AN EXAMPLE OF IMPLEMENTING ADAS SYSTEMS ON A SMALL-SCALE MOBILE PLATFORM

ALEXANDRU AVRAM[1], VLĂDUŢ ADRIAN PETRUŢA[2]

Abstract: In this paper an example of Advance Driving Assistance Systems (ADAS) from automotive are implemented on a small-scale vehicle. The ADAS technologies reefer to integration of multiple electronic systems of sensors and microcontrollers for detecting nearby obstacles or driver errors and respond accordingly. Several functions of ADAS uses cameras to record external environmental images for traffic signs recognition and lane assist. For implementing the above functionalities, modern software techniques are considered for processing the data acquired by the Raspberry camera on the mobile platform. Deep learning and image processing algorithms have been interconnected in a Raspberry Pi controller on the vehicle. Software development and implementation of the above algorithms have been carried out with Phyton programing language. The proposed algorithm is used to calculate the angle of turn required to stay in the current lane while reading the traffic signs for adapting the speed. Results and conclusions are presented at the end of this article.

Keywords: ADAS, autonomous mobility, smart vehicle, image processing, pattern recognition, microcontroller.

1. INTRODUCTION

The Advanced Driver Assistance Systems (ADAS) [1] are interconnected electronic systems that aid drivers in driving safely, parking the car, guiding the vehicle in the lane, and much more. The ADAS systems improves vehicle and road safety by integrating electronic devices such as controllers, radars, cameras, and other sensors, to identify nearby obstacles or driver errors and respond accordingly.

Considerable traffic accidents are caused by human mistake. According to World Health Organization, road traffic accidents would be one of the top five major causes of mortality by 2030. Many of these deaths and injuries can be avoided by driving vehicles that are appropriately equipped with state-of-the-art technology [8], such as ADAS.

[1] *Ph.D., Prof. Eng.,"1st of December 1918" University of Alba Iulia, alex.avram@uab.ro*

[2] *Stud. at "1st of December 1918" University of Alba Iulia, vladut.petruta.ea@uab.ro*

Automatic braking and self-collision avoidance, parking or alcohol sensors, and traffic sign recognition [9] are a few examples that are becoming increasingly popular on the market.

According to recent research [10] conducted by a Swedish insurance company, an autonomous braking system can minimize driver injuries by up to 64% on highways with speeds of up to 50 km/h. Automated braking mitigated the consequences of around 40% of collisions, resulting in no casualties.

The goal of this project is to create a small-scale mobile platform [2] that can maintain its path and obey traffic signs by capturing images in real time and processing them with machine learning and pattern recognition algorithms [1].

The following specific objectives are set to achieve this goal:

- Electronic system design through current consumption calculation from power supply at idle and load.
- Electronic components selection based on technical specifications, drawing electronic control signal diagrams, and projecting PCB layout.
- Developing software algorithms using OpenCV [6] and TensorFlow libraries with use of Phyton programming language.
- Implementing those algorithms in the Raspberry Pi 4 controller.

2. OVERVIEW OF THE ADAS

The Society of Automotive Engineers (SAE) defines six level of driving automation in J3016 standard where SAE Level 0 indicates no automation and required manual control while the highest SAE Level 5 indicates a fully autonomous vehicle. As in year 2020, non-commercially SAE Level 4 vehicle exists in the market yet. In order to reach a higher autonomation level, the vehicle must be able to understand its surrounding environment and make critical decision to prevent accidents. That is where ADAS comes into play in autonomous vehicle. Multiple sensors such as camera are deployed in the vehicle granting visual capability to ADAS [12].

