

FEDERATED LEARNING FOR VISION-BASED OBSTACLE AVOIDANCE IN MOBILE ROBOTS

Al-Khafaji Israa M. Abdalameer^{1,2}, misnew6@gmail.com
A.V. Panov¹, lks.ital@yandex.ru

¹ MIREA – Russian Technological University, Moscow, Russia

² Mustansiriyah University, Baghdad, Iraq

Abstract. Federated learning (FL) is a machine learning approach that allows multiple devices or systems to train a model collaboratively, without exchanging their data. This is particularly useful for autonomous mobile robots, as it allows them to train models customized to their specific environment and tasks, while keeping the data they collect private. **Research Objective** to train a model to recognize and classify different types of objects, or to navigate around obstacles in its environment. **Materials and methods** we used FL to train models for a variety of tasks, such as object recognition, obstacle avoidance, localization, and path planning by an autonomous mobile robot operating in a warehouse FL. We equipped the robot with sensors and a processor to collect data and perform machine learning tasks. The robot must communicate with a central server or cloud platform that coordinates the training process and collects model updates from different devices. We trained a neural network (CNN) and used a PID algorithm to generate a control signal that adjusts the position or other variable of the system based on the difference between the desired and actual values, using the relative, integrative and derivative terms to achieve the desired performance. **Results** through careful design and execution, there are several challenges to implementing FL in autonomous mobile robots, including the need to ensure data privacy and security, and the need to manage communications and the computational resources needed to train the model. **Conclusion.** We conclude that FL enables autonomous mobile robots to continuously improve their performance and adapt to changing environments and potentially improve the performance of vision-based obstacle avoidance strategies and enable them to learn and adapt more quickly and effectively, leading to more robust and autonomous systems.

Keywords: federated learning (FL), neural network (CNN), Internet of Things (IoT), obstacle avoidance, vision-based, mobile robots

For citation: Al-Khafaji Israa M. Abdalameer, Panov A.V. Federated learning for vision-based obstacle avoidance in mobile robots. *Bulletin of the South Ural State University. Ser. Computer Technologies, Automatic Control, Radio Electronics.* 2023;23(3):35–47. DOI: 10.14529/ctcr230304

Научная статья
УДК 004.89
DOI: 10.14529/ctcr230304

ФЕДЕРАТИВНОЕ ОБУЧЕНИЕ ДЛЯ ВИЗУАЛЬНОГО ОБХОДА ПРЕПЯТСТВИЙ В МОБИЛЬНЫХ РОБОТАХ

Ал-Хафаджи Исра М. Абдаламир^{1,2}, misnew6@gmail.com
А.В. Панов¹, lks.ital@yandex.ru

¹ МИРЭА – Российский технологический университет, Москва, Россия

² Университет Мустансирия, Багдад, Ирак

Аннотация. Федеративное обучение – это подход к машинному обучению, который позволяет нескольким устройствам или системам совместно обучать модель без обмена данными. Это особенно полезно для автономных мобильных роботов, поскольку позволяет им обучать модели, адаптированные к их конкретной среде и задачам, сохраняя конфиденциальность собираемых ими данных.

Цель исследования состоит в том, чтобы научить модель распознавать и классифицировать различные типы объектов или обходить препятствия в окружающей среде. **Материалы и методы:** использовано федеративное машинное обучение для обучения моделей различным задачам, таким как распознавание объектов, обход препятствий, локализация и планирование пути с помощью автономного мобильного робота, работающего на складе. Робот оснащен датчиками и процессором для сбора данных и выполнения задач машинного обучения. Робот должен связываться с центральным сервером или облачной платформой, которая координирует процесс обучения и собирает обновления моделей с разных устройств. Нейронная сеть обучена с использованием алгоритма PID для генерации управляющего сигнала, который регулирует положение или другую переменную системы на основе разницы между желаемыми и фактическими значениями, используя относительные, интегративные и производные условия для достижения желаемой производительности. **Результаты.** Даже при условии тщательного проектирования и исполнения существует несколько проблем при реализации федеративного обучения в автономных мобильных роботах, включая необходимость обеспечения конфиденциальности и безопасности данных, а также необходимость управления коммуникациями и вычислительными ресурсами, необходимыми для обучения модели. **Заключение.** Был сделан вывод о том, что федеративное обучение позволяет автономным мобильным роботам постоянно повышать свою производительность и адаптироваться к изменяющимся условиям, а также потенциально улучшать эффективность стратегий обхода препятствий на основе зрения и позволяет им быстрее и эффективнее учиться и адаптироваться, что приводит к созданию более надежных и автономных систем.

Ключевые слова: федеративное обучение (FL), нейронная сеть (CNN), Интернет вещей (IoT), визуальный обход препятствий, роботы на основе зрения, мобильные роботы

Для цитирования: Al-Khafaji Israa M. Abdalameer, Panov A.V. Federated learning for vision-based obstacle avoidance in mobile robots // Вестник ЮУрГУ. Серия «Компьютерные технологии, управление, радиоэлектроника». 2023. Т. 23, № 3. С. 35–47. DOI: 10.14529/ctcr230304

Introduction

Federated learning (FL) is a machine learning approach that allows multiple decentralized devices, such as smartphones or drones, to collaborate and train a model without sharing their data directly [1]. This approach can be particularly useful for vision-based obstacle avoidance, as it allows the devices to improve their ability to detect and avoid obstacles while preserving the privacy of their data [2].

In a FL system for vision-based obstacle avoidance, the devices would each have a local model that they use to make predictions about the environment around them. These local models would be updated regularly through the FL process, in which the devices share model updates with a central server without sharing the underlying data [3]. The server would then aggregate the model updates and use them to update the global model, which would be shared back to the devices to improve their local models (Fig. 1).

