Miniature Mobile Robot Detection Using an Ultralow-Resolution Time-of-Flight Sensor

Jan Pleterski, Gašper Škulj, Corentin Esnault, Jernej Puc, Rok Vrabič, *Member, IEEE*, and Primož Podržaj

***Abstract*— Miniature mobile robots in multirobotic systems require reliable environmental perception for successful navigation, especially when operating in a real-world environment. One of the sensors that have recently become accessible in microrobotics due to their size and cost-effectiveness is a multizone time-of-flight (ToF) sensor. In this research, object classification using a convolutional neural network (CNN) based on an ultralow-resolution ToF sensor is implemented on a miniature mobile robot to distinguish the robot from other objects. The main contribution of this work is an accurate classification system implemented on low-resolution, low-processing power, and low-power consumption hardware. The developed system consists of a VL53L5CX ToF sensor with an 8 × 8 depth image and a low-power RP2040 microcontroller. The classification system is based on a customized CNN architecture to determine the presence of a miniature mobile robot within the observed terrain, primarily characterized by sand and rocks. The developed system trained on a custom dataset can detect a mobile robot with an accuracy of 91.8% when deployed on a microcontroller. The model implementation requires 7 kB of RAM, has an inference time of 34 ms, and an energy consumption during inference of 3.685 mJ.**

***Index Terms*— Binary classification, convolutional neural network (CNN), low power, microcontroller, miniature robot, time-of-flight (ToF), tiny machine learning (TinyML), ultralow resolution.**

I. Introduction

MICROROBOTICS is a rapidly evolving field where object localization for navigation plays a significant role. A recent increase in the computational power of energy-efficient microcontrollers and the development of advanced neural networks enables the implementation of machine-learning-based navigation on low-power robotic systems. Individual implementations, however, need to be adapted to a specific scenario. The combination of sensors, computational hardware, and neural network architecture determines the system's performance in terms of energy consumption and level of accuracy. This is especially important when transitioning from a laboratory setting to a real-world environment where microrobots need to act autonomously. Therefore, the ideal microrobotic navigation system should have low energy consumption, low memory footprint, short inference time, and reliable level of accuracy in a real-world environment with possible unforeseen situations.

Robotic systems for navigation most commonly use visual, sound, and electromagnetic sensors. The most used technologies are 2-D and 3-D LiDAR [1], visible spectrum [2], infrared spectrum [3], time-of-flight (ToF) cameras [4], and ultrasonic sensors [5]. Low-cost miniature ToF sensors are becoming increasingly popular in miniature mobile robotics in combination with microcontrollers [6], [7], [8]. In recent years, more advanced multizone ranging miniature ToF sensors have been introduced that can capture a complete depth image at once without requiring relative motion between the sensor and the observed surface. This type of ToF sensor provides information in the form of depth images that can be used for image classification in miniature mobile robotics applications. Importantly, depth images usually have lower resolution than images from visual spectrum cameras and contain less information useful for classification.

Image classification is a task in which images are associated with specific labels. Traditional methods that can be used for image classification (scale invariant feature transform (SIFT), speeded-up robust feature (SURF), binary robust independent elementary feature (BRIEF), etc.) heavily rely on manually acquired feature descriptors to classify images [9]. To avoid manual feature extraction, classical ML algorithms (support vector machine (SVM), k-nearest neighbor (KNN), decision tree (DT), etc.) can be used that only require a large labeled image dataset, but rely on data separation. However, a more modern approach to image classification is to use deep learning that automatically detects important image features within neural network hidden layers. Based on the literature, deep-learning methods usually outperform classical ML methods [10] and, therefore, a convolutional neural network (CNN) was selected for image classification in this research. CNN architecture is specific for each application and needs to be determined based on the image dataset and used computational hardware.

Image classification with deep learning usually requires computationally very powerful hardware for the learning process, such as graphics processing units and tensor processing units. Implementation of deep-learning models can be done on

Manuscript received 4 January 2023; revised 24 July 2023; accepted 8 September 2023. Date of publication 25 September 2023; date of current version 6 October 2023. This work was supported by the Slovenian Research Agency Javna agencija za raziskovalno dejavnost Republike Slovenije [Public Research Agency of the Republic of Slovenia (ARRS)] through "Junior Researchers" Research Program under Grant P2-0270 and Grant L2-3168. The Associate Editor coordinating the review process was Dr. Lei Zhang. *(Corresponding author: Jan Pleterski.)*

Jan Pleterski, Gašper Škulj, Jernej Puc, Rok Vrabič, and Primož Podržaj are with the Laboratory for Mechatronics, Production Systems, and Automation (LAMPA), Faculty of Mechanical Engineering, University of Ljubljana, 1000 Ljubljana, Slovenia (e-mail: jan.pleterski@fs.uni-lj.si; gasper.skulj@fs.uni-lj.si; jernej.puc@fs.uni-lj.si; rok.vrabic@fs.uni-lj.si; primoz.podrzaj@fs.uni-lj.si).

Corentin Esnault was with the Faculty of Mechanical Engineering, University of Ljubljana, 1000 Ljubljana, Slovenia. He is now with the Brest National School of Engineering, 292280 Plouzané, France (e-mail: c8esnaul@enib.fr).

Digital Object Identifier 10.1109/TIM.2023.3318710

devices with less computational power and memory; however, implementation on microcontrollers, because of their constrained hardware, is only possible for smaller models. Major obstacles when deploying deep-learning models on low-power devices include low processing speed and small memory resources resulting in low inference speed and accuracy [11], [12]. To overcome these challenges and create a successful implementation of deep-learning models, specialized frameworks such as tensorflow lite micro (TFLM), STM Cube AI, embedded learning library (ELL), and so on were developed under the domain of tiny machine learning (TinyML) [13], [14]. TinyML focuses on executing optimized ML models on ultralow-power (<1 mW) MCUs with minimal power consumption [15], [16].