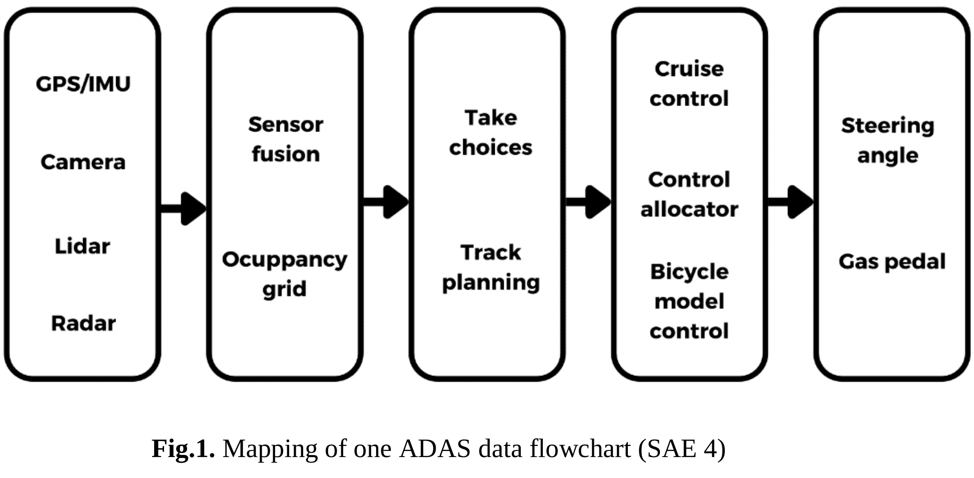


Fig.1. Mapping of one ADAS data flowchart (SAE 4)

Behind the scenes described in Figure 1 is a complex algorithm that provides all these features. This data flux is implemented into a controller through programing language, which allows for quick source code developing. Image management [5] and steering control [2] are the two parts of the programming hidden behind this flowchart. The image captured by the camera is processed using computer vison [6] algorithms, the angle of steering is calculated using a pre-trained model, common method used for sign recognition [7] too. Knowing the required steering angle for lane travel, it is communicated to the steering mechanism, and the vehicle can continue to drive at the set speed until it meets a sign that imposes the next behavior.

3. THE PLATFORM HARDWARE ARCHITECTURE

This project describes the operation of a Raspberry Pi 4 based mobile system that uses software algorithms to acquire and process images captured by a video camera. Image processing [4] is used to calculate the angle of turn required to stay in the current lane. A servo motor provides precise steering for the platform [5], pointing the two steering wheels in the desired direction. A DC motor drives the platform, distributing its rotation to a differential attached to the rear axle. The platform is equipped with a 8MP camera that provides a stable image in daylight conditions, being the main sensor of the system.

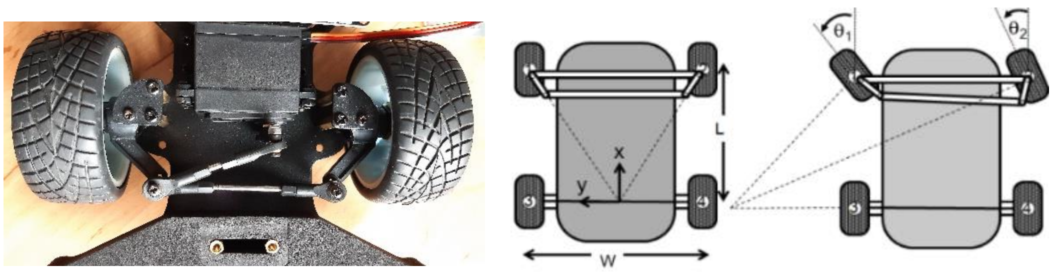


Fig.2. The mechanical steering system of the mobile platform: servo motor for steering (left) and front wheels steering principle (right)

The configuration of the mechanical components that comprise the mobile platform's chassis, steering, and distribution system, derived from a teaching kit specifically developed for such uses. The mobile platform mechanical components include a metal chassis, steering assembly (Figure 2 left), differential gearbox, and traction wheels.

The vehicle features a traditional four-wheel mobile system in terms of kinematics (Figure 2 right). The rear propulsion axle rotates along the X-axis, propelling the vehicle in the direction specified by the front steering axle. The latter conducts XoY plane motions along both axes, rotation along the X axis, and yaw in the Y axis direction.

Unlike conventional road vehicles, which have both traction and steering on the front axle, the proposed mobile platform incorporates a rear wheel propulsion mechanism that improves steering control.

4. THE ELECTRONIC SYSTEM DESIGN

This chapter describes the designing of the electronic system by calculating the current consumption and supply voltage at idle and load, selecting electronic components with technical specifications that match those of the design, drawing electronic control and control schematics, making PCB layout, and assembly of all electronic devices as shown in Figure 3.

The Fritzing freeware software, which permits the construction and viewing of electronic schematics, was used throughout the circuit design stage as presented in Figure 3, left.

