Proceedings of the of the 17th Conference on Computer Science and Intelligence Systems pp. 117–120

DOI: 10.15439/2022F89
ISSN 2300-5963 ACSIS, Vol. 30

Aspects of autonomous drive control using NVIDIA Jetson Nano microcomputer

Kacper Podbucki
Poznan University of Technology, ul. Jana Pawła II 24,
60-965 Poznań, Poland
Email: kacper.podbucki@put.poznan.pl

Tomasz Marciniak
Poznan University of Technology, ul. Jana Pawła II 24,
60-965 Poznań, Poland
Email: tomasz.marciniak@put.poznan.pl

***Abstract*— The article describes the training process and experiments regarding autonomous movement by the autonomous car Waveshare JetRacer AI. The central unit responsible for controlling the vehicle's systems, i.e. the steering servo and the DC motors used for the drive, is the NVIDIA Jetson Nano embedded device. The application of the IMX219 camera module for data acquisition and training of a neural network models on microcomputer and their use for the implementation of autonomous driving are described.**

I. Introduction

THE autonomous movement of vehicles is a task that requires a careful analysis of the environment. For this purpose, vision or sensory systems are used. They play the role of a source of data about the environment in specific assistance systems, e.g., active cruise control, emergency braking, environment mapping, digital side and rear view mirrors, parking assist and lane assist [1]-[3]. Due to the advancement of such systems and their continuous development, it is possible that they will soon allow the achievement of higher levels of driving automation, ultimately leading to the mass production of fully autonomous vehicles.

There are six levels of vehicle driving autonomy defined by SAE International in the SAE J3016 standard, which are numbered from 0 to 5 [4]:

- Level 0: No Automation
- Level 1: Driver Assistance
- Level 2: Partial Automation
- Level 3: Conditional Automation
- Level 4: High Automation
- Level 5: Full Automation.

At level 0 there is no automatic vehicle control, but the system may issue warnings. In the next stage, it is defined that the driver must be ready to take control of the vehicle at any time. The automated system may be equipped with features such as Adaptive Cruise Control (ACC), Parking Assistance with automated steering, and Lane Keeping Assistance (LKA) Type II in any combination. The second level of partial automation obliges the driver to detect objects and events and to be ready for response if automated system fails to react properly. A computer executes accelerating, braking, and steering, but can be immediately deactivated while driver takes over the control. Level 3 allows the driver to safely turn attention from driving tasks within known, limited environments (e.g. freeways). But if any warning or alert occurs, the driver must intervene. In the 4th level the vehicle is controlled by an automated system but only when the conditions in the environment allow the enabling system by the driver. When running, there is no need to pay attention to the driver while driving. The last level requires only to set the destination and press the start button. The automatic system will drive to any location where it is legal to drive [5].

Real-time processing of data received from vision modules or sensors and sending feedback control signals requires the use of computers with high computing power. Due to the dynamic development of embedded devices and increasing their efficiency, it is possible to use them for tasks that require real-time calculations [6]. Examples of such modules are microcomputers from the NVIDIA Jetson family. They are equipped with powerful GPUs that allow performing complex calculations using models of neural networks.

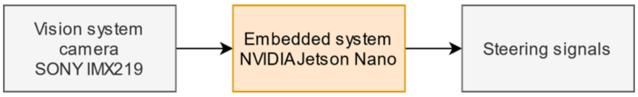

Fig. 1 Block diagram of the tested system

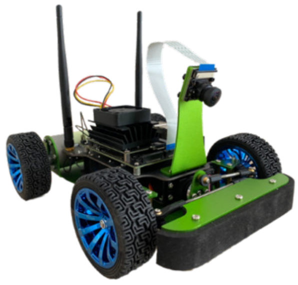

Fig. 2 Waveshare JetRacer AI with NVIDIA Jetson Nano microcomputer and Waveshare IMX219 camera