The motivation for this research is the prospect of applying the TinyML approach to ToF data on a microcontroller-based system to improve environment perception for miniature robots in terms of low processing power and energy consumption. The novelty of the research is the application of a deep-learning method on a unique robotic system that combines a low-power RP2040 [17] microcontroller and a VL53L5CX [18] ToF sensor. The problem of performing object classification using CNNs on low-resolution depth images to detect miniature robots on terrain, consisting mainly of sand and rocks is addressed. Accurately classified depth images from a low-resolution ToF sensor using CNNs are demonstrated. The process of CNN architecture development with a comparison of tested architectures and corresponding energy consumption of models during inference is presented. The datasets and the source code are released as open source.

II. Related Work

In this research, ToF sensor input data is collected using pulsed modulation, which involves sending short pulses and measuring the time needed for the pulses to be received [19]. ToF sensors have been used in miniature robotics on constrained devices before the advent of deep learning, with ultrasonic and optical sensors being the most popular.

Although it is a long-standing technology, ultrasonic sensors remain an active area of research. Shen et al. [20] presented a new ultrasonic method for positioning an autonomous mobile robot, in which the developed system consists of three ultrasonic sensors instead of a single one to improve accuracy without using additional temperature sensors. Compared to optical sensors, ultrasonic sensors are more cost-efficient, have lower energy consumption, and require less computing power, which are important features in the field of miniature robotics [21], [22]. However, ultrasonic sensors generally have lower depth measurement accuracy than optical sensors, primarily due to the influence of ambient temperature.

Several studies have been conducted using optical ToF sensors in miniature robotics, mostly developed by ST Microelectronics. Laković et al. [23] used a ToF sensor ST VL53L0X. The accuracy of the sensor is analyzed using experiments with different types of materials and under different illumination conditions. The final result shows that the accuracy of the ToF sensor largely depends on the reflectivity of the surface, especially for darker, less reflective surfaces. Another generation of an ST sensor was used in [24], where a recent swarm robot platform mROBerTO2.0 is presented that uses an ST VL6180X sensor to measure distance.

In addition to distance measurement, applications also use ToF data in systems that use deep learning. This is significant in applications that require the ToF sensing and inference to be on the same device. Device-based inference enables independent unit-level behavior without the need for external connectivity [25]. Since the advent of deep learning, image recognition, detection, and classification have been widely used for machine vision applications using cloud computing, but the announcement of TinyML has motivated researchers to implement similar applications on constrained devices [26].

Callenberg et al. [27] have implemented an ST VL53L1X ToF sensor on a P-NUCLEO-53L1A1 board. The system is equipped with a 32-bit microcontroller, 84-MHz CPU, 512-kB flash memory, and 96-kB SRAM. The system is used for depth imaging, material classification, and object tracking, which were previously limited to computationally more powerful systems. In the case of material classification, the sensor light passes through different materials, and part of this light is reflected back depending on the structure of the material. Based on the histograms obtained with temporal and spatial dimensions, the materials are classified using CNNs.

Some recent research studies use the latest 8×8 VL53L5CX sensor, which can produce an 8×8 depth image of the observed surface. This is a unique feature for a miniature ToF sensor and has previously been implemented for the purpose of indoor navigation [28] and obstacle avoidance [29] to improve autonomous navigation, but no classification of physical objects has been done based on a depth image of this sensor.

However, some researches classify higher-resolution images by implementing computationally more demanding ToF cameras. Ruvalcaba-Cardenas et al. [30] used a low-resolution 64×64 pixel resolution ToF sensor to classify four different objects using a VGG-16 CNN model. The dataset consists of 1615 training images, 285 validation images, and 60 testing images under different brightness conditions. Nash and Devrelis [31] performed a similar classification where a CNN classifier is developed for vehicle classification. The dataset consists of 32×32-sized depth images of different car types. The final model achieved an accuracy of 86.3% on an NVIDIA Jetson TX2 embedded system.

This research is novel by deploying a CNN for object classification on a resource-constrained system, integrating a ToF sensor with significantly lower resolution than those utilized in related publications. This makes the system more challenging to train and deploy, but it also makes it more affordable and accessible.

III. Methods and Experiments

In this section, a detailed description of the microcontroller and the ToF sensor as well as the experimental setup is given. The acquisition of the dataset is described in the following subsections, including data processing and augmentation.

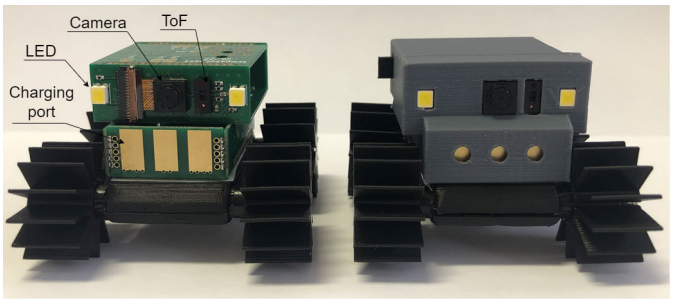


Fig. 1. Mobile robot with the main components used in the research.

Fig. 2. Experimental setup with mobile robots, rocks, and sand-like environment.

Fig. 3. Comparison between the robot and the rock.

Finally, the training and the evaluation of the models are presented.

A. Hardware

The robot platform with an integrated microcontroller, a ToF sensor, and a camera is shown in Fig. 1. The microcontroller used to implement a deep-learning model is the RP2040 from the Raspberry Pi Foundation. The RP2040 is a 32-bit microcontroller chip with 133-MHz dual-ARM Cortex-M0+ cores and 264-kB RAM with an additional 2 MB of flash memory.

The depth image is acquired with the state-of-the-art ST VL53L5CX ToF sensor, released in 2021. It can be used for distance measurements up to 4 m at 60 Hz. It is capable of capturing depth images with a resolution of up to 8 × 8 pixels and a diagonal field of view of 63°. The size of the sensor module is 6.4 × 3.0 × 1.5 mm.

The data generated by the ToF sensor is an array of 64 8-bit integers. This data is transmitted to the microcontroller via an I2C bus. Since most of the features of the robot are lost after 50 cm, the functional range is reduced. Specifically, the 8-bit value range (from 0 to 255) of each pixel represents a distance between 0 and 50 cm.

