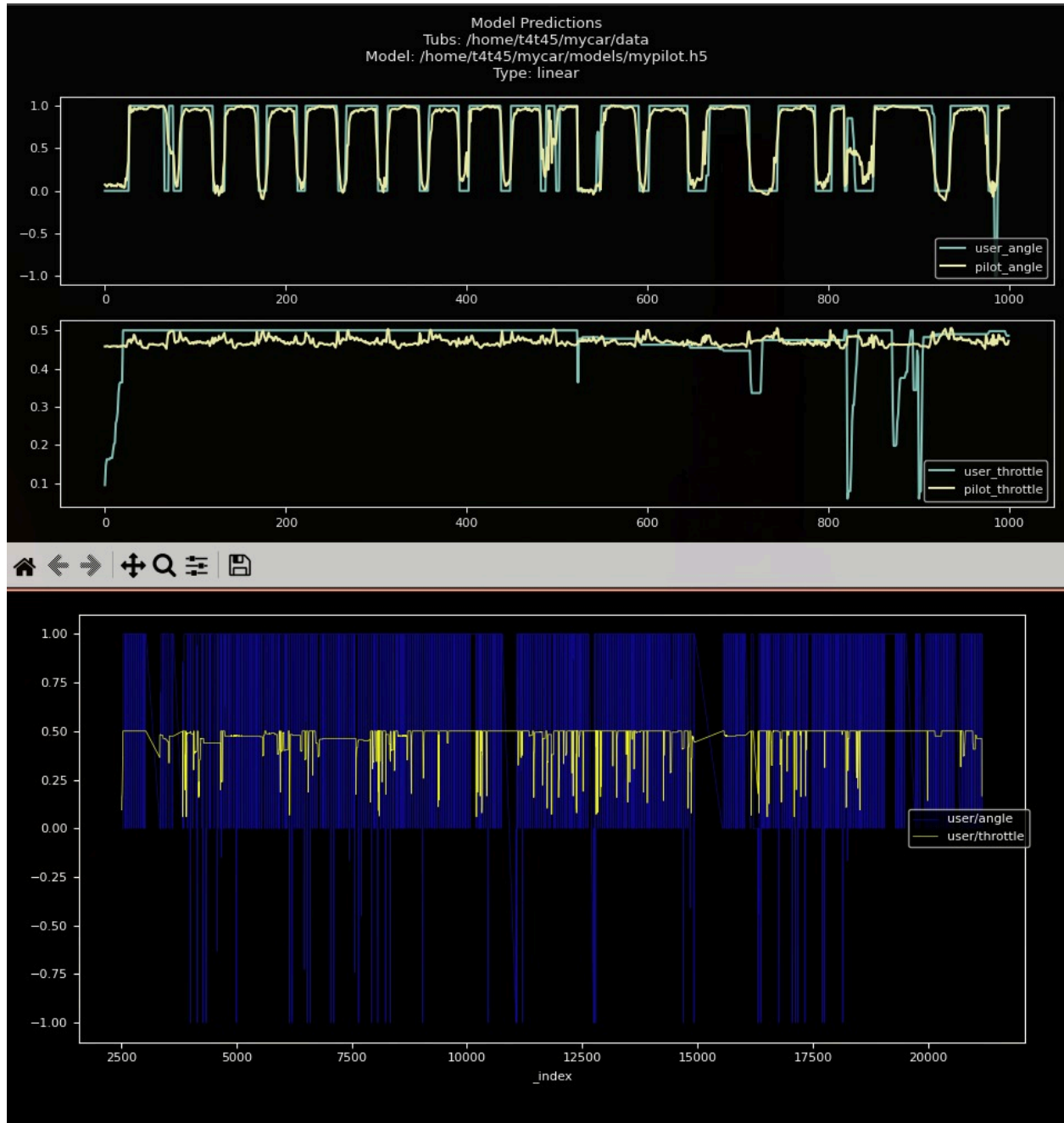


Figure 2: Model predictions, Keras (.h5, linear) export, with raw driving data below.

Comparing the two exports of the trained model, both the TensorFlow Lite and Keras versions closely tracked the general shape of the user's steering input, correctly identifying straight sections and turns, but showed the largest deviations during rapid direction changes, which lines up with the sections where manual driving input was itself least consistent. This reinforced our conclusion that the quality of the driving data, not just its quantity, had a direct and visible effect on how closely the trained model was able to reproduce the intended driving behaviour.

The project also highlighted that building an autonomous vehicle involves more than only training an artificial intelligence model. A significant amount of time was required for hardware configuration, software installation and system calibration. In particular, adjusting PWM values for steering, speed and reverse movement was necessary to achieve reliable vehicle control. The installation of the DonkeyCar environment on the Raspberry Pi also required careful configuration of multiple software dependencies. These challenges showed that successful implementation of autonomous systems requires both programming knowledge and practical understanding of hardware components.

The chosen Agile methodology was effective because it allowed the team to continuously test and improve the system. Instead of attempting to create a complete solution immediately, the vehicle was developed through smaller stages, so problems could be identified and solved during the development process. Regular teamwork, communication through Microsoft Teams and shared documentation helped maintain progress and coordinate tasks. However, the project also had several limitations. The vehicle was tested only on one predefined track, a simple oval circuit, so the model was never evaluated on more varied layouts or different environments. Although the track itself was not geometrically complex, its corners were fairly sharp, and this was enough to expose steering angle prediction as the main limiting factor rather than the route as a whole. The system was also developed using one main machine learning model, and further comparison with other models would be beneficial. Additionally, the vehicle was a small educational platform rather than a full-scale autonomous vehicle, meaning the results cannot be directly transferred to larger autonomous driving systems.