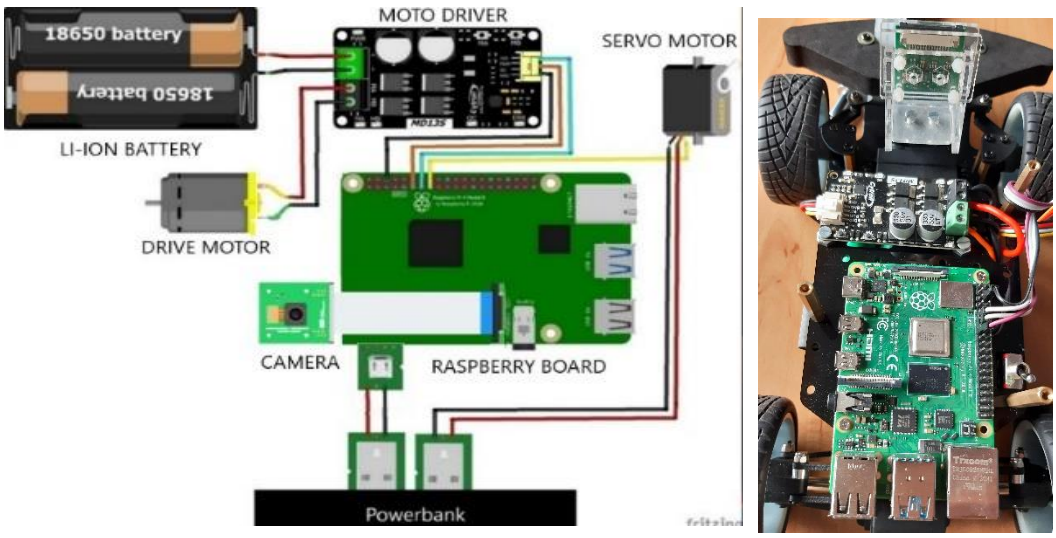


Fig.3. Electronic system architecture: schematic design in Fritzing (left) and hardware assembly on the mobile platform (right)

To achieve the project's goals, it was decided that the electronic circuit should include a DC motor driver to control the engine speeds. Thus, a DC motor is coupled to the driving differential and a servo motor for steering were connected to a microcontroller based on the Raspberry Pi 4 development platform that would process all the information from the environmental camera and make appropriate decisions.

Project advanced to the second phase after designing the electronic scheme, which was the physical execution of the wiring, with the assembly taking place on a test board as presented in Figure 3, right. The user's access to the relevant pins was considered when designing the physical circuit.

5. THE SOFTWARE ALGORITHM DESCRIPTION

The self-driving algorithm works as shown in the diagram below (Figure 4). It begins with the execution of the source code, followed by the initiation of a video capture object that stores each frame. If it captures the current frame, the program can continue;

otherwise, it shows a specified error, and the execution terminates. Following the acquisition of the current picture, preprocessing is required to accomplish the generation of the turn angle [11] and sign recognition [7]. The application uses the initial picture captured to impose the direction of travel based on the pre-trained model that computes the current turn angle.