B. Experimental Setup

The goal of the broader research is to apply the developed classification system to a swarm of robots navigating a sand-like terrain. The robot's classification system was tested in an enclosed environment with sand and rocks. The experimental setup is shown in Fig. 2. The experimental setup (75 × 35 × 25 cm) was filled with 10 kg of sand (grain size: 0.7–1 mm), which enabled us to create various landscape features (e.g., valleys, slopes, hills, etc.). The setup consisted of mobile robots with a footprint of 5 × 5 cm^2.

The rocks were selected on the basis of their similarity to the robot to evaluate the system's ability to distinguish the robot from a rock of similar shape, size, and color. The comparison between the two objects is shown in Fig. 3.

C. Collecting the Dataset

For mobile robots to successfully navigate through unknown terrain in swarms without collisions, they must also be able to detect various objects, including other mobile robots. Due to the specific output of the sensor with a resolution of 8 × 8, no datasets or pretrained models were available for training. Therefore, a custom dataset was created in the experimental setup. The dataset consisted of 4150 unique ToF images with and without the model of a mobile robot [32], which were labeled for binary classification. Fig. 4 shows the field of view (FOV) in combination with the associated Himax HM01B0 robot camera module. The maximum distance of the observed robot from the sensor is 20 cm. The dataset is acquired under different ambient lighting and various shapes of sand-like terrain (e.g., slope, hill, valley, etc.). Separate images of the robot, the rock, and the environment are taken at specific positions of the ToF. The specific collection of the dataset is as follows: First, modify the sand-like terrain, then deploy the mobile robot and collect an image of the terrain, and third deploy another mobile robot and collect an image of the robot in different positions and repeat the same for the rocks. Finally, repeat the process in a different formation of the sand-like terrain.

The dataset is divided into two classes depending on whether the robot is present or not. The dataset is evaluated using five-fold cross-validation, where each dataset consisted of 80% training images, of which 20% were retained for validation, and 20% test images. The noise of the ToF sensor was reduced with a temporal median filter and the training images were augmented with image flipping and Gaussian blur. Sections III-C1 and III-C2 describes the means used to enhance the dataset.

1) Median Temporal Filter: The default ToF images were initially noisy. The noise appeared in the form of inaccurate distance measurements, which can noticeably affect the learning process of the neural network. According to the

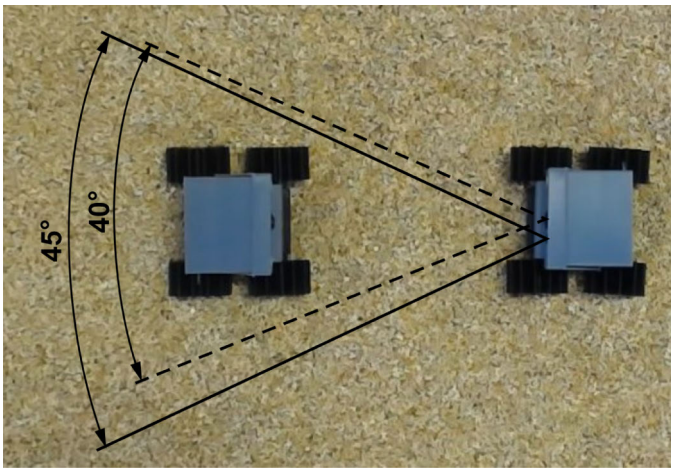


Fig. 4. Comparison of FOV for a ToF sensor (solid) and camera (dashed).

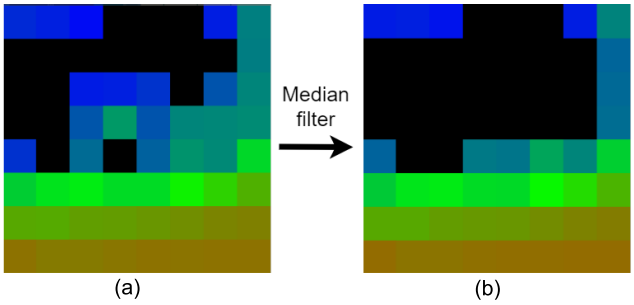


Fig. 5. (a) Single ToF depth image of flat terrain without a robot that includes inaccurate distance measurements that appear as noise. (b) Depth image after median filter of nine sequential measurements is applied showing a reduction in noise.

manufacturer, the noise is due to the sensor's glare, which can be partially removed by setting the manufacturer's recommended sharpener value. The remaining noise was removed with a temporal median filter. In [30], both spatial and temporal median filters were used to remove the noise. However, the ToF sensor in this study had a considerably lower resolution, so the spatial filter was not a viable option. Several median filters were tested on a series of consecutive depth images. The evaluation of the preliminary tests showed that nine consecutive images are the optimal filter size, based on the values of the standard deviations. Therefore, the final temporal median filter was calculated based on nine consecutive distance values per pixel within an overlapping moving average window of a ToF sensor, and a median for each pixel was determined for each image. Considering a sorted list of values in ascending order, the fifth value was used as the median in this case of nine-pixel sequences.

In Fig. 5, an effect of the median filter for one specific instance of a background without the robot can be observed. Fig. 5(a) represents the ToF image before applying the filter, while Fig. 5(b) represents the ToF image after applying the median filter.

To evaluate the effect of median filtering, 20 depth images were taken with and without the filter, and the standard deviations of the measured distances were calculated. The scene of the background remained the same, as shown in Fig. 5. The

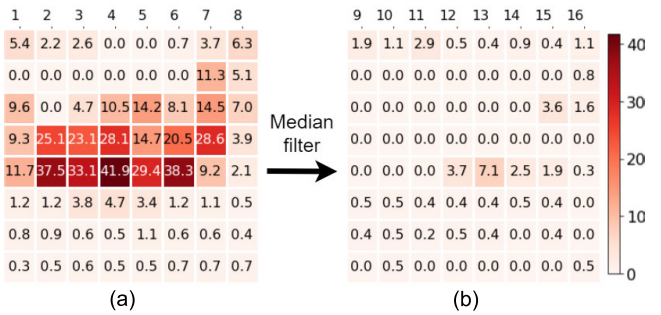


Fig. 6. (a) Standard deviations of inaccurate distance measurements that appear as noise. (b) Standard deviations after the median filter of nine sequential measurements are applied showing a reduction in noise.

evaluation of the median filter is shown in Fig. 6. Note that the median filter removed most of the falsely detected distances.