The purpose of processing the captured video sequences is to obtain parameters that allow precise control of the steering system of the vehicle (Fig. 1). The implementation of algorithms for autonomous driving in the model environment was carried out with the use of a four-wheeled Waveshare JetRacer AI mobile vehicle with a NVIDIA Jetson Nano

This research was funded partly by the 2022 subvention and partly with the SMART4ALL EU Horizon 2020 project, Grant Agreement No 872614.

minicomputer and a Waveshare IMX219 camera (Fig. 2). Various models of neural networks were trained and tested for the autonomous driving of a vehicle along a designated route. The software allows to stream the image from the car's on-board camera and enables the user to communicate with the environment by controlling the displayed widgets with the use of a mouse or an external controller.

II. Waveshare JetRacer AI hardware based on NVIDIA Jetson microcomputer

Waveshare JetRacer AI is a platform that allows to independently construct a four-wheeled mobile vehicle. The kit consists of mechanical elements of the chassis, bumpers, camera arm, and wheels. The steering system is controlled by the MG996R servo with a torque of 9 kg/cm. The drive is provided by two 37-520 DC motors with a reduction rate of 1:10 and idle speed of 740 RPM. The 12.6V power supply is provided by the use of 3 Li-Ion 18650 batteries connected in series. Wireless communication, which allows remote programming and car control, is carried out using the AC8265 WiFi module with two antennas. The main PCB is equipped with a 0.91" OLED display with a resolution of 128×32 pixels. It shows the parameters for the use of computer resources and the IP address used to establish wireless communication. The manufacturer also includes a universal gamepad that can be used to control the car [7].

The central unit responsible for controlling the steering and drive systems as well as processing data from the camera is the NVIDIA Jetson Nano B01 microcomputer. Its most important technical specifications [8] are presented in Table I. This microcomputer was designed mainly for use in tasks related to artificial intelligence. It allows running many neural networks, which can perform processes such as image classification or object detection. Their simultaneous operation while maintaining appropriate performance is ensured by the CUDA architecture, which allows for the performance of complex and computationally expensive operations, such as matrix calculations or 3D rendering using the potential of the CPU and GPU [9].

Table I.
NVIDIA Jetson Nano B01 specification

GPU	128-core Maxwell
CPU	Quad-core ARM A57 1.43 GHz
RAM	4 GB 64-bit LPDDR4
Camera	2×MIPI CSI-2 DPHY lanes
Connectivity	M.2 Key e.g. for network card
Operating System	Linux for Tegra – JetCard 4.5.1

Image recording is possible by equipping the vehicle with a Waveshare IMX219 camera module 160 degree FoV [10]. Its resolution is 8 megapixels and the lens is wide-angle 160 degrees. This shows the significant extension of the perspective in comparison to 62.2 degrees in the commonly used Raspberry Pi Camera v2. The Waveshare IMX219 camera is also characterized by an aperture of f2.35 and a focal length of 3.15mm. It allows to take pictures with a maximum resolution of 3280×2464 pixels and video recording with a resolution of 1080p and frequency of 30 frames per second. The camera module itself (without dedicated PCB) is compatible with both the Raspberry Pi Camera v2 PCB and the NVIDIA Jetson Nano minicomputer used by connecting via the CSI connector.

Waveshare JetRacer AI car programming can be performed through a wireless connection to the NVIDIA Jetson Nano microcomputer. For this purpose, the JupyterLab environment is used, which is installed by default in the software package provided by the manufacturer. The system also has implemented libraries necessary to support neural networks and video sequence processing operations, such as: OpenCV, Tensorflow, NumPy, PyTorch, or NVIDIA TensorRT [11].

III. Data collection for neural network training

The task for the Waveshare JetRacer AI vehicle is to follow an oval track, on which the edges are marked with yellow lines and the central axis with a white dotted line [12]. Based on the image from the camera, which is sent to the car's main computer, the neural network algorithm decides to correct the direction of the vehicle's movement by changing the position of the steering servo. In the presented study, movement at variable speed is not assumed. Its value is manually controlled by the program widget, but it is possible to extend the algorithm functionality of its autonomous selection in the future.