Despite these limitations, the project demonstrated the educational potential of low-cost autonomous platforms. The main value of the project is not only the creation of a working vehicle but also the demonstration of how students can interact with real AI systems through practical experimentation. Because the project was framed around outreach from an early stage, we expect its main impact to be on engagement rather than on course content directly: seeing a small car navigate a track autonomously tends to spark more curiosity in prospective students than an abstract description of AI would, and gives a natural, low-pressure way to talk about the underlying concepts, data, training, models, without assuming prior technical knowledge. With improved documentation and further testing, the platform could be used in workshops, open days and outreach activities to encourage students to explore artificial intelligence and engineering.

Conclusion and Future Work

This project showed that a low-cost autonomous vehicle based on the DonkeyCar platform can be successfully used as an educational tool for introducing students to artificial intelligence, robotics and computer vision. The team successfully developed a working autonomous vehicle and demonstrated that machine learning models can be trained using a relatively small amount of collected driving data.

The project also showed that successful autonomous driving depends not only on the model itself but also on proper hardware configuration, calibration and preparation of suitable training data. Because we ran into a number of setup and calibration problems that took time to work through, the documentation produced alongside this report should mean that future users, whether maintaining this car or building a similar platform of their own, do not have to start from zero.

Future work will focus on improving the accessibility and functionality of the platform. Planned improvements include creating a simplified setup and training guide, testing additional machine learning models, collecting larger datasets and exploring further computer vision applications. The platform could also be expanded with additional features and used in workshops, open days and educational demonstrations.

In addition, future versions of the project could investigate how the platform can support students with different levels of technical experience. By improving documentation and reducing the complexity of the initial setup, the system could become a more accessible resource for wider educational use.

Overall, the project provides a practical foundation for further exploration of autonomous systems and demonstrates the potential of affordable open-source platforms in engineering education.

Lessons Learned

This project provided the team with practical experience in developing a complete hardware and software system from the initial setup to a working prototype. One of the main lessons learned was the importance of understanding the full development process rather than focusing only on the final result. Small configuration details, such as hardware calibration, software dependencies and system settings, had a significant impact on the overall performance of the vehicle.

The team also learned that troubleshooting is an essential part of engineering projects. Several issues, including problems with the gamepad control, PWM calibration and the installation of the DonkeyCar environment, required testing different solutions before finding a working approach. These experiences improved the team's ability to analyse problems, search for solutions and work with unfamiliar technologies.

The project also gave us insight into how staff-student partnerships can work in practice. Because staff involvement was concentrated at the outset, the initial idea and direction, rather than ongoing throughout, we had to build much of our own process for resolving problems, relying on each other and on the wider DonkeyCar community documentation rather than routine staff guidance during the technical stages. Looking back, agreeing earlier on when to check in with the staff member who proposed the project would likely have saved time on some of the hardware troubleshooting.

Another important lesson was the importance of planning and documentation. Although the team successfully configured the platform, the initial setup process required significant time and effort. Preparing clearer instructions earlier in the project would have made the development process more efficient and would help future users avoid similar difficulties.

The project also improved the team's understanding of collaborative work. Regular meetings at the university, communication through Microsoft Teams and shared documentation allowed all members to contribute to different stages of development. Working together on hardware, software and testing helped the team develop better communication and problem-solving skills.

If the project were continued, the team would improve the planning stage by creating a clearer timeline for testing, documentation and evaluation, and would agree on fixed points to check in with staff rather than only reaching out once something had already gone wrong. More time would also be allocated for experimenting with different models and collecting additional training data.

Overall, the project provided valuable experience in practical engineering, teamwork and working with emerging technologies. It demonstrated that successful project development requires not only technical knowledge but also organisation, patience and continuous improvement.

Group Reflection

Working as a three-person team allowed us to approach the project in a practical and collaborative way. Most of the work was completed together during meetings at the university, where we assembled the vehicle, tested different configurations and discussed possible solutions to problems. Working together during these stages was useful because each team member could contribute ideas and help with troubleshooting instead of working separately on isolated tasks.

Communication was mainly managed through Microsoft Teams, which was used for organising project milestones, sharing updates and keeping track of progress. We also used shared Google Docs for writing documentation and preparing the final report, which allowed everyone to contribute and review the work. This helped us maintain a clear overview of the project and ensured that all members were involved in the main decisions.

One of the biggest challenges was that many technical issues appeared unexpectedly during development. Problems with software installation, PWM calibration and gamepad control often required additional testing and time to understand. Since this was our first experience working with an autonomous vehicle platform, some solutions were found through trial and error rather than immediately knowing the correct approach.

Staff involvement in the project was mostly at the outset: a staff member proposed the idea and pointed us toward the DonkeyCar platform, after which we largely worked independently, going back to them only when we hit specific hardware problems we couldn't solve ourselves. This gave us a slightly different experience of partnership than we might have expected, less a continuous exchange with an academic partner, and more staff providing a starting point and then trusting us to take ownership of the direction and execution. Both are legitimate ways for a partnership to work, and it's something we'd be more deliberate about agreeing on upfront if we did a project like this again, for example by fixing a few check-in points in advance rather than only reaching out once something had already gone wrong.

Looking back, we would improve the project organisation by preparing a clearer setup plan and documentation earlier in the development process. This would make troubleshooting easier and leave more time for testing additional ideas. Overall, the project was a valuable experience that combined technical development with teamwork, communication and problem-solving.

References

- [1] Autorope, "Donkey Car documentation," Donkey Car. [Online]. Available: <https://docs.donkeycar.com/>. [Accessed: Feb. 2, 2026].
- [2] Waveshare, "Pi Racer Pro AI Kit," [Online]. Available: <https://www.waveshare.com/piracer-pro-ai-kit.htm?sku=18492>. [Accessed: Feb. 2, 2026].