2) Data Augmentation: The initial dataset of 4150 images was enlarged by offline data augmentation. The dataset was artificially expanded by flipping the images horizontally and adding per pixel Gaussian noise with a standard deviation of five based on the discrete normalized Gaussian distribution: $\mathcal{N}(0, \sigma^2)$.

By mirroring the images and adding Gaussian noise four different datasets were obtained consisting of the initial dataset with 4150 images, the flipped dataset, the dataset with Gaussian noise, and the dataset with combined augmentations.

D. Training

A CNN with a custom architecture was chosen to train the model. The architecture was chosen based on the research in [10] where a comparison of various ML methods was made on microcontroller-based systems, where they found that neural networks bear better performance than traditional ML methods.

The selection of the convolutional network architecture was made based on preliminary results with different CNNs, as the best results were obtained with convolutional layers. Less than three convolutional layers resulted in overfitting and lower accuracy, while more than three convolutional layers resulted in larger model sizes and long inference times that were not applicable to the microcontroller used. The performance of the different models was evaluated using a preliminary dataset of 1000 images, unrelated to the final user-defined dataset. The preliminary results using different CNNs can be seen in Table I and in Fig. 7 where the evaluation was performed on five models with three convolutional layers. Each model was evaluated on the criteria of test accuracy, model size, inference time, and power consumption. All voltage regulators were bypassed during the current consumption measurement. The circuit diagram for the current measurement is shown in Fig. 8.

Note in Fig. 7 that after the third model, the size of the model increases rapidly while the increase in accuracy begins to decrease. From the current measurements in Fig. 9, it can be seen that the acquisition of the depth images accounts for most of the energy consumption. The measurement was carried out in the intermittent ToF mode. The classification

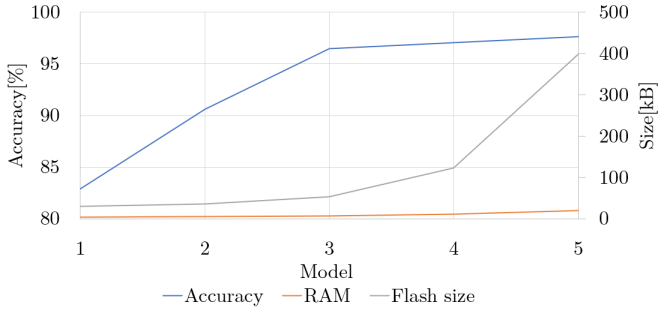


Fig. 7. Preliminary test on five convolutional models regarding the accuracy, RAM, and flash size.

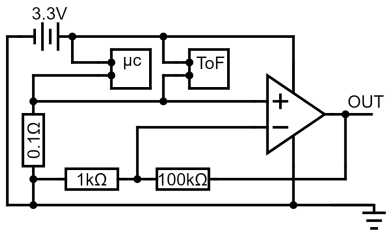


Fig. 8. Electric circuit used for current measurement.

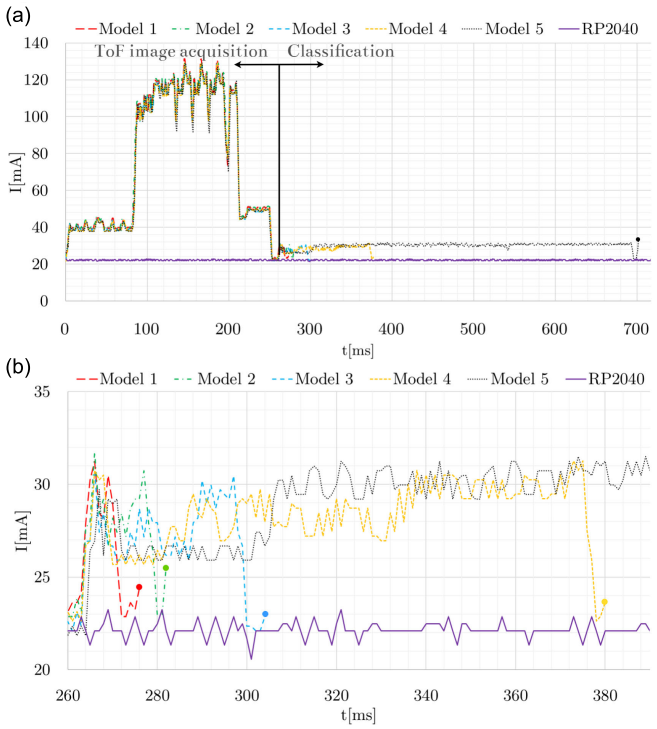


Fig. 9. Current measurements for the five tested CNNs. The colored circles represent the end of each cycle, while the black arrows separate the ToF image acquisition part and the classification part of the program. (a) Whole cycle. (b) Inference part of the cycle.

was, therefore, performed between the ToF measurements so that the current measurements for both operations could be performed separately. ToF image acquisition is independent of classification and consumes 67.06 mJ of energy. The energy consumption for the inference part of the program can be seen in Table I. During classification, each model peaks at about 30 mA of current consumption, but the duration of

TABLE I
COMPARISON OF FIVE MODELS WITH THREE CONVOLUTIONAL LAYERS REGARDING INFERENCE TIME AND INFERENCE ENERGY CONSUMPTION

Model	Convolution channels	Inference time [ms]	Inference energy consumption [mJ]
1	4, 8, 16	7	0.993
2	8, 16, 32	14	1.693
3	16, 32, 64	34	3.685
4	32, 64, 128	112	10.50
5	64, 128, 256	435	42.10

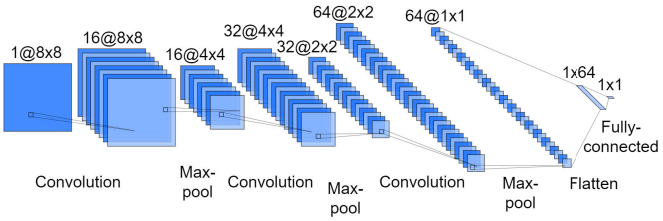


Fig. 10. Final CNN architecture that was selected: convolution channels (16, 32, 64).

classification changes significantly from 7 to 435 ms, while the energy consumption for inference increases from 0.993 to 42.10 mJ. Based on the preliminary results, neural network model 3 was selected as the optimal model in terms of the tradeoff between test accuracy, model size, inference time, and energy consumption.