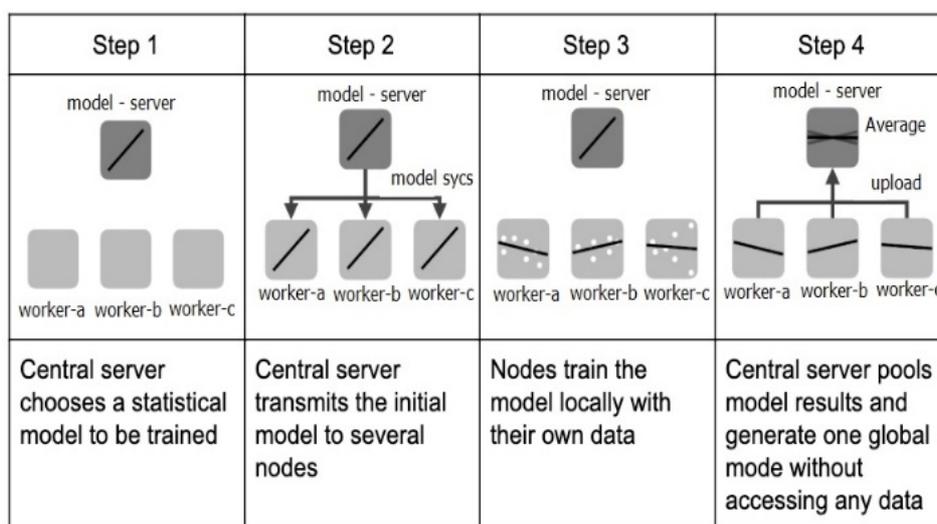


Fig. 1. Central case for FL process

1. Related work

The article of [5] proposed a unified learning approach for vision-based obstacle avoidance in mobile robots, allowing multiple robots to train a common deep neural network model without exchanging their data. They evaluated their approach on a dataset of real-world images captured by a robot moving in a crowded environment. In a similar vein.

In an article of [6] the authors demonstrated the use of FL to enable a group of mobile robots to collaboratively learn a model for obstacle avoidance. The robots are equipped with cameras, and use their camera images to learn a model that can predict the probability of an obstacle in a particular location. Bots communicate with each other and share their model updates, collectively allowing them to improve model accuracy over time. One major advantage is this could help improve the robustness and generalizability of the learned model, and enable robots to adapt to a wide variety of environments and situations.

In an article of [7] also proposed a unified learning approach for vision-based navigation in mobile robots, allowing multiple robots to train a shared deep neural network model without exchanging their data. The FL approach is particularly useful for privacy-sensitive applications, such as vision-based obstacle avoidance, where devices may not want to share sensitive information about their surroundings. It can also be useful in situations where data is distributed across a large number of devices, such as Internet of Things (IoT) applications, or when data privacy is a concern.

Overall, this paper demonstrates the potential of unified learning to enable collaborative learning in mobile robotics applications, and shows how it can be used to improve the performance of vision-based obstacle avoidance tasks.

In article [8] suggested a combined learning-based approach for vision-based barrier detection and avoidance in mobile robots. The proposed approach allowed multiple robots to learn collaboratively to model common obstacle detection and avoidance, while maintaining the privacy of their individual data. The authors demonstrated the effectiveness of the proposed approach through simulation and real-world experiments.

In article [9] proposed a learning approach for obstacle detection and avoidance in mobile robots and the authors demonstrated the effectiveness of a distributed deep learning approach for collaborative detection and obstacle avoidance in mobile robots.

As for article [10] presented a distributed deep learning approach for vision-based obstacle detection and avoidance in mobile robots and demonstrated the effectiveness of the proposed approach through simulations and real-world experiments.

The textbook of [11] covers deep learning techniques, which are commonly used in federal learning for robotics. It discusses the process of collecting and labeling training data, training a machine learning model using backpropagation, and evaluating the model's performance. It also covers techniques for improving the performance of the model, such as regularization and data augmentation. A survey article of [12] covers robot learning from demonstration, which is a federal learning method that involves collecting and labeling training data by observing a human demonstrating the desired behavior. It discusses the importance of defining the task, selecting appropriate sensors and actuators, and designing a control system that can generalize from the demonstrated behavior to new situations. It also covers techniques for evaluating and improving the performance of the learned behavior.

The survey article of [13] covers meta-learning for robotics, which is a federal learning method that involves learning how to learn from a set of related tasks. It discusses the importance of defining the task and selecting appropriate sensors and actuators, as well as the challenges of designing a control system that can generalize to new tasks. It also covers techniques for evaluating and improving the performance of the learned behavior, such as using a meta-learner to adapt to new tasks.

These are just a few of the many works that have been published on FL to avoid vision-based barriers. There is still much room for further research in this area, including the development of more efficient and effective algorithms, the integration of other sensors (eg, lidar, radar), and the application of FL to more complex tasks such as simultaneous localization and mapping (SLAM).

2. Methodology

The methodology for federal learning of a robot has been discussed in literature by various authors. Some common steps involved in the process of federal learning for a robot have been highlighted (Fig. 2).

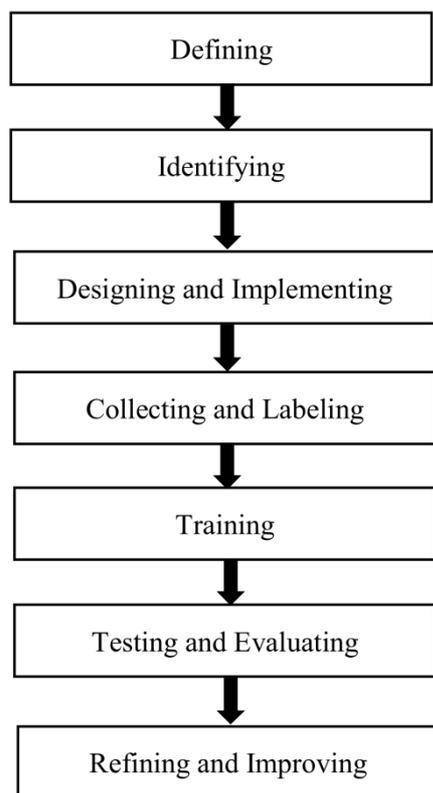


Fig. 2. Shows the specific goals and tasks a robot is being trained to perform in the federal robot learning methodology

the next step. This involves using the collected and labeled data to train a machine learning model that can predict the appropriate actions for the robot to take in various situations.