The implementation of the Waveshare JetRacer AI car autonomous movement algorithm consists of several stages. The first one is the hardware configuration of the device, i.e. the appropriate mechanical setting of the steering system and empirical determination of its correction coefficients and the parameters of the drive system. They depend on the design of a specific car. After establishing communication with the computer via the wireless network, it is possible to adjust them through the widgets included in the program. Then, in the next step, for convenience of use, it is worth connecting the wireless controller to the computer to be able to remotely control the vehicle. For this purpose, can be used the gamepad provided by the manufacturer or any other controller such as the Xbox 360 Controller.

In the next stage, the camera widget is initialized to show a preview on one screen and on the second one to precisely define the coordinates of the direction vector, according to which the car should move autonomously. These values can be set using the widget sliders, or after appropriate code modification with a wireless controller. The user interface and the actual vehicle position are shown in Figs. 3 and 4.

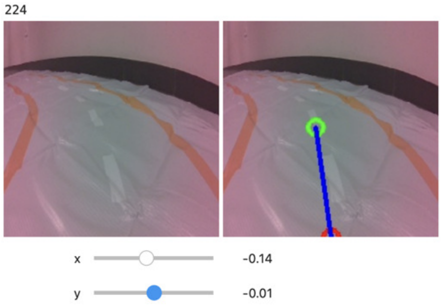

Fig. 3 User interface camera view in JupyterLab widget for training data collection

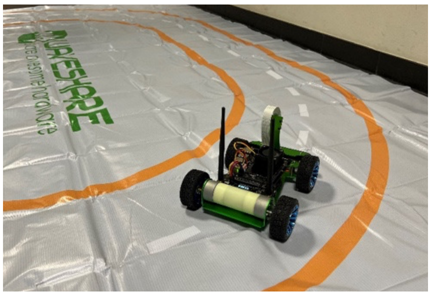

Fig. 4 Actual car position on track during data collection

Having prepared all the described tools and the vehicle on the track in the target environment, the neural network model training can be started. This process should begin with collecting training and validation data. It involves the car going over the track several times, but not continuously and recording the video sequence, but by stopping, changing the position, and taking pictures.

The key information, apart from the camera frame itself, is also the coordinates of the vector of the direction in which the car should move. They are saved in the file name along with a unique identifier for the image. Setting the vector position is a task that must be performed manually using the interface sliders and preview on the widget, or with the use of a controller.

It is required to prepare the dataset in such a way that it contains data allowing to choose the optimal path for the vehicle, but also boundary conditions, e.g. car on the edge of the track for the case to return to the correct trajectory. The larger the dataset that defines how a car behaves, the greater the chance of more precise autonomous driving.

IV. Implementation of a Neural Network for Autonomous Vehicle Movement

The key element of the software that allows the implementation of the task of autonomous vehicle movement is the appropriate selection of the neural network model. The concept of the structure of the implemented solution is presented in Fig. 5.

The algorithm for autonomous driving of the Waveshare JetRacer AI car was based on the ResNet18 neural network. This architecture was chosen for its effectiveness at the level of other solutions such as VGG. The advantage of ResNet solutions, however, is the small size of the model at the level of 22.7 MB [13]. The architecture of this convolutional neural network is 18 layers deep (17 convolutional and one connected) [14]. The shape of its last layer has been changed by the Linear function from the PyTorch library, and finally it has the same number of results as the number of classes in the dataset. The Linear PyTorch function creates a single-layer unidirectional feedforward network in which data only flows from input to output. An important aspect of this type of solution is backpropagation, which consists of improving at each step of the training process due to the correction of weights based on the estimation of the error made by neurons during training. The Adam algorithm was responsible for optimizing the network model with such a structure. The set of collected data (600 frames) was divided as follows: 80% of the training images and 20% of the validation data. To optimize the performance of the software, it was decided to convert the PyTorch model to the TensorRT model, which enables more than three times the increase in the number of processed frames per second, from 29.4 to 90.2 FPS [15].