The models were trained using the EdgeImpulse [33] platform. The platform is used to import data, create a new neural network or use an existing one, train a model, and finally deploy it on an embedded device. The architecture used is shown in Fig. 10 with three convolutional layers, a kernel size of 3, max pooling with a stride of 2, and a fully connected final layer [34].

After training, the model was exported as a C library to be used on the microcontroller. Additionally, 8-bit quantization was performed as this is the most common method of model reduction in constrained embedded systems. The model was quantized by reducing the weights from 32-bit floating-point numbers to 8-bit integers. After quantization, the reduction in model memory was 4.9%, 13.7%, 22.5%, 30.4%, and 36.3%, respectively.

E. Evaluation

The receiver-operating characteristic (ROC) analysis together with the area under the curve (AUC) score was chosen to evaluate the models obtained. The models were evaluated for each version of the dataset. The trained models were first evaluated with a fivefold cross-validation before deployment, where the test images consisted of 20% of the respective dataset without augmentations. Once the model with the best performance was selected, it was deployed on the mobile robot and evaluated again with an ROC analysis. Postdeployment testing consisted of 1000 unique images taken from different perspectives within the testing enclosure with sand. The ROC analysis was performed with the collection of classification data for thresholds with 10%

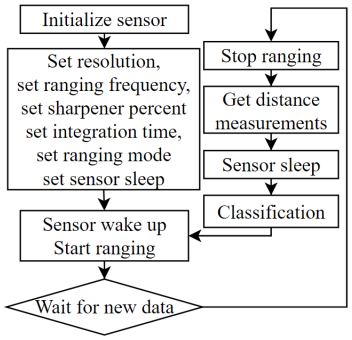


Fig. 11. Program flowchart for the classification system.

TABLE II

COMPARISON OF HOW DIFFERENT MODELS AFFECT CLASSIFICATION ACCURACY, RECALL, AND PRECISION WITH STANDARD DEVIATIONS

Model (Threshold)	Accuracy [%]	Recall [%]	Precision [%]
Default (50%)	96.8 ± 0.004	95.7 ± 0.009	97.7 ± 0.01
Flip (60%)	97.4 ± 0.006	96.7 ± 0.006	97.4 ± 0.006
Gauss (50%)	98.4 ± 0.006	98.5 ± 0.008	97.5 ± 0.008
Augment (50%)	97.8 ± 0.004	97.0 ± 0.005	98.5 ± 0.005
Deployed (60%)	91.8	86.5	94.7

increments. If the probability value was above the threshold after inference, the robot was considered detected. The ROC analysis parameters for each threshold were calculated based on (1) and (2), where TP, FP, TN, and FN represent true positive, false positive, true negative, and false negative results, respectively. The optimal threshold was calculated based on the highest value of the geometric mean between the true-positive and true-negative rates given in (3).

The flowchart of the program used for ToF data acquisition and depth image classification is shown in Fig. 11. The values for sharpener percent (20%), integration time (10 ms), and ranging mode were set according to the manufacturer's recommendations. The source code of the program for the microcontroller is publicly available at [35].

1) *True-Positive Rate:*

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{1}$$

2) *False-Positive Rate:*

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}}. \tag{2}$$

3) *Geometric Mean:*

$$G\text{-Mean} = \sqrt{\text{TPR} \cdot (1 - \text{FPR})}. \tag{3}$$

IV. RESULTS

The ROC comparisons were made to show that the classification system can distinguish a mobile robot from other objects. The first ROC evaluation is shown in Fig. 12. The models were evaluated using fivefold cross-validation with the test datasets consisting of 830 images. The red line represents the classification model trained on the default unaugmented dataset. The best test accuracy of the model

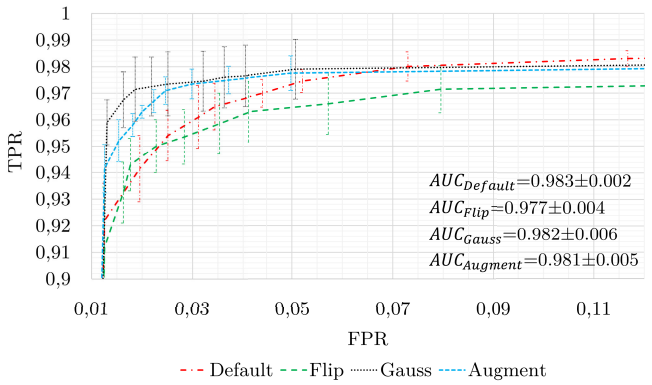


Fig. 12. Predeployment ROC evaluation for every trained model.

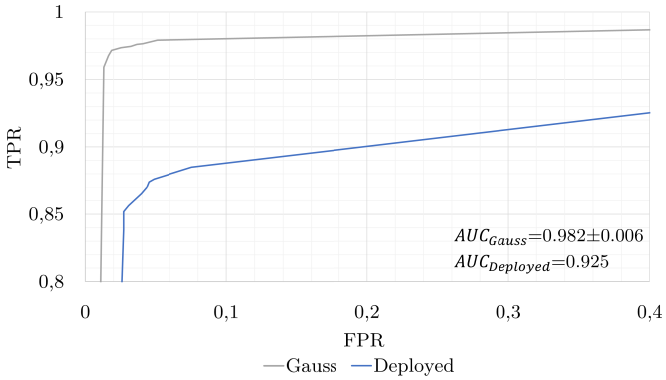


Fig. 13. Comparison between best-performing model predeployment and the model deployed to the microcontroller.

was 96.8%. The green line shows the effect of the horizontal flip augmentation. The best test accuracy increased to 97.4%. The Gaussian augmentation, which is presented with a black line, increased the test accuracy to 98.4%. The model with the largest dataset containing both augmentations achieved a test accuracy of 97.8%. Note that the dataset with both augmentations achieved lower accuracy than the dataset with only the Gaussian augmentation, however, it achieved higher precision. It is also worth noting that despite the model with Gaussian augmentation having a lower AUC score than the default model, it has a much higher accuracy and recall.