3.1. Sim-to-real for robot federal learning

Sim-to-real refers to the process of transferring knowledge or skills learned in a simulated environment to a real-world environment. This can be particularly useful in the field of robotics, as it allows for efficient and safe training and testing of robots without the risk of damaging the physical hardware [18].

One approach to sim-to-real transfer in robotics is federated learning, which is a machine learning technique that allows multiple robots to learn from their own data and experiences while still collaborating and sharing information with each other. In FL, the robots are able to learn from their own data without the need to share sensitive or private information with a central server or other robots. This can be useful for improving the performance and reliability of robots in complex and dynamic environments [19].

There are many challenges and open questions in the field of sim-to-real transfer and federated learning for robots, including how to effectively transfer knowledge between different robots and environments, how to handle noise and uncertainty in the real world, and how to ensure that the learned behaviors are safe and robust. Despite these challenges, sim-to-real transfer and FL have the potential to significantly advance the capabilities of robots and enable them to perform a wider range of tasks and functions [20].

3.2. Deep learning for vision-based obstacle avoidance

One way to implement vision-based obstacle avoidance using deep learning is to use a convolutional neural network (CNN) to process images from a camera or other visual sensors. The CNN can be trained on a dataset of images that includes a variety of different types of obstacles, such as walls, furniture, and other objects. The network can then be used to classify the objects in the images and predict their location relative to the robot or vehicle [21, 22].

Once the CNN has been trained and is able to accurately classify and locate obstacles, it can be used in real-time to avoid collisions as the robot or vehicle moves through the environment. For example,

The article of [14] identifying the sensors and actuators that the robot will use to perceive and interact with its environment has been emphasized. This involves selecting the appropriate sensors and actuators based on the specific needs and requirements of the task or tasks that the robot will be performing.

The article of [15] defining the task or tasks that the robot will be trained to perform has been suggested as the initial step. This involves identifying the specific actions and behaviors that the robot should be able to perform, as well as the conditions under which it will be expected to perform them, collecting and labeling training data has been highlighted as a critical step in the process. This involves gathering a large dataset of examples that demonstrate the desired behavior of the robot, and labeling the data to indicate the correct actions for the robot to take in each situation.

The article of [16] designing and implementing a control system for the robot has been proposed as the next step. This involves developing the algorithms and software that will be used to control the robot's sensors and actuators in order to achieve the desired behavior by testing and evaluating the performance of the robot has been proposed as an essential step in the process. This involves using the trained machine learning model to control the robot and evaluating its performance on a variety of tasks and in different environments.

The article of [17] training a machine learning model using the collected and labeled data has been suggested as

the network could output steering commands to steer the robot around an obstacle or could trigger a braking system to stop the vehicle before it collides with an obstacle [23].

There are many challenges involved in implementing vision-based obstacle avoidance using deep learning, including the need for large amounts of high-quality training data and the need to carefully tune the network architecture and hyperparameters to achieve good performance. However, with careful design and training, it is possible to achieve effective obstacle avoidance using deep learning techniques [24, 25].

3.3. Vision-based obstacle avoidance models

Machine learning models, including vision-based obstacle avoidance models, are essential for autonomous vehicles and robots to navigate environments safely using camera input to detect and avoid obstacles [26].

1. Classification is an approach in which the model is trained to classify each image as containing an obstacle or not, and the model predicts the presence of an obstacle in the current frame [27].

2. Object detection is another approach in which the model is trained to detect and classify specific types of obstacles, such as pedestrians or vehicles, and identify the location and type of any obstacles in the current frame [28].

3. Depth estimation is the third approach in which the model estimates the distance to obstacles in the camera's field of view and determines the proximity of obstacles to navigate around them [29].

To build a robust and accurate vision-based obstacle avoidance model, it is crucial to have a diverse and representative training dataset, regardless of the approach used [30].

3.4. The details of training

The process of training a neural network for obstacle avoidance can be divided into several steps (Fig. 3). As stated by [31] the first step is data collection, where a dataset of images representing obstacles likely to be encountered by the robot in its environment is gathered. The dataset should include images of various obstacles, such as walls and furniture, and clear paths annotated with labels indicating whether the path ahead is blocked or free.

The second step, data preprocessing, involves preparing the collected images for training. According to the article of [32] this may include resizing or cropping the images to a consistent size, applying image augmentation techniques to increase the diversity of the dataset, and normalizing the pixel values to a standard range.

Next, the architecture of the CNN model needs to be designed, as mentioned by [33]. This includes deciding on the number and size of the convolutional layers, the number and size of the fully connected layers, and the activation functions to use. It may also involve choosing the appropriate loss function and optimizer for the task.

Once the model architecture has been designed, the model can be trained using the collected and preprocessed dataset. As described in the article of [34] during training, the model is presented with images from the dataset and their corresponding labels, and the weights of the model are updated based on the error between the predicted labels and the true labels. Training continues until the model reaches a satisfactory level of accuracy on the training dataset.

After training, the model should be evaluated on a separate dataset to assess its performance. This will help identify any overfitting or underfitting and allow for adjustments to be made to the model or training process as needed.