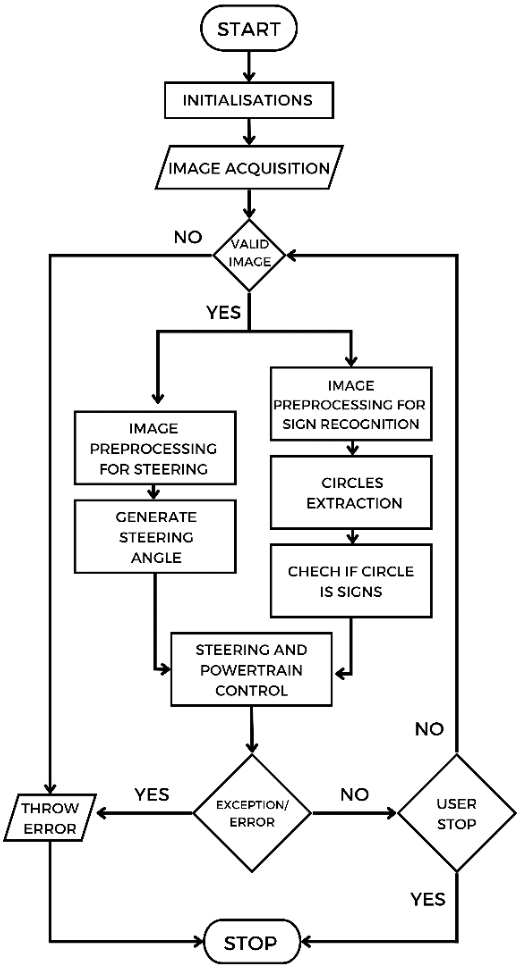


Fig.4. Software algorithm diagram

Once the needed steering angle for lane navigation is determined, the steering control function that drives the wheels to the angle specified on call is called. For sign recognition, all valid circles must be recognized and extracted from the picture, and each must be verified to determine whether it is a traffic sign. When a genuine sign is identified, the action imposed by it is carried out. Only the drive motor control remains to be conducted after establishing the direction of driving.

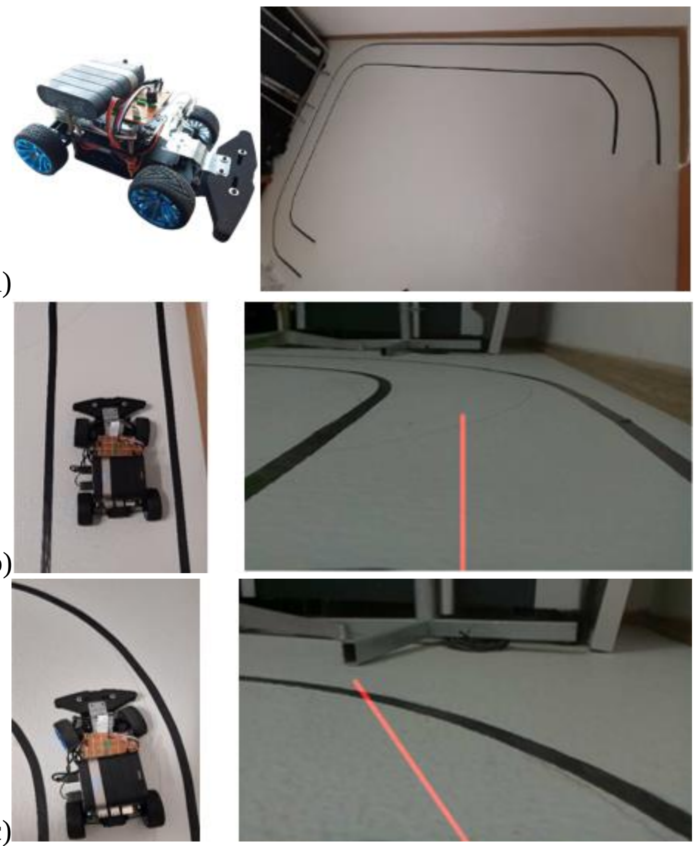
Following that, the process of capturing a fresh image and validating it is repeated in a loop. All the preceding stages are carried out as long as the camera obtains a valid image or as long as execution is not user interrupted.

For the implementation of the algorithm described above, the Python programming language was used, which was chosen for reasons of speed in the drafting of the source code and good compatibility with the development board used.

Generating the drive direction prediction model [8] and sign recognition model [7] was done with the help of Google Colab, a platform that provides free graphics power to run the algorithm generating the drive direction prediction and sign recognition model. The training algorithm used is based on the Keras library, which functions as an interface for TensorFlow, a fundamental library for deep learning [3] used in this project.

6. RESULTS AND INTERPRETATION

Finally, a mini vehicle [11] that stays in its designed lane (Figure 5, a), reads the traffic signs, and satisfies the intended objectives was obtained.



a)

b)

c)

Fig.4. Final results of the autonomous mobile platform: a) final hardware architecture and lane design; b) no steering angle for forward direction; c) left steering angle for changing direction

On the project's lane driving algorithm the following were accomplished: The steering angles (Figure 5, b) are generated at a timestep of between 12ms and 30ms for each angle generated based on the current frame, with an error rate of 5% regardless of the route configuration, either the one in which it was trained or another identical to that. The vehicle, which is usually parked in the middle of the traffic lane, has followed the tight road (Figure 5, c) correctly.

In the sign recognition algorithm, the following milestones was passed: All common round traffic signs are recognized by the system. Only a few of them are used in this project, and they all feature a motor and a directional character, such as "stay right", "turn right ahead", "forward only", "no entrance" or "speed limit x".