Based on the tradeoff of the ROC results, the model with only the Gaussian augmentation was selected as the best performing. The selected model was then used on the mobile robot and evaluated in the experimental setup using 1000 new and unique ToF images. The comparison between the best-performing model before deployment and the model deployed on the mobile robot can be seen in Fig. 13. The model evaluated after deployment achieved an accuracy of 91.8%. The model with the best performance had an inference time of 34 ms with a memory size of 7 kB. The inference time was calculated based on the time it takes the classifier to make the prediction. Accuracy, recall, and precision with standard deviations at optimal thresholds can be seen in Table II. Examples of probability scores for four different specifically selected scenarios of robot detection in the experimental setup are presented in Fig. 14.

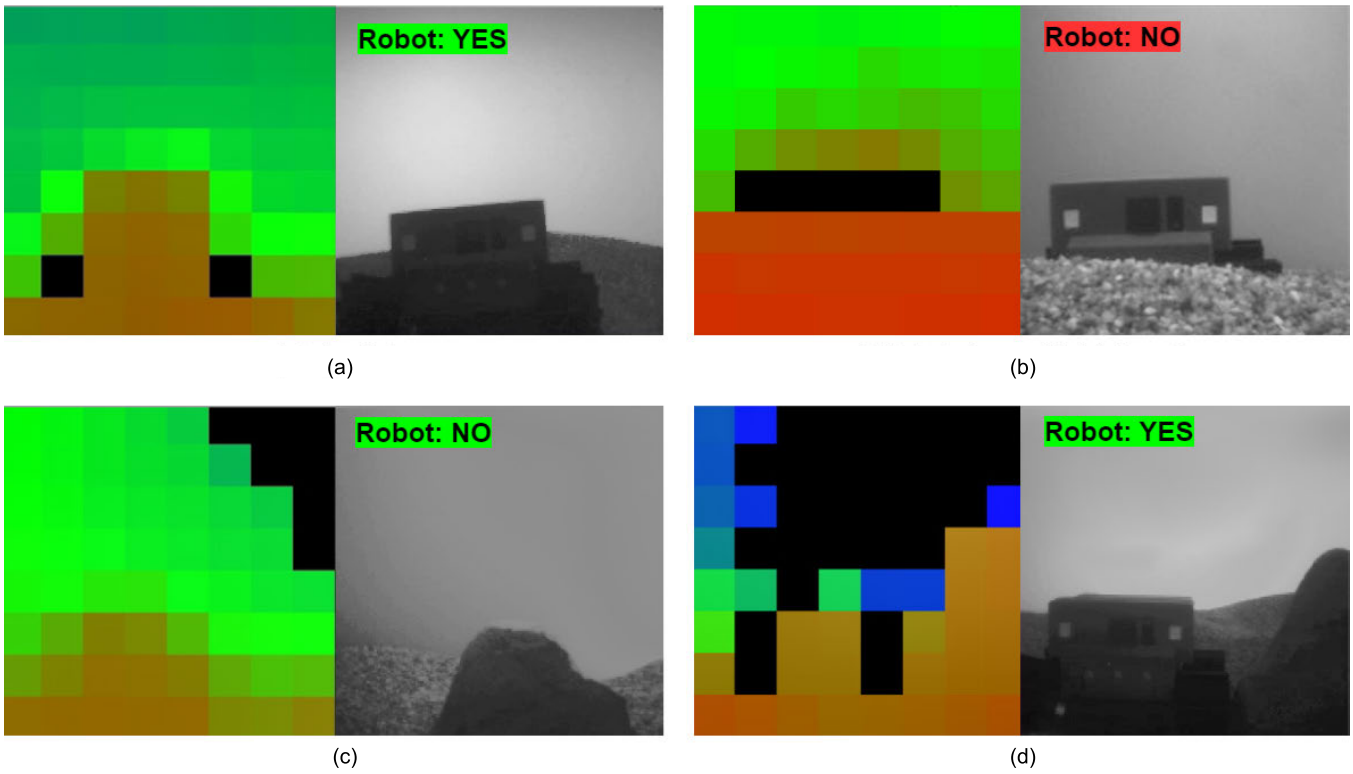


Fig. 14. ToF depth image (left) and a corresponding camera image (right). The grayscale camera images are only added for illustration purposes and have no role in classification. (a) Correctly classified robot. (b) Incorrectly classified robot partially obscured by terrain. (c) Correctly classified background without a robot. (d) Correctly classified robot next to a rock.

V. Discussion

The subject of this work is object classification on a very constrained and low-power device using an ultralow resolution ToF sensor. The presented model was applied to a mobile robot classification system in a closed environment. The model was trained using 4150 images of a particular mobile robot in a sand-like terrain with rocks. The models were first trained and evaluated with and without augmentations before deployment. Based on the evaluation results after training, the model with the best performance was selected. The model was later reevaluated after deployment on the mobile robot. The comparison of the probability scores in Fig. 14 obtained on the mobile robot in the scenario with the robot and in the scenario where only the surrounding objects were present, together with the ROC evaluation indicates, that an accurate classification system was developed by distinguishing the mobile robot from a rock with comparable size, shape, and color.

The best accuracy was achieved with 98.4% before deployment. When the model was deployed on the mobile robot and evaluated after deployment, the classification accuracy dropped to 91.8%. The accuracy after deployment was 6.6% points lower than that of the model before deployment, which means that the model is slightly overfitting.

During the research, it was found that the ToF sensor does not always detect the robots that are in the same position in the same way. This property of the ToF sensor can be seen in Fig. 14. The ideal robot detection can be seen in the detection example in Fig. 14(a), where the robot is shown with approx. nine depth image pixels, where two black pixels represent black wheels. However, in Fig. 14(d), the robot is represented with four depth image pixels. Robots that are in the same position can, therefore, be represented by the ToF sensor in a different way, with a different number of depth image pixels. This ToF feature also hindered optimal classification. Note that the black pixels in the background of the ToF image in Fig. 14(d) are due to the testbed background being more than 50 cm away. By expanding the dataset, it can be seen that although the shapes of the robot are not constant across all depth images, the robot is still detected even in ToF images where the human eye cannot distinguish the robot from the environment.