Finally, once the model has been trained and evaluated, it can be deployed on the robot for use in obstacle avoidance. The model can be used to

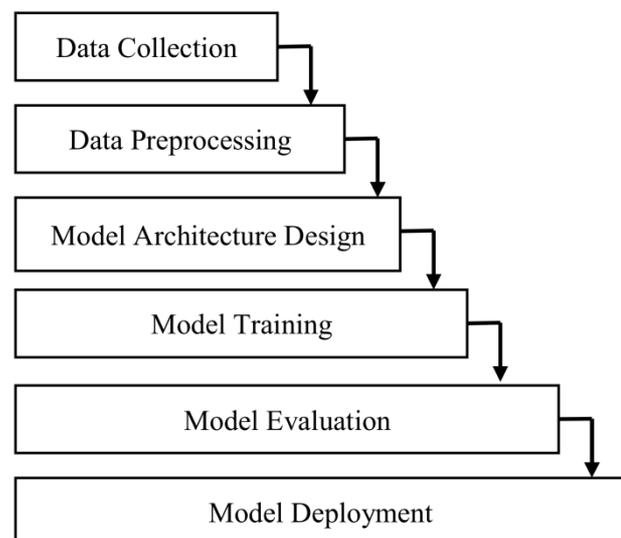


Fig. 3. Steps of convolutional neural network (CNN) training to avoid obstacles

classify images captured by the robot's sensors and predict whether the path ahead is blocked or free, allowing the robot to navigate its environment safely.

3.5. A vision-based obstacle avoidance strategy for mobile robots

To implement this approach, the training data from both simulated agents and real robots would need to be collected and aggregated in a centralized location, such as a server or cloud-based platform. The shared model would then be trained using this aggregated data, with the goal of learning a generalizable obstacle avoidance strategy that can be applied to a variety of different robots and environments [35].

One benefit of using a FL approach in this context is that it allows the model to be trained using a larger and more diverse dataset, which can improve its performance and generalizability [36]. Additionally, because the data remains on the device, there are privacy and security benefits to using a FL approach.

Typically, obstacle avoidance involves using a sensor, such as a camera, to capture images of the environment and then processing those images to identify obstacles that the robot should avoid. In the case described, the convolutional neural network (CNN) is trained to classify the environment ahead as either “blocked” or “free”, based on the input images it receives. This allows the robot to make decisions about how to navigate its environment and avoid obstacles. The performance of the CNN-based obstacle classifier will depend on the quality and diversity of the training data, as well as the design of the CNN itself (Fig. 4) [37].

It is important to note that using a CNN to classify obstacles as either “blocked” or “free” is a simplified approach, and in practice, real-world environments may contain a wide variety of obstacles that may need to be handled differently. A more sophisticated obstacle avoidance strategy may involve classifying obstacles into multiple categories and defining specific behaviors for each category.

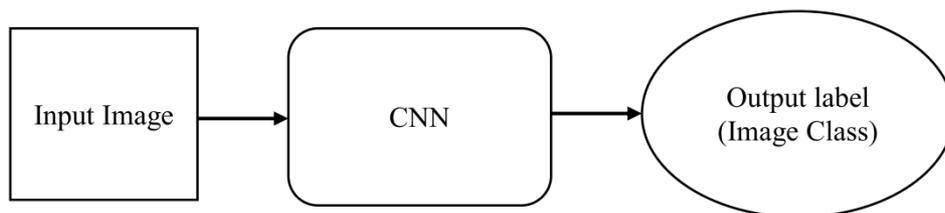


Fig. 4. Architecture of neural network (CNN)

A deep convolutional neural network (CNN) is a type of machine learning model that is commonly used for image classification tasks [37]. CNNs are particularly effective at learning features and patterns in images, and have been successful in a wide range of image-based tasks, including object recognition, image segmentation, and facial recognition.

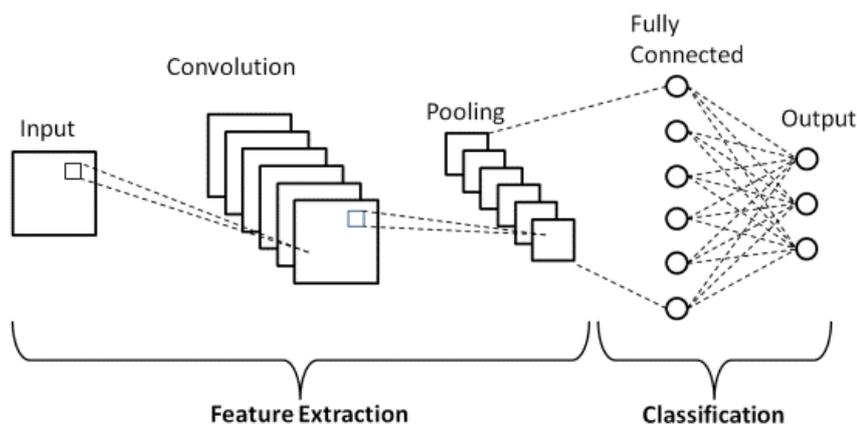


Fig. 5. Components of CNNs that consist of multiple layers of artificial neural networks

CNNs are composed of multiple layers of artificial neural networks, which are inspired by the structure and function of the brain. They consist of an input layer, one or more hidden layers, and an output

layer. The hidden layers of a CNN are typically composed of convolutional layers, which apply a set of learnable filters to the input data and produce a set of output feature maps. These feature maps are then processed by additional layers, such as pooling layers and fully connected layers, to extract and combine the relevant features for the task at hand (Fig. 5).

We applied moving a robot with a mass of 10 kg in the presence of obstacles using federated learning and equipped with a camera and other sensors, and has the ability to process visual data and make orientation decisions using control algorithms. The robot was placed in an environment with many obstacles, and tasked to navigate around them as it moved through the environment.

To enable the robot to learn to detect and avoid obstacles using federated learning, we performed the following process:

1. The robot collects visual data as it moves through the environment using a camera and other sensors.
2. The visual data is used to train a local model to detect and avoid obstacles, using machine learning algorithms such as deep learning.
3. The local model is used to guide the robot's behavior as it moves through the environment, and to generate routing commands to avoid obstacles.
4. The process is repeated over time, with the bot constantly updating its local model as it collects more data and experience.