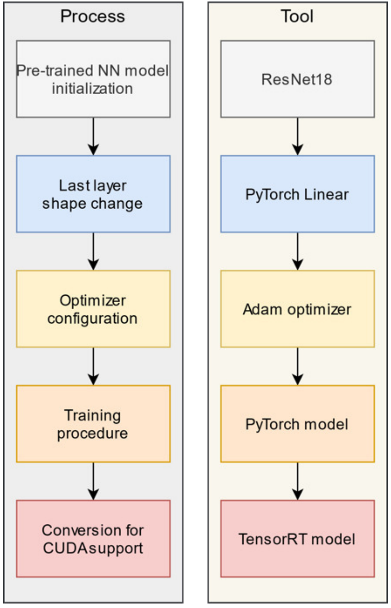

Fig. 5 Block diagram of neural network model for autonomous driving

TABLE II.
MEAN LOSS FUNCTION VALUE FOR DIFFERENT EPOCHS NUMBER

Epochs number	**Training data**	**Validation data**	**Training time [s]**
30	0.042	0.035	754.7
50	0.024	0.018	1229.6
100	0.014	0.008	2459.1

During the training of the neural network, an experiment was carried out to check the impact of the number of epochs on the loss function and the behavior of the vehicle in autonomous driving mode. The epoch is the one-time use of all the cases included in the training set during the whole teaching process. These repetitions are intended to perform many steps of the training algorithms until the error in the output is acceptably small. The following values of the number of epochs were checked during the experiments: 30, 50 and 100. The mean values of the loss function and training times are summarized in Table II.

The average values obtained for the loss function show the high efficiency of the neural network model training process. As expected, the dependence of the decrease in the function value with the increase in the number of epochs can be noticed. The results for the validation data are slightly lower than for the training images, but these are average values, so there is no risk of overfitting phenomenon. The analysis of the complete input data confirmed that the structure of the network, the ratio of the number of training and validation images, as well as the selection of network parameters made it possible to achieve results that allow testing under real conditions.

Due to the use of the neural network model, when the vehicle is moving, an image is taken from the camera, which, after analysis, returns the value of the function responsible for steering the torsion of the front axle of the platform, depending on the model's prediction. In addition, parameters such as steering gain and steering bias have also been implemented, which have to be set manually because they are the result of the physical imperfections of the vehicle's design and the servo used for the steering system.

Depending on the number of epochs implemented during the neural network training process, the following observations were made during real tests. The car, using a model trained for 30 epochs, tended to cross the orange lines at the exit of the curve and cut them off. This problem was regular and repetitive, since crossings of the track limits occurred more or less at the same points on the route. This phenomenon was eliminated for the model with 50 epochs. The vehicle was already moving inside the orange lines, in accordance with the planned trajectory, but on straight sections its path deviated from the central axis marked by a white dotted line. This effect was completely eliminated for a model trained with 100 epochs. The car was driven without problems along the set trajectory, not exceeding the limits of the track and keeping its position in the axis of the route.

V.Conclusions

The tested model of the Waveshare JetRacer AI vehicle allowed for the implementation of an algorithm for the task of autonomous driving along a designated route. Due to the use of the NVIDIA Jetson Nano B01 microcomputer supporting the GPU-based CUDA architecture, it was possible to run a complex model of the neural network. It processes the data obtained from the Waveshare IMX219 160 degree FoV camera, and then sends the control signals to the servo responsible for the steering system.

Preparation of training data for a dataset that reflects the test environment, selection of appropriate parameters of the neural network, and optimization of its performance allow for obtaining a reliable algorithm for autonomous driving. It should be noted that it does not require powerful computing power from efficient graphics cards dedicated to PC devices, but can also be successfully used on embedded devices [16].

The conducted experiments confirm the correct operation of both the hardware and the software part. After an appropriate selection of the number of epochs during the neural network model training process, it is possible to eliminate undesirable behavior of the vehicle, such as going beyond the boundaries of the route or deviation from the set movement trajectory. The trajectory of the movement of Waveshare JetRacer AI model based on the algorithm analyzed is stable.