The most difficult scene for the developed system to classify was the mobile robot behind a sandy hill, as shown in Fig. 14(b). The final classification revealed that the most important features of the robot are the wheels, which are mostly represented as black pixels on each side of the robot body. In Fig. 14(a) and (d), the wheels can be identified by the black pixels around the robot. This is because the IR rays from the ToF transmitter reflect poorly off the black wheels and therefore do not return back to the sensor, resulting in a black pixel. The final trained model achieved an accuracy of 91.8% with a size of 7 kB after quantization and an inference time of 34 ms.

VI. Conclusion

The focus of this research is the application of a deep-learning classification system on a miniature mobile robot. The main contributions are as follows.

1) *Feasibility:* The demonstration of an accurate (>90%) detection of a small mobile robot (5×5 cm^2) on an uneven terrain using low resolution (8×8) time-of-flight sensor and CNNs implemented on a low-power microcontroller (133-MHz CPU and 264-kB RAM).
2) *Tradeoffs:* The comparison of CNN architectures and corresponding energy consumption shows that a small increase in accuracy demands a high increase in energy consumption. Therefore, a compromise is required on low-power devices.
3) *Improvement in Data Preprocessing:* The comparison of various augmentations shows that the model with the Gauss augmentation results in the best accuracy. The analysis shows that a depth image median filter significantly reduces noise in in-depth measurements of the background.
4) *Open Dataset:* The created dataset and source code were released publicly under an open-source license.

The main bottleneck in running the models in this case is not the model size but the processing power of the microcontroller. It should be noted that the RP2040 microcontroller contains two cores, while we only used one core. It was also shown that inference for smaller models consumes a considerably smaller amount of energy compared to depth image acquisition.

Future research will focus on improving the dataset and further testing of architectures. We also plan to apply the developed object classification system to a swarm of autonomous mobile robots, where the developed system will support the autonomous navigation of each robot.

REFERENCES

[1] W. Luo and W. Wei, "A low-cost high-resolution LiDAR system with nonrepetitive scanning," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–10, 2022.

[2] T. Ye, W. Qin, Y. Li, S. Wang, J. Zhang, and Z. Zhao, "Dense and small object detection in UAV-vision based on a global-local feature enhanced network," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–13, 2022.

[3] Y. C. Hou et al., "Development of collision avoidance system for multiple autonomous mobile robots," *Int. J. Adv. Robotic Syst.*, vol. 17, no. 4, pp. 1–15, 2020.

[4] V. Frangez, D. Salido-Monzú, and A. Wieser, "Assessment and improvement of distance measurement accuracy for time-of-flight cameras," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–11, 2022.

[5] B. Bühling, S. Küttenbaum, S. Maack, and C. Strangfeld, "Development of an accurate and robust air-coupled ultrasonic time-of-flight measurement technique," *Sensors*, vol. 22, no. 6, p. 2135, Mar. 2022.

[6] K. N. McGuire et al., "Minimal navigation solution for a swarm of tiny flying robots to explore an unknown environment," *Sci. Robot.*, vol. 4, no. 35, pp. 1–14, Oct. 2019.

[7] J.-T. Lin, C. A. Newquist, and C. K. Harnett, "Multitouch pressure sensing with soft optical time-of-flight sensors," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–8, 2022.

[8] F. Rubio, F. Valero, and C. Llopis-Albert, "A review of mobile robots: Concepts, methods, theoretical framework, and applications," *Int. J. Adv. Robotic Syst.*, vol. 16, no. 2, pp. 1–22, 2019.

[9] N. O'Mahony et al., "Deep learning vs. traditional computer vision," in *Proc. Sci. Inf. Conf.* Cham, Switzerland: Springer, 2019, pp. 128–144.

[10] F. Sakr, F. Bellotti, R. Berta, and A. De Gloria, "Machine learning on mainstream microcontrollers," *Sensors*, vol. 20, no. 9, p. 2638, May 2020.

[11] S. Ma, X. Zhang, C. Jia, Z. Zhao, S. Wang, and S. Wang, "Image and video compression with neural networks: A review," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 6, pp. 1683–1698, Jun. 2020.

[12] A. Elsts and R. McConville, "Are microcontrollers ready for deep learning-based human activity recognition?" *Electronics*, vol. 10, no. 21, p. 2640, Oct. 2021.

[13] J. Gorospe, R. Mulero, O. Arbelaitz, J. Muguerza, and M. Á. Antón, "A generalization performance study using deep learning networks in embedded systems," *Sensors*, vol. 21, no. 4, p. 1031, Feb. 2021.

[14] F. Svoboda, J. Fernandez-Marques, E. Liberis, and N. D. Lane, "Deep learning on microcontrollers: A study on deployment costs and challenges," in *Proc. 2nd Eur. Workshop Mach. Learn. Syst.*, 2022, pp. 54–63.

[15] M. Giordano, N. Baumann, M. Crabolu, R. Fischer, G. Bellusci, and M. Magno, "Design and performance evaluation of an ultralow-power smart IoT device with embedded TinyML for asset activity monitoring," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–11, 2022.

[16] P. P. Ray, "A review on TinyML: State-of-the-art and prospects," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 34, no. 4, pp. 1595–1623, Apr. 2021.

[17] Raspberry Pi Foundation. *RP2040—A Microcontroller Chip Designed by Raspberry Pi*. Accessed: Nov. 10, 2022. [Online]. Available: https://www.raspberrypi.com/products/rp2040

[18] ST Microelectronics. *Time-of-Flight 8×8 Multizone Ranging Sensor With Wide Field of View*. Accessed: May 18, 2022. [Online]. Available: https://www.st.com/en/imaging-and-photonics-solutions/vl53l5cx.html

[19] S. Foix, G. Alenya, and C. Torras, "Lock-in time-of-flight (ToF) cameras: A survey," *IEEE Sensors J.*, vol. 11, no. 9, pp. 1917–1926, Sep. 2011.

[20] M. Shen, Y. Wang, Y. Jiang, H. Ji, B. Wang, and Z. Huang, "A new positioning method based on multiple ultrasonic sensors for autonomous mobile robot," *Sensors*, vol. 20, no. 1, p. 17, Dec. 2020.