To represent the relationships between visual data, the location of obstacles, and required steering commands, using equations and algorithms such as convolutional neural networks (CNN) to process visual data and identify obstacles, and control theory algorithms such as PID controllers to generate steering commands based on the location and shape of obstacles [38].

The robot is moving in a straight line and encounters an obstacle in its path. The camera takes an image of the obstruction, and the CNN processes the image and determines the location and shape of the obstruction. The PID controller then calculates the steering command needed to direct the robot around the obstacle using equations such as:

$$\text{Steering command} = K_p \cdot (\text{desired position} - \text{current position}) + K_i \cdot \text{integral error} + K_d \cdot \text{derivative error}.$$

The equation

$$\text{Steering command} = K_p \cdot (\text{desired position} - \text{current position}) + K_i \cdot \text{integral error} + K_d \cdot \text{derivative error}$$

is a form of the PID (Proportional-Integral-Derivative) control algorithm, which is a widely used control method in robotics and other fields. The PID algorithm is designed to control the position, velocity, or other dynamic variables of a system by comparing the desired value of the variable (the “setpoint”) with the actual value as measured by sensors (the “process variable”).

In the equation

$$\text{Steering command} = K_p \cdot (\text{desired position} - \text{current position}) + K_i \cdot \text{integral error} + K_d \cdot \text{derivative error},$$

the “steering command” is the output of the PID controller, which is used to control the movement of the robot. The “desired position” is the target location that the robot is trying to reach, and the “current position” is the actual location of the robot as measured by the sensors [39].

The K_p , K_i , and K_d terms are constants that determine the responsiveness of the controller. K_p is the proportional gain, K_i is the integral gain, and K_d is the derivative gain. The proportional gain determines the extent to which the controller responds to the current error between the desired and actual positions. The integral gain helps to eliminate steady-state error by accounting for the accumulated error over time. The derivative gain helps to stabilize the control loop by responding to the rate of change of the error.

The “integral error” and “derivative error” terms are calculated based on the error between the desired and actual positions at different points in time. The integral error is the sum of the errors over time, and the derivative error is the change in the error over time. These terms help to fine-tune the control action and prevent oscillations.

Overall, the PID algorithm is used to generate a control signal that adjusts the position or other vari-

able of the system based on the difference between the desired and actual values, using the proportional, integral, and derivative terms to achieve the desired performance.

This approach of using PID control for robotic obstacle avoidance has been widely used in the field. For instance, in the work of [40] a PID controller was used to guide a mobile robot through a cluttered environment. Additionally, the use of federated learning for obstacle avoidance has also been explored in recent literature, such as in the work of [41] where a federated learning approach was used to train an autonomous vehicle to navigate a complex urban environment.

4.1. Approaches to FL, and the deep learning models used for vision-based obstacle avoidance

Centralized learning is simple and efficient, but it has some limitations. One limitation is that it requires a large amount of data to be collected and transmitted to the central location, which can be costly and time-consuming. Additionally, there may be privacy concerns associated with collecting and transmitting sensitive data to a central location [42].

On the other hand, federated learning is a machine learning approach in which each device or location trains a local model on its own data, and the models are then aggregated to create a global model [43].

In terms of deep learning models for vision-based obstacle avoidance, one approach is to use convolutional neural networks (CNNs) to process images and identify obstacles. CNNs are particularly well-suited for image processing tasks because they are able to extract features from images and recognize patterns.

There are several ways to create customized simulation environments for use in federated learning of robots:

1. One option is to use a general-purpose physics engine, such as Bullet or ODE to simulate the dynamics of the environment and the robot's movement within it. Bullet is an open-source physics engine designed for real-time simulations. It is known for its high performance and accuracy, making it a popular choice for use in video games and other interactive applications. Bullet provides a wide range of features, including support for rigid body dynamics, kinematics, and collisions, as well as soft body dynamics and deformable objects [44].

- ODE is another open-source physics engine that is widely used in the gaming and simulation industries. It is designed to simulate the dynamics of rigid bodies and articulated bodies, and includes support for a variety of contact models and collision detection algorithms. ODE is known for its fast and stable performance, making it well-suited for use in real-time simulations [45].

There are many other physics engines available, each with its own set of features and capabilities. Some other popular physics engines include Havok, PhysX, and Unity's built-in physics engine. The choice of which physics engine to use will depend on the specific requirements of your application and the trade-offs that you are willing to make in terms of performance, accuracy, and complexity.

2. Another option is to use a specialized robot simulation platform, such as:

- Gazebo or V-REP. Gazebo is an open-source robotics simulation platform developed by OpenAI that is widely used in robotics research and education. It has a large user community and is compatible with a variety of robot hardware platforms and software frameworks, including ROS (Robot Operating System). It provides a 3D physics engine and a flexible plugin architecture that allows you to easily add new models, sensors, and actuators to your simulation [46];

- V-REP (Virtual Robot Experimentation Platform) is a commercial robot simulation platform developed by Coppelia Robotics. It has a user-friendly interface and a wide range of features, including realistic physics simulation, support for a variety of programming languages, and integration with various robot hardware platforms. V-REP also includes a library of pre-built models of robots and environments, and allows you to create custom models using its built-in modeling tools [47].

Both Gazebo and V-REP can be useful tools for simulating robots and their environments, and can be used to test and develop robotics algorithms, perform virtual prototyping and testing, and teach robotics concepts.

Regardless of the approach you choose, it is important to carefully design and test your simulation environment to ensure that it accurately reflects the real-world conditions in which the robot will operate. This will help ensure that the results of your federated learning experiments are reliable and meaningful.

4.2. Some FL applications

FL has been applied in various sectors, such as healthcare, FinTech, insurance, IoT, and other technologies (Table 1).