Further research will allow the system to be expanded with additional sensor modules for more precise environment monitoring (including obstacles detection) using data fusion techniques [17].

References

[1] Yeong, D.J.; Velasco-Hernandez, G.; Barry, J.; Walsh, J. Sensor and Sensor Fusion Technology in Autonomous Vehicles: A Review. Sensors 2021, 21, 2140. https://doi.org/10.3390/s21062140

[2] Baïou M., Quilliot A., Adouane L., Mombelli A., and Zhu Z., Algorithms for the Safe Management of Autonomous Vehicles, Annals of Computer Science and Information Systems. IEEE, Sep. 26, 2021. doi: 10.15439/2021f18.

[3] Podbucki K., Possibilities and limitations of environment monitoring with usage of LiDAR scanner, Przegląd Elektrotechniczny, vol. 1, no. 1. Wydawnictwo SIGMA-NOT, sp. z.o.o., pp. 186–189, Jan. 04, 2022. doi: 10.15199/48.2022.01.40.

[4] SAE International, Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles, SAE J3016 standard, 2021

[5] Jencoe P., The AV from Algorithm to Acceleration, Under the Digital Hood: Adaptive Computing and AI for Autonomous Vehicles, ElectronicDesign, 2020, pp. 2-7

[6] Suder, J.; Podbucki, K.; Marciniak, T.; Dąbrowski, A. Low Complexity Lane Detection Methods for Light Photometry System. Electronics 2021, 10, 1665. https://doi.org/10.3390/electronics10141665

[7] Waveshare, JetRacer AI Kit, AI Racing Robot Powered by Jetson Nano, 17.03.2022, https://www.waveshare.com/jetracer-ai-kit.htm

[8] NVIDIA Corporation, Jetson Nano Developer Kit,17.03.2022, https://developer.nvidia.com/embedded/jetson-nano-developer-kit

[9] Madrin F.P.; Rosenberger M.; Nestler R.; Dittrich P-G.; Notni G., The evaluation of CUDA performance on the Jetson Nano board for an image binarization task, Proc. SPIE 11736, Real-Time Image Processing and Deep Learning 2021, 117360G, 12 April 2021, https://doi.org/10.1117/12.2586650

[10] Waveshare, IMX219 Camera Module, 160 degree FoV, 17.03.2022, https://www.waveshare.com/imx219-d160.htm

[11] NVIDIA AI IOT, JetCard description, 22.03.2022, https://github.com/NVIDIA-AI-IOT/jetcard

[12] Świderski A., Wałęsa S., Monitoring of pedestrian crossings using an embedded system, Bachelor's thesis, Supervisor: Tomasz Marciniak, Auxiliary supervisor: Kacper Podbucki, Poznan University of Technology, 2022

[13] S. Bianco, R. Cadene, L. Celona and P. Napoletano, "Benchmark Analysis of Representative Deep Neural Network Architectures," in IEEE Access, vol. 6, pp. 64270-64277, 2018, doi: 10.1109/ACCESS.2018.2877890.

[14] Mathworks, Documentation of resnet18, 21.03.2022, https://www.mathworks.com/help/deeplearning/ref/resnet18.html

[15] NVIDIA AI IOT, Benchmark of torch2rt function, 21.03.2022, https://github.com/NVIDIA-AI-IOT/torch2trt

[16] Suder, J., Możliwości przetwarzania sekwencji wizyjnych w systemach wbudowanych, Przegląd Elektrotechniczny, No 01/2022, pp. 188-191, https://doi.org/10.15199/48.2022.01.41

[17] Barreto-Cubero, A.J.; Gómez-Espinosa, A.; Escobedo Cabello, J.A.; Cuan-Urquizo, E.; Cruz-Ramírez, S.R. Sensor Data Fusion for a Mobile Robot Using Neural Networks. Sensors 2022, 22, 305. https://doi.org/10.3390/s22010305