[21] M. Dorigo, G. Theraulaz, and V. Trianni, "Swarm robotics: Past, present, and future [point of view]," *Proc. IEEE*, vol. 109, no. 7, pp. 1152–1165, Jul. 2021.

[22] S.-J. Chung, A. A. Paranjape, P. Dames, S. Shen, and V. Kumar, "A survey on aerial swarm robotics," *IEEE Trans. Robot.*, vol. 34, no. 4, pp. 837–855, Aug. 2018.

[23] N. Lakovic, M. Brkic, B. Batinic, J. Bajic, V. Rajs, and N. Kulundzic, "Application of low-cost VL53L0X ToF sensor for robot environment detection," in *Proc. 18th Int. Symp. Infoteh-Jahorina*, 2019, pp. 1–4.

[24] K. Eshaghi, Y. Li, Z. Kashino, G. Nejat, and B. Benhabib, "MROBerTO 2.0—An autonomous millirobot with enhanced locomotion for swarm robotics," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 962–969, Apr. 2020.

[25] M. Merenda, C. Porcaro, and D. Iero, "Edge machine learning for AI-enabled IoT devices: A review," *Sensors*, vol. 20, no. 9, p. 2533, Apr. 2020.

[26] S. S. Saha, S. S. Sandha, and M. Srivastava, "Machine learning for microcontroller-class hardware: A review," *IEEE Sensors J.*, vol. 22, no. 22, pp. 21362–21390, Nov. 2022.

[27] C. Callenberg, Z. Shi, F. Heide, and M. B. Hullin, "Low-cost SPAD sensing for non-line-of-sight tracking, material classification and depth imaging," *ACM Trans. Graph.*, vol. 40, no. 4, pp. 1–12, Aug. 2021.

[28] V. Niculescu, H. Müller, I. Ostovar, T. Polonelli, M. Magno, and L. Benini, "Towards a multi-pixel time-of-flight indoor navigation system for nano-drone applications," in *Proc. IEEE Int. Instrum. Meas. Technol. Conf. (I2MTC)*, May 2022, pp. 1–6.

[29] I. Ostovar, V. Niculescu, H. Müller, T. Polonelli, M. Magno, and L. Benini, "Demo abstract: Towards reliable obstacle avoidance for nano-UAVs," in *Proc. 21st ACM/IEEE Int. Conf. Inf. Process. Sensor Netw. (IPSN)*, May 2022, pp. 501–502.

[30] A. D. Ruvalcaba-Cardenas, T. Scoleri, and G. Day, "Object classification using deep learning on extremely low-resolution time-of-flight data," in *Proc. Digit. Image Comput., Techn. Appl. (DICTA)*, 2018, pp. 1–7.

[31] G. Nash and V. Devrelis, "Flash LiDAR imaging and classification of vehicles," in *Proc. IEEE SENSORS*, Oct. 2020, pp. 1–4.

[32] J. Pleterski, G. Skulj, C. Esnault, J. Puc, R. Vrabic, and P. Podrzaj. (2023). *Miniature Mobile Robot Detection Using an Ultra-Low Resolution Time-of-Flight Sensor Dataset*. [Online]. Available: https://dx.doi.org/10.21227/28ha-8921

[33] S. Hymel et al., "Edge impulse: An MLOps platform for tiny machine learning," 2022, *arXiv:2212.03332*.

[34] A. LeNail, "NN-SVG: Publication-ready neural network architecture schematics," *J. Open Source Softw.*, vol. 4, no. 33, p. 747, Jan. 2019, doi: 10.21105/JOSS.00747.

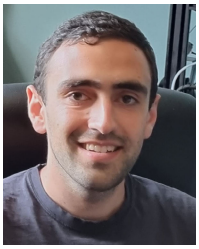
[35] LAMPA. *8×8 Time-of-Flight Miniature Mobile Robot Classification Program*. Accessed: Dec. 12, 2022. [Online]. Available: https://github.com/janplet/Miniature-Mobile-Robot-Detection

Jan Pleterski received the M.Sc. degree from the Faculty of Mechanical Engineering, University of Ljubljana, Ljubljana, Slovenia, in 2020.

He is a Junior Researcher with the Laboratory for Mechatronics, Production Systems, and Automation (LAMPA), Faculty of Mechanical Engineering, University of Ljubljana. His research interests include the implementation of complex algorithms on low-powered constrained devices utilizing deep learning and machine vision.

Gašper Škulj received the Ph.D. degree from the Faculty of Mechanical Engineering, University of Ljubljana, Ljubljana, Slovenia, in 2016.

He is currently an Assistant Professor with the Laboratory for Mechatronics, Production Systems, and Automation (LAMPA), Faculty of Mechanical Engineering, University of Ljubljana. His research interests include mechatronic systems, robotics, distributed systems, and advanced production systems with an emphasis on self-organization.



Corentin Esnault received the M.Sc. degree from the Brest National School of Engineering, Plouzané, France, in 2023.

His research interests include embedded electronics, robotics, and aeronautics.

Jernej Puc received the M.Sc. degree from the Faculty of Mathematics and Physics, University of Ljubljana, Ljubljana, Slovenia, in 2021.

He is a Junior Researcher with the Laboratory for Mechatronics, Production Systems, and Automation (LAMPA), Faculty of Mechanical Engineering, University of Ljubljana. His research interests include simulation design, sim-to-real transfer, and deep learning, particularly reinforcement learning for multirobot environments.

Rok Vrabič (Member, IEEE) received the Ph.D. degree from the Faculty of Mechanical Engineering, University of Ljubljana, Ljubljana, Slovenia, in 2012.

He is an Assistant Professor with the Laboratory for Mechatronics, Production Systems, and Automation (LAMPA), Faculty of Mechanical Engineering, University of Ljubljana. His research interests include robotics, multiagent systems, and reinforcement learning.

Primož Podržaj received the Ph.D. degree from the Faculty of Mechanical Engineering, University of Ljubljana, Ljubljana, Slovenia, in 2004.

He is a Full Professor with the Laboratory for Mechatronics, Production Systems, and Automation (LAMPA), Faculty of Mechanical Engineering, University of Ljubljana. His research interests include control systems, artificial intelligence, and machine vision.