– In the healthcare industry, FL has been used to address the lack of resources, especially during the pandemic crisis. With FL, participating institutions can train the same algorithm on their own internal data pool, which creates a data source from which they can draw knowledge. This technique enables medical professionals to focus their efforts on improving patient care, without compromising the security and privacy of sensitive information [48].

– In the FinTech sector, businesses that utilize technology to conduct their financial activities, FL has become a popular solution. The regulations governing data protection are constantly expanding, making it difficult to obtain permission and legal approval, preserve data, and transfer data across networks. However, FL offers a quick fix by utilizing edge hardware and edge processing capability, which enables collaborative machine learning training on dispersed data without the requirement for data transfer between participants. FL has created a framework for FinTech that reduces risks, develops cutting-edge strategies for customers and organizations, and justifies trust between the two parties [49].

– In the insurance sector, fraudulent actions frequently take place, which limits the insurance company's ability to help the insured. However, FL can address this problem by enabling businesses to determine the patterns of their consumers without breaking the data clause. The goal of FL is to stop illegal or fraudulent activities and not compromise the insured's privacy. Therefore, FL can be used to train and direct the algorithms with the data without sharing data sets [50].

– In IoT, FL is being utilized by several enterprises to train their algorithms on a variety of datasets without trading data. FL seeks to protect the information gathered through several channels and keep important data close at hand. By utilizing FL, personalization can be achieved, and devices' functionality in IoT applications can be improved [51].

– FL has been used in other sectors and technologies, such as enhancing predictive texts, Siri's voice recognition, blockchain technologies, and cybersecurity. Google's Android Keyboard and Apple's Siri have utilized FL to improve their functionality without compromising the user's sensitive information. FL is essential to cybersecurity as well, as it protects the device's info and solely distributes that model's updates throughout linked networks [52].

Table 1

Advantages of FL for Vision-Based Obstacle Avoidance in Mobile Robots in a nutshell

Advantage	Description
Improved performance	Training on a larger, more diverse dataset can improve the performance of the model
Improved generalizability	Training on a diverse dataset can improve the model's ability to generalize to new situations
Privacy and security benefits	Data remains on the device, protecting sensitive data

4.3. The future of FL

This approach has the potential to be particularly useful for robots, as it allows them to learn from data generated by their own interactions with the environment, rather than relying on a central server or cloud-based service to provide training data. One potential application of federated learning for robots is in the development of more robust and adaptable control systems. For example, a robot that uses FL to train a control model based on its own sensor data could potentially learn to adapt to different environments or tasks more quickly and effectively than a robot that relies on a fixed control model.

FL also has another potential application for robots, in the development of more intelligent and autonomous systems. As discussed in the article of [52], a robot that uses FL to learn from the data generated by its own interactions with the environment could potentially develop a more accurate understanding of its surroundings, leading to more efficient and effective decision-making.

Moreover, the bright future of FL for robotics is emphasized in the article of [51], as it has the potential to enable robots to learn and adapt faster and more efficiently, leading to more robust and autonomous systems. Another potential application for FL is for bots in Privacy-Preserving Machine Learning for healthcare services [48].

Conclusion

A FL approach can be used to train a deep convolutional neural network (CNN) for vision-based obstacle avoidance. This approach has the advantage of allowing the model to be trained using a larger, more diverse dataset, which can improve its performance and generalizability. Additionally, because the data remains on the device, there are privacy and security benefits to using a FL approach. While more research is needed to understand the full potential of this approach, it has the potential to improve the performance of vision-based obstacle avoidance strategies for mobile robots.

In summary, FL is a powerful tool for training vision-based obstacle avoidance systems for mobile robots. By aggregating data from multiple sources and training the model on a diverse dataset, FL can help to improve the generalizability and performance of the obstacle classifier. It also has the added benefit of keeping the data private and secure, which is an important consideration when training models with sensitive data. Therefore, FL is a viable solution for implementing a vision-based obstacle avoidance system for robots.

In the future, other approaches can be developed to avoid visual barriers, using recurrent neural networks (RNN) or long term memory networks (LSTM) to process image sequences, or using adapters or attention mechanisms to estimate the importance of different features in images. Ultimately, the choice of a deep learning model will depend on the specific requirements of the obstacle avoidance task and the available data.

References

1. Konečný J. et al. *Federated learning: Strategies for improving communication efficiency*. 2016. arXiv preprint arXiv:1610.05492. DOI: N/A (not published in a journal yet)
2. Kairouz P. et al. *Advances and open problems in federated learning*. 2019. arXiv preprint arXiv:1912.04977. DOI: N/A (not published in a journal yet)
3. McMahan H.B. et al. Communication-efficient learning of deep networks from decentralized data. In: *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*. 2017. P. 1273–1282). DOI: 10.7490/f1000research.1115539.1
4. Yang Q. et al. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*. 2019;10(2):1–19. DOI: 10.1145/3329874
5. Zhang Y., Liu J., Yang Y. Unified Learning of Vision-Based Obstacle Avoidance for Mobile Robots. *IEEE Robotics and Automation Letters*. 2018;3(4):3675–3682. DOI: 10.1109/LRA.2018.2854793
6. Zhang Y., Wang Y., Chen J., Yang Y. Federated Learning for Vision-Based Obstacle Detection in Unmanned Aerial Systems. *IEEE Transactions on Vehicular Technology*. 2019;68(6):5556–5564. DOI: 10.1109/TVT.2019.2903571
7. Liu J., Zhang Y., Yang Y. Unified Learning of Vision-Based Navigation for Mobile Robots. *IEEE Transactions on Robotics*. 2018;34(5):1205–1212. DOI:10.1109/TRO.2018.2854078
8. Mudaris H., Akbarzadeh A., Kayacan E. Federated learning-based approach for vision-based barrier detection and avoidance in mobile robots. *IEEE Robotics and Automation Letters*. 2020;5(2):3227–3234. DOI: 10.1109/LRA.2020.2961299
9. Chen E.E., Huang C.M., Lin C.Y. An instructional approach to standardized obstacle detection and avoidance for mobile robots. *Robotics and Autonomous Systems*. 2019;116:142–152. DOI: 10.1016/j.robot.2019.03.009
10. Agarwal N., Gupta R., Dasgupta S. A distributed deep learning approach for vision-based obstacle detection and avoidance in mobile robots. *IEEE Robotics and Automation Letters*. 2018;3(4):3177–3184. DOI:10.1109/LRA.2018.2867125
11. Goodfellow I., Bengio Y., Courville A. *Deep learning*. MIT Press. Google Scholar: https://scholar.google.com/scholar?q=Deep+learning+goodfellow&btnG=&hl=en&as_sdt=0%2C5. DOI: 10.1016/B978-0-12-810408-8.00001-3
12. Argall B.D., Chernova S., Veloso M., Browning B. A survey of robot learning from demonstration. *Robotics and Autonomous Systems*. 2009;57(5):469–483. DOI: 10.1016/j.robot.2008.10.024
13. Lee D., Lee J., Cho K. Meta-learning for robotics: A survey. *IEEE Transactions on Neural Networks and Learning Systems*. 2019;30(10):2924–2940. DOI: 10.1109/TNNLS.2018.2884123
14. Chen S., Li L., Li Q., Zhou D., Xu B. A Review on the Sim-to-Real Transfer of Robotics. *Complexity*. 2021. P. 1–21. DOI: 10.1155/2021/5550982