7. CONCLUSIONS

Implementation of lane assist and sign recognition functions from ADAS was presented in this article, that marks two of the main achievements proposed from the autonomous driving platform. This project describes the operation of a Raspberry Pi 4 based mobile system that uses software algorithms to acquire and process images captured by a video camera. Image processing is used to calculate the angle of turn required to stay in the current lane. A servo motor provides precise steering for the platform, pointing the two steering wheels in the desired direction. A DC motor drives the platform, distributing its rotation to a differential attached to the rear axle.

The driver assistance concept is well known and there are more ADAS functions that can still be implemented into the above algorithm. There is room for certain changes in the mobile platform electronic system too: like the elimination of the power bank, step down the voltage provided by the propulsion power source and rely exclusively on its energy. Another improvement of the electronics system could be energy monitoring and the redesign of a new PCB for module interconnection.

A major conclusion regarding the future developments is using a radar to monitor the surrounding objects proximity. Implementation of parking assist, collision avoidance, and adaptive cruise control are some future potential directions. Hardware and software optimisation could lead this project to a prototype of self driving vehicle that will be aware of all major parameters souroundig, state of the project that will make it worth the naming of autonomouse vehicle.

AKNODLEGEMENTS

The research presented in this article was founded through the Scientific performance scholarship program granted by the "1st of December 1918" University of Alba Iulia registered with the contract no. 493/04.07.2022.

REFERENCES

[1]. Moujahid A. et al., *Machine Learning Techniques in ADAS: A Review*, 2018 International Conference on Advances in Computing and Communication Engineering (ICACCE), 2018.

[2]. Siegwart R., Nourbakhsh I.R. and Scaramuzza D., *Introduction to autonomous mobile robots*, MIT press, 2011.

[3]. Guo Y., Liu Y., Oerlemans A., Lao S., Wu S., Lew M.S., *Deep learning for visual understanding: A review*, Neurocomputing Volume 187, 2016.

[4]. Alex K., Ilya S., Geoffrey H., *ImageNet Classification with Deep Convolutional Neural Networks*, Neural Information Processing Systems, 2012.

[5]. Peter H., Peter P., *Application of a Visual System for Mobile Robot Navigation (OpenCV)*, AT&P journal, 2010.

[6]. Zelinsky A., *Learning OpenCV---Computer Vision with the OpenCV Library*, IEEE Robotics & Automation Magazine, 2009.

[7]. Johannes S., Marc S., Jan S., Christian I., *The German Traffic Sign Recognition Benchmark: A multi-class classification competition,* The 2011 International Joint Conference on Neural Networks, 2011.

[8]. Mariusz B., Davide D.T., Daniel D., et all., *End to End Learning for Self-Driving Cars,* arXiv preprint arXiv:1604.07316, 2016.

[9]. Dan C., Ueli M., Jonathan M., Jurgen S., *A committee of neural networks for traffic sign classification,* Proceedings of the International Joint Conference on Neural Networks, 2011.

[10]. Cicchino J., *Real-world effects of General Motors Forward Collision Alert and Front Automatic Braking Systems,* Insurance Institute for Highway Safety, 2018.

[11]. Rayan S., Amer S., Jad S., Sarmad T., *The Complete Self-Driving Car Course – Applied Deep Learning,* Packt Publishing, 2019.

[12]. Shengli C., *Advanced driver assistance system based on thermal imaging and machine learning,* Universiti Teknologi Malaysia, 2022.

[13]. Avram Al., Samoila L., Samoila F., Afewerki D., *Designing virtual reality approach for plc applications development,* Annals of the University of Petrosani, Electrical Engineering 23, 2021.

[14]. Remus D., Mircea R., Mihaela A., Samoila F., Stoica P.C., *Computational method and experimental measurement for determining voltage standing wave ratio*. Annals of the University of Petrosani Electrical Engineering 23, 2021.

[15]. Risteiu M., Samoila F., Dobra R., Avram Al., *Implementation of the error management code diagnostic software in distributed electronic circuits using ARM Cortex controller,* Sixth International Conference on Modelling and Development of Intelligent Systems October 3-5, 2019.

[16]. Pasculescu D., Fita N.D., Popescu F.G., Grigorie E., Lazar T., Pupaza C., *Risks Assessment in Terms of Electrical Safety for Power Substation from National Power Grid,* Research Developments in Science and Technology 4, 1-28, 2022.