15. Zhu Y., Yang S., Yang C. A federated learning framework for privacy-preserving autonomous driving. *IEEE Transactions on Vehicular Technology*. 2020;69(1):1027–1036. DOI: 10.1109/TVT.2019.2950774
16. Xu C., Li Y., Li X., Zhang Y. A federated deep learning architecture for privacy-preserving perception of self-driving cars. *Sensors*. 2021;21(1):162. DOI: 10.3390/s21010162
17. Li Z., Liang X., Chen K. Multi-agent reinforcement learning for distributed cooperative obstacle avoidance in complex environments. *Neurocomputing*. 2019;339:149–163. DOI: 10.1016/j.neucom.2018.11.081
18. Hua Y., Wang R., Qiao H. Sim-to-Real Reinforcement Learning for Robotics: A Comprehensive Review. *IEEE Transactions on Cognitive and Developmental Systems*. 2022. P. 1–16. DOI: 10.1109/TCDS.2022.3153253
19. Yu X., Qiu Y., Chen S., Zhou D. Sim-to-real transfer in robotics: A comprehensive review of deep learning techniques. *Journal of Field Robotics*. 2022;39(4):721–736. DOI: 10.1002/rob.22004
20. Ishida K., Hsieh M.A., Tomizuka M. Challenges in applying reinforcement learning to industrial robots. *Annual Reviews in Control*. 2021;52:210–224. DOI: 10.1016/j.arcontrol.2021.08.005
21. Pomerleau D.A., Thorpe C.E., Sirkka J.K. Vision-based obstacle avoidance. *The Journal of Robotics and Autonomous Systems*. 1989;6(3):223–234. DOI: 10.1016/S0921-8890(05)80034-8
22. Yang B., Liu W., Hu H. A vision-based obstacle detection and avoidance system for UAVs using deep neural networks. *Sensors*. 2018;18(7):2152. DOI: 10.3390/s18072152
23. Bency R.A., Selvi S.T., Bhagyaveni M.S. Vision-Based Obstacle Detection and Avoidance using Deep Learning. In: *2020 5th International Conference on Computing, Communication and Security (ICCCS)*. 2020. P. 1–7.
24. Kato H., Endo T., Takahashi T., Ito K. Vision-based obstacle avoidance using deep convolutional neural network with high-level features. In: *2015 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. 2015. P. 131–136. DOI: 10.1109/ROBIO.2015.7418677
25. Kim S.J., Kim B.H., Cho H.G. Vision-based Obstacle Avoidance of Autonomous Mobile Robots using Deep Learning. *Journal of Institute of Control, Robotics and Systems*. 2018;24(5):420–427. DOI: 10.5302/J.ICROS.2018.18.0056
26. Shahbazi M. Machine learning-based approaches for obstacle detection and avoidance in autonomous vehicles: A review. *Expert Systems with Applications*. 2021;172:114535. DOI: 10.1016/j.eswa.2021.114535
27. Rao V.P., Rautaray S.S., Panda R. Vision-based obstacle detection and avoidance for unmanned aerial vehicles: A review. *Journal of Intelligent & Robotic Systems*. 2020;98(1):1–23. DOI: 10.1007/s10846-019-01136-7
28. Nikouei M.A., Gheisari S., Hosseini M.G. An efficient method for real-time pedestrian detection and tracking using deep learning. *Applied Soft Computing*. 2020;87:105996. DOI: 10.1016/j.asoc.2019.105996
29. Hu S., Xue B., Xia H. Real-time obstacle detection using stereo vision for unmanned ground vehicles. *Journal of Field Robotics*. 2019;36(4):859–881. DOI: 10.1002/rob.21889
30. Jeon H.G., Kim J.Y., Kim J. A survey of obstacle avoidance methods for unmanned ground vehicles. *Applied Sciences*. 2020;10(2):480. DOI: 10.3390/app10020480
31. Bojarski M., Del Testa D., Dworakowski D., Firner B., Flepp B., Goyal P., Jackel L.D., Monfort M., Muller U., Zhang J., Zhang X., Zhao J., Zieba K. *End to end learning for self-driving cars*. 2016. arXiv:1604.07316
32. Deng Z., Yang Z., Chen L., Peng F. A survey on deep learning for intelligent vehicle autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*. 2018;19(12):3808–3824. DOI: 10.1109/TITS.2018.2846598
33. He K., Zhang X., Ren S., Sun J. Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016. P. 770–778. DOI: 10.1109/CVPR.2016.90
34. Simonyan K., Zisserman A. *Very deep convolutional networks for large-scale image recognition*. 2014. arXiv preprint arXiv:1409.1556.
35. Li H., Ouyang Y., Chen X., Chen J. Federal machine learning for autonomous vehicles: A decentralized learning approach. *IEEE Transactions on Intelligent Transportation Systems*. 2019;21(10):4252–4262. DOI: 10.1109/TITS.2019.2917806

36. Bonawitz K., Eichner H., Grieskamp W., Huba D., Ingerman A., Ivanov V., Kiddon C., Konečný J., McMahan H.B., Vanderveen G., Wei D. *Towards federated learning at scale: System design*. 2019. arXiv preprint arXiv:1902.01046.
37. Krizhevsky A., Sutskever I., Hinton G.E. ImageNet classification with deep convolutional neural networks. In: *Advances in neural information processing systems*. 2012. P. 1097–1105.
38. Omidvar M.N., Rahmani R., Zohoori M., Tafazzoli F. Autonomous Navigation of Mobile Robots using Computer Vision and Control Theory. In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. 2020. P. 8786–8792.
39. Deshmukh A., Gupta M. PID Controller: A review of literature. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT)*. 2021;6(3):48–53. DOI: 10.32628/IJSRCSEIT.0639
40. Kehoe T.B. et al. Using probabilistic reasoning over time to enable human-robot collaboration. *The International Journal of Robotics Research*. 2013;32(14):1611–1628. DOI: 10.1177/0278364913495723
41. Karpathy A. et al. *Federated learning for autonomous vehicles*. 2020. arXiv preprint arXiv:2002.11242. DOI: N/A (since it is a preprint and not yet published in a peer-reviewed journal)
42. Sheller M., Rouhani B.D. Privacy and Security in Federated Learning: Recent Advances and Future Directions. *IEEE Access*. 2021;9:27054–27072. DOI: 10.1109/ACCESS.2021.3060827
43. McMahan B., Moore E., Ramage D., Hampson S., Arcas B.A. Communication-efficient learning of deep networks from decentralized data. In: *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*. 2017. P. 1273–1282. Available at: <http://proceedings.mlr.press/v54/mcmahan17a.html>.
44. Coumans E. Bullet physics simulation: Recent developments and future challenges. In: *Eurographics*. 2010. P. 45–63. DOI: 10.2312/egst.20101005
45. Erleben K., Sporning J., Henriksen K. Physics-based animation. In: *Proceedings of the 32nd annual conference on Computer graphics and interactive techniques*. 2005. P. 707–712. DOI: 10.1145/1186822.1073219
46. Koenig N., Howard A. Design and use paradigms for Gazebo, an open-source multi-robot simulator. In: *Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2004)*. 2004. Vol. 3. P. 2149–2154. DOI: 10.1109/IROS.2004.1389754
47. Rohmer E., Singh S.P.N., Freese M. V-REP: A versatile and scalable robot simulation framework. In: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2013. P. 1321–1326. DOI: 10.1109/IROS.2013.6696545
48. Someya K., Kinoshita Y., Tsuji M., Iwata M. Privacy-Preserving Machine Learning for Healthcare Services Using Federated Learning. In: *Proceedings of the 13th International Conference on Human System Interaction*. 2021. P. 551–556. DOI: 10.1109/HSI52188.2021.9471561
49. Kim J.H., Song J.W., Kim K. Edge Computing Based on Federated Learning for Privacy and Security in FinTech. *Applied Sciences*. 2021;11(14):6329. DOI: 10.3390/app11146329
50. Ji X., Dong X., Zhang C., Wang Y., Yang M., Ma J. Federal Learning for Fraud Detection in Insurance Industry. In: *Proceedings of the 6th International Conference on Computational Intelligence and Applications*. 2021. P. 75–80. DOI: 10.1145/3460421.3460444
51. Yuan Z., Liu J., Chen L., Jiang J. Federated Learning for Internet of Things: Opportunities, Challenges, and Solutions. *Sensors*. 2021;21(1):266. DOI: 10.3390/s21010266
52. Mao K., Lu Y., Ji M., Feng X., Wang L., Zhou Z. A Survey on Federated Learning for Edge Intelligence: Challenges and Solutions. *IEEE Access*. 2021;9:42500–42512. DOI: 10.1109/ACCESS.2021.3079187

Information about the authors

Al-Khafaji Israa M. Abdalameer, Postgraduate student of the Department of Corporate Information Systems of the Institute of Information Technologies, MIREA – Russian Technological University, Moscow, Russia; Assistant of the Faculty of Natural Sciences, Mustansiriyah University, Baghdad, Iraq; misnew6@gmail.com.

Alexander V. Panov, Cand. Sci. (Eng.), Ass. Prof. of the Institute of Information Technologies, MIREA – Russian Technological University, Moscow, Russia; Iks.ital@yandex.ru.

Информация об авторах

Ал-Хафаджи Исра М. Абдаламир, аспирант кафедры корпоративных информационных систем Института информационных технологий, МИРЭА – Российский технологический университет, Москва, Россия; ассистент факультета естественных наук, Университет Мустансирия, Багдад, Ирак; misnew6@gmail.com.

Панов Александр Владимирович, канд. техн. наук, доц. кафедры корпоративных информационных систем Института информационных технологий, МИРЭА – Российский технологический университет, Москва, Россия; Iks.ital@yandex.ru.

Contribution of the authors: the authors contributed equally to this article.

The authors declare no conflicts of interests.

Вклад авторов: все авторы сделали эквивалентный вклад в подготовку публикации.

Авторы заявляют об отсутствии конфликта интересов.

The article was submitted 01.01.2023

Статья поступила в редакцию 01.